INCORPORATING TAXONOMIES OF CYBER INCIDENTS INTO DETECTION NETWORKS FOR IMPROVED DETECTION PERFORMANCE.

Ryan Warnick
Microsoft Security Research
Microsoft
Redmond, W.A. 98052
ryanwarnick@microsoft.com

August 15, 2025

ABSTRACT

Many taxonomies exist to organize cybercrime incidents into ontological categories. We examine some of the taxonomies introduced in the literature; providing a framework, and analysis, of how best to leverage different taxonomy structures to optimize performance of detections targeting various types of threat-actor behaviors under the umbrella of precision and recall. Networks of detections are studied, and results are outlined showing properties of networks of interconnected detections. Some illustrations are provided to show how the construction of sets of detections to prevent broader types of attacks is limited by trade-offs in precision and recall under constraints. An equilibrium result is proven and validated on simulations, illustrating the existence of an optimal detection design strategy in this framework.

***K*eywords** Cybersecurity, Detection, Performance Metrics, Detection Network, Taxonomy, Nash Equilibrium

1 Introduction

Cybercrime is a multi-billion dollar a year criminal enterprise, with attacks occurring an estimated 2244 [1] times on every device, every day, as far back as 2007; with these numbers likely greatly increasing in the intermediate time window. The World Bank cites that cybercrime incident direct costs are expected to reach 251 billion USD globally by 2031 [2]; with direct costs entailing tangible financial losses, damages, and hardships endured by victims. Additionally, estimates are provided for the expected individual cost of a data breach to increase to 4.35 million USD per breach. As a consequence, the cybersecurity industry has grown at an exponential rate year over year, with cybercrime remediation being a large part of this sector.

Cybercrime remediation is contingent on accurate detection of criminal activity, and detection work is a significant point of investment for cybersecurity vendors; with time to remediation being of utmost importance [3] [4]. Taxonomies for cybersecurity incidents have achieved increasing interest in recent years as the difficulty of navigating the complexity of various computing and networking architectures makes pinning down a uniform ontological mapping between a cyberattack and a categorization difficult. However, having a taxonomy allows one to more appropriately and efficiently tailor remediation and response strategies [5]. Ways to categorize taxonomies can be found in [5]; with a thorough review of older taxonomies, including links to github repositories for presented taxonomy architectures, available in [6]. These taxonomies and ontologies provide a schematic for placing cybersecurity incidents in general populations and sub-populations, and allow the design of effective detections targeting specific categories of attack strategies. The work presented here seeks to assess networks of detections targeting different locations on the taxonomy in terms of their individual, as well as joint, performance.

Various measures of performance exist to quantify how well a binary detection is doing. These are well summarized in statistical literature. They include Signal to Noise Ratio (SNR) [7], Precision and Recall [8], the F1 Score (F1) [9], and many others [10]. The most common metric by far to measure the accuracy of detections in cybersecurity settings is SNR, as this can be readily retrieved from feedback from individual Security Operations Centers (SOCs). However, this metric suffers as a poor descriptor of the performance of a detection, due to not properly assessing accuracy and retrieval ranges in a multifaceted way.

For this work, we examine precision and recall as counter-balanced performance metrics, in the context of cyberthreat taxonomy architectures such as the MITRE ATT&CK[11] framework. The goal of the work is to provide a common set of tools to understand how a taxonomy and set of logical detections built on said taxonomy can be optimized for semi-local (a local set of detections) and global (the complete set of detections) performance. Additionally, results are proven showing the existence of a Nash equilibrium [12] between detections in a detection set acting as noncooperative agents trying to maximize the squared sum of precision and recall in a noncooperative game; leading to an interpretation of optimal detection performance.

The remainder of the article proceeds as follows: Section 2 introduces some preliminary notation and concepts to understand the rest of the article; with Section 2.1 introducing notation related to the taxonomy and detections, Section 2.2 introducing the notation related to the performance metrics, and Section 2.3 introducing the concept of a conditional detection and notation for that aspect of the work. Section 3 contains the principal results of the work. Section 4. Section 5 describes the principal contributions of the article; with possible future directions outlined at the end of that section. An Appendix (Section A) contains the proofs of the theorems.

2 Notation and Basic Concepts

We introduce some notation here to assist in navigating the article. Section 2.1 introduces the framework for modeling the taxonomical structure and detections, and relevant notation. Section 2.2 illustrates the notation and concepts for the relevant performance metrics. Section 2.3 outlines dependency structures between detections common in SOC operations. Section 2.4 ties this all together with the overlaid network of detections, and consolidates the notation of the detections into a directed acyclic graph.

2.1 Notation for Taxonomy and Detections

A diagram of a simple taxonomy for an event is provided in Figure 1, with a starting level in the taxonomy leading into several potential branches or leaves, with a leaf terminating and a branch leading to other potential branches or leaves.

Assume each branch is an ordered set. Our current location in the taxonomy is denoted by T_{ϵ_1} where ϵ_1 denotes the current branch we are on. We then partition the current branch into associated leaves or branches by appending the index of the next leaf or branch ϵ_2, to navigate to $T_{\epsilon_1\epsilon_2}$, where ϵ_2 is a branch or leaf of ϵ_1. This is repeated until an endpoint is reached: $T_{\epsilon_1,\ldots,\epsilon_n}$. Note that n is always less than or equal to the depth of the taxonomy.

Each branch or leaf represents an opportunity for a detection, with detections at increasingly low levels in the hierarchy. Indicate a detection by $D_{\epsilon_1,\cdots\epsilon_n}$.

2.2 Notation for Performance Metrics

Given a detection $D_{\epsilon_1,\ldots,\epsilon_n}$ for arbitrary $n \leq$ the depth of the taxonomy, we denote by FP the total number of false positives of the detection, FN the total number of false negatives of the detection, TP the total number of true positives of the detection, and TN the total number of true negatives of the detection.

Precision and Recall are statistical measurements indicating different types of performance of a particular detection [8]. In layman's terms, precision is how targeted a detection is, and recall is how broad of a stroke the detection takes in what it signals on. The formulae for these two quantities are as follows:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

Figure 2 and 3 give illustrations, for toy binary detections, of the reciprocal relationship between Precision and Recall. Figure 2 shows a detection with a quantitative threshold between 0 and 1, with higher thresholds indicating more parsimonious selection of when to fire. As precision goes up, recall goes down; and vice versa. Figure 3 instead shows

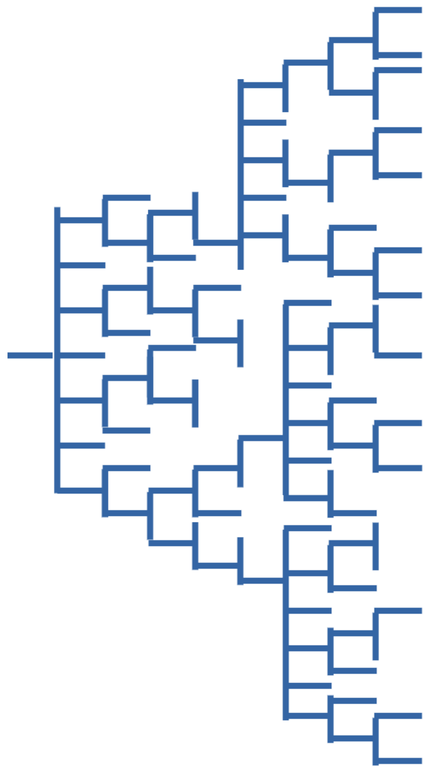

Figure 1: Illustration of a taxonomy tree for a given attack strategy and potential leaves (endpoints) of detections in the taxonomy. The taxonomy has ten levels with each branch branching off into new leaves or branches.

a logical detection (if X occurs fire the signal), and as the resolution of the filter (the specificity of its steps in the logical chain) goes up, precision goes up; and conversely recall goes down.

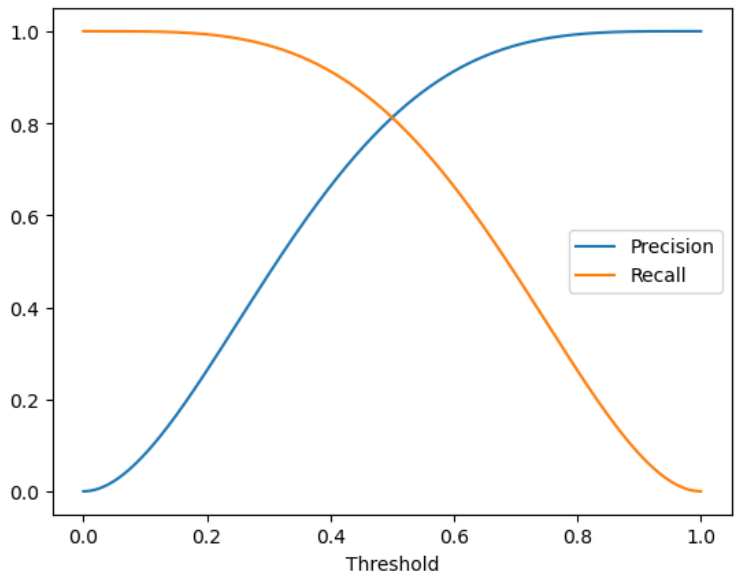

Figure 2: Illustration of reciprocal relationship between precision and recall of statistical detection with threshold between 0 and 1.

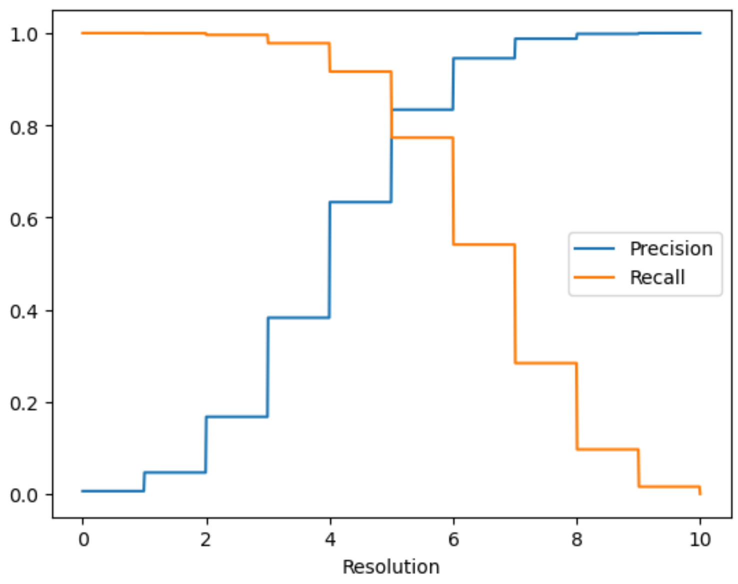

Figure 3: Illustration of reciprocal relationship between precision and recall for logical filter of increasing resolution.

A thorough overview of performance metrics for binary classifiers can be found in [10]; including but not limited to Precision and Recall.

We denote the precision and recall of a detection by applying the operators $Precision(\circ)$ and $Recall(\circ)$ in a conceptual scenario to the detection $D_{\epsilon_1,\ldots,\epsilon_n}$.

It is important to note that precision and recall can be thought of as estimates from a conceptual scenario of conditional probabilities [13]. Let $Y_{\epsilon_1,\ldots,\epsilon_n} \in \{P, N\}$ denote the true class of the signal at $T_{\epsilon_1,\ldots,\epsilon_n}$, and $D_{\epsilon_1,\ldots,\epsilon_n} \in \{P, N\}$ denote the estimated class. Then:

$$Precision(D_{\epsilon_1,\ldots,\epsilon_n}) = P(Y_{\epsilon_1,\ldots,\epsilon_n} = P | D_{\epsilon_1,\ldots,\epsilon_n} = P) \tag{3}$$

$$Recall(D_{\epsilon_1,\ldots,\epsilon_n}) = P(D_{\epsilon_1,\ldots,\epsilon_n} = P | Y_{\epsilon_1,\ldots,\epsilon_n} = P) \tag{4}$$

We work in the rest of this paper with idealized operators for precision and recall, $\lim_{N\to\infty} Precision(D_{\epsilon_1,\ldots,\epsilon_n})$ and $\lim_{N\to\infty} Recall(D_{\epsilon_1,\ldots,\epsilon_n})$; equating to what would happen under the true underlying distribution $P(Y_{\epsilon_1,\ldots,\epsilon_n})$ and $P(D_{\epsilon_1,\ldots,\epsilon_n})$ in Equations 3 and 4. Or, in other words:

$$Precision(D_{\epsilon_1,\ldots,\epsilon_n})(\lambda) = \lim_{N\to\infty} \frac{TP(\lambda)}{TP(\lambda) + FP(\lambda)} \tag{5}$$

and

$$Recall(D_{\epsilon_1,\ldots,\epsilon_n})(\lambda) = \lim_{N\to\infty} \frac{TP(\lambda)}{TP(\lambda) + FN(\lambda)} \tag{6}$$

For a particular threshold of the detection λ.

Note that the precision and recall of a detection depend upon what threshold we use (either quantitative or logical). Some detections will have multiple thresholds that have to be set (having some logical and some quantitative statements, or multiple expressions all of the same type).

Denote by $\Lambda_{\mathcal{D}} = \prod_{D\in\mathcal{D}} \Lambda_D$ the cross product of the set of all possible values for each individual threshold that needs to be set for detection $\mathcal{D}$, and λ_D compact interval contained individual element of $\Lambda_{D_{\epsilon_1,\ldots,\epsilon_n}}$. We assume that Λ_D is continuous (or upper hemi-continuous) for all $D \in \mathcal{D}$:

$$Precision(D)(\lambda_D) = \frac{P(Y_D = P, D = P)(\lambda_D)}{P(D = P)(\lambda_D)} \tag{7}$$

and

$$Recall(D)(\lambda_D) = \frac{P(Y_D = P, D = P)(\lambda_D)}{P(Y_D = P)} \tag{8}$$

are piecewise continuous or piecewise constant over λ_D.

2.3 Conditioned Detections

A detection can be influenced by another detection. That is, if a detection has a logical filter in it based on whether another detection has fired or not. Denoting a pair of detections $D^{(1)}_{\epsilon_1,\ldots,\epsilon_n}$ and $D^{(2)}_{\epsilon_1,\ldots,\epsilon_m}$, we denote that $D^{(2)}_{\epsilon_1,\ldots,\epsilon_m}$ is conditioned on $D^{(1)}_{\epsilon_1,\ldots,\epsilon_n}$ by writing, in an abuse of probability notation, $D^{(2)}_{\epsilon_1,\ldots,\epsilon_m}|D^{(1)}_{\epsilon_1,\ldots,\epsilon_n}$. We are using the operator $|$ to denote a *design constraint* (operational conditioning). Not a stochastic conditional distribution.In the conditional detection scenario, optimizing a detection to have improved Precision or Recall influences the Precision or Recall of dependent detections. From here forward we assume all relevant detections are contained in the set $\mathcal{D}$

This equates to a change in measure of $D^{(2)}_{\epsilon_1,\ldots,\epsilon_m}|D^{(1)}_{\epsilon_1,\ldots,\epsilon_n}$ on the probability distribution over $D^{(2)}_{\epsilon_1,\ldots,\epsilon_m}|D^{(1)}_{\epsilon_1,\ldots,\epsilon_n} = P$ vs. $D^{(2)}_{\epsilon_1,\ldots,\epsilon_m}|D^{(1)}_{\epsilon_1,\ldots,\epsilon_n} = N$, where the mass in the binary random variable (dependent upon λ) is renormalized in $D^{(2)}_{\epsilon_1,\ldots,\epsilon_m}|D^{(1)}_{\epsilon_1,\ldots,\epsilon_n}$ conditional on $D^{(1)}_{\epsilon_1,\ldots,\epsilon_n}$, similar to a typical binary random variable/binary random variable probabilistic relationship.

Figure 4 provides an illustration of what conditional detections signify. The dependencies between the FN, TN, FP, and TP of $D^{(2)}_{\epsilon_1,\ldots,\epsilon_m}$ are dependent on what occurs with $D^{(1)}_{\epsilon_1,\ldots,\epsilon_n}$.

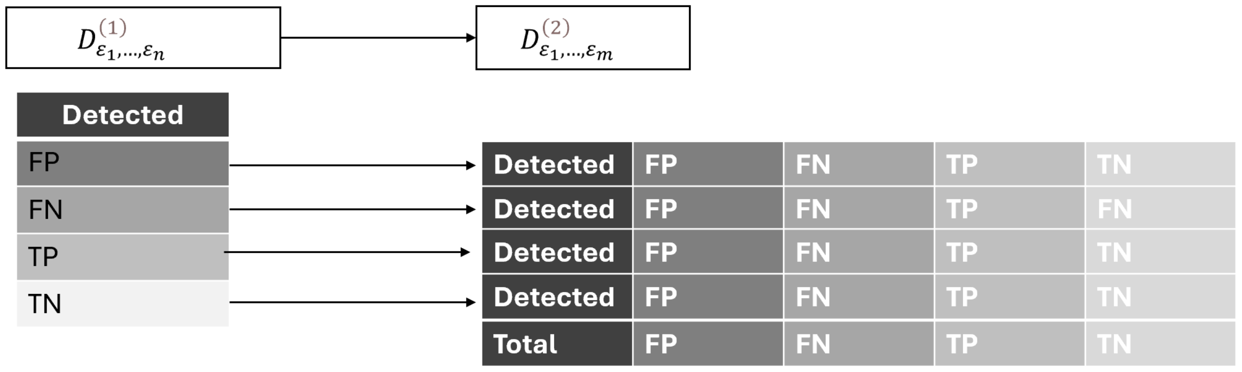

Figure 4: An illustration of how a logical detection can be conditioned on another logical detection, and how the event of a FN, TN, FP, or TP in $D^{(1)}_{\epsilon_1,\ldots,\epsilon_n}$ and influence the outcome in $D^{(2)}_{\epsilon_1,\ldots,\epsilon_m}$.

Additionally, detections can depend upon sets of other detections. This constructs a type of structure analogous to a Bayes Network [14] of conditional detections. Where a detection D is conditioned on another set of detections, and we denote its conditioning set $\mathcal{P}(D)$ and term it the parent set. This gives us the following notational representation:

$$D|\mathcal{P}(D) \tag{9}$$

This creates, as opposed to the 4×4 table in Figure 4, the cross product of $m + 1$ times 4 dimensional vectors to create a table of dimension $m + 1$. Each axis has 4 elements; with each element being an $m + 1$-tuple of FP, FN, TN, and TP.

We introduce a concept here to make the proofs in Section 3 more appropriately general in concordance with the framework just discussed.

Operational gating. A detector D fires operationally ([15]) if its internal rule $R_D(\lambda_D)$ is positive *and* its gate condition over parents holds. We analyze two canonical gates:

$$\text{AND:} \quad D_\lambda^{\text{op}} = R_D(\lambda_D) \wedge \bigwedge_{C \in \mathcal{P}(D)} C_{\lambda_C}^{\text{op}}; \qquad \text{NOT:} \quad D_\lambda^{\text{op}} = R_D(\lambda_D) \wedge \bigwedge_{C \in \mathcal{P}(D)} \neg C_{\lambda_C}^{\text{op}}.$$

Results extend to mixed monotone gates by isotonicity (Remark on isotonicity and what we mean by internal rule outlined in Remark 1).

2.4 Graphs on Detections

We construct a graph of conditional detections and assume the graph is directed and acyclic. Denote the graph $\mathcal{G} = (\mathcal{D}, \mathcal{E})$, where $\mathcal{D}$ is defined as previously, and $\mathcal{E}$ is the edge set of directed edges between the taxonomy detections and other conditioning detections existing as ordered pairs. E.g. $(D_i, D_j) \in \mathcal{E}$, for detections $D_i, D_j \in \mathcal{D}$, if and only if $D_i \in \mathcal{P}(D_j)$.

An example of a simple set of detections A, B, C, D, E on the taxonomy presented in Figure 1 is provided in Figure 5. Denote These detections have locations in the taxonomy T_A, T_B, T_C, T_D, T_E, but the graph existing between them exists as the set $\mathcal{D} = \{A, B, C, D, E\}$ and edge set $\mathcal{E} = \{(A, C), (A, E), (C, E), (B, D), (D, E)\}$. The conditional detection graph is denoted by $\mathcal{G} = (\mathcal{D}, \mathcal{E})$.

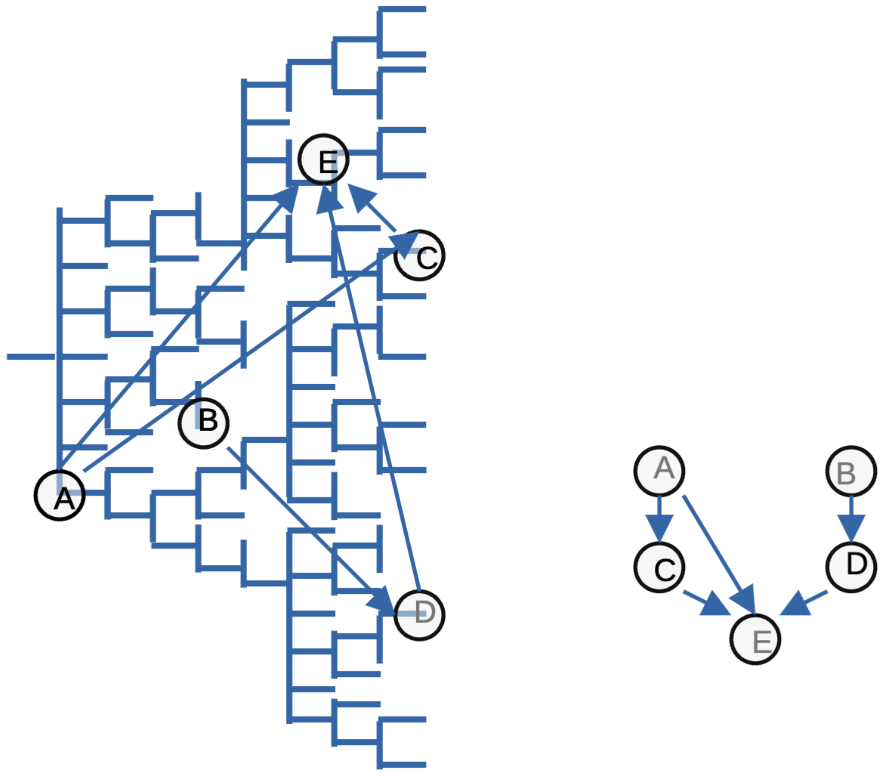

Figure 5: Illustration of the graph G generated by the set of conditional detections in the taxonomy presented in Figure 1. The taxonomy to the left illustrates the locations of the detections A, B, C, D, E at$\{T_A, T_B, T_C, T_D, T_E\}$ in the taxonomy, with associated conditional relationships between them illustrated in the graph on the right.

A complete account of graphical architecture for probabilistic networks is available in [16]. However, we note here that our goal is optimal design of detections and not understanding the architecture of the Bayes network. Also of note is that our graph is between detections having varying levels of performance (Precision, Recall, FP, TP, FN, and TN), and this is our principal interest, and not probability distributions with associated marginals and conditionals.

Remark 1 (Isotonicity of gates and Detection Rule). Let $E(\lambda)$ be any gate event that is monotone in each parent (increasing for AND, decreasing for NOT). If the parent's precision increases while recall does not increase, then the induced change in $E(\lambda)$ preserves the stated monotonic direction for the child's precision/recall. Mixed gates with monotone Boolean formulas inherit these properties coordinatewise [17]. The internal rule of detection D, denoted by $R_D(\lambda_D)$, is simply saying that the detection's rule to fire (the threshold has been reached) has been activated; whether the parents set's arrangement is in concordance with the gating or not. This allows the the statement outlined above to follow as a logical operation; not a probabilistic one. [15]

3 Results

Assumptions. Let $\Lambda_D \subset \mathbb{R}$ be a nonempty, compact interval of admissible thresholds for detector D. For each fixed profile of other detectors' thresholds λ_{-D}, define the population

$$\mathrm{Prec}_D(\lambda_D\,;\,\lambda_{-D}) := \frac{\mathbb{P}(D_\lambda = P \mid Y = P)}{\mathbb{P}(D_\lambda = P)}, \quad \mathrm{Rec}_D(\lambda_D\,;\,\lambda_{-D}) := \mathbb{P}(D_\lambda = P \mid Y = P),$$

whenever the denominator is positive.

(A1) For every D, $\lambda_D \mapsto \mathrm{Rec}_D(\lambda_D\,;\,\lambda_{-D})$ is continuous on Λ_D and $\lambda_D \mapsto \mathrm{Prec}_D(\lambda_D\,;\,\lambda_{-D})$ is upper semicontinuous on Λ_D. (This allows piecewise-constant or stepwise detectors.)

(A2) For every D, there exists $\eta_D > 0$ such that $\mathbb{P}(D_\lambda = P) \geq \eta_D$ for all $\lambda_D \in \Lambda_D$ and all λ_{-D} considered, or else precision is defined on $\{\mathbb{P}(D_\lambda = P) > 0\}$ and optimized over this closed subset.

(A3) For any conditioning set $\mathcal{P}(D)$ of parents used operationally (gates defined below), $\mathbb{P}(\bigwedge_{C\in\mathcal{P}(D)} Y_{\lambda_C} = P) \geq \eta_{\mathcal{P}} > 0$ uniformly on the relevant action sets (or precision is evaluated conditionally on this event, with the same upper semicontinuity).

Theorem 1 (**Continuity/semi-continuity under gating**). Under (A1)–(A3) and fixed λ_{-D}, the map $\lambda_D \mapsto \mathrm{Rec}_{D^{\mathrm{op}}}(\lambda_D\,;\,\lambda_{-D})$ is continuous on Λ_D. Moreover, $\lambda_D \mapsto \mathrm{Prec}_{D^{\mathrm{op}}}(\lambda_D\,;\,\lambda_{-D})$ is upper semicontinuous on Λ_D, and continuous wherever $\mathbb{P}(D_\lambda^{\mathrm{op}} = P)$ stays bounded away from 0.

Proof. Proof is available in appendix section A.1 □

Theorem 2 (**Monotonicity of precision/recall under AND gating**). Fix a detector D with parents $\mathcal{P}(D)$ under AND gating. Suppose that for every parent $C \in \mathcal{P}(D)$ we move λ_C to weakly increase Prec_C (with recall not increasing). Then for fixed λ_D, $\mathrm{Prec}_{D^{\mathrm{op}}}$ weakly increases and $\mathrm{Rec}_{D^{\mathrm{op}}}$ weakly decreases.

Proposition 1 (**Monotonicity under NOT gating**). Under NOT gating, if Prec_C weakly increases (with recall not increasing) for each parent C, then $\mathrm{Prec}_{D^{\mathrm{op}}}$ weakly *decreases* and $\mathrm{Rec}_{D^{\mathrm{op}}}$ weakly *increases*.

Proof. Proofs available in appendix Section A.2. □

Note that the reciprocal is true for Recall as Precision and Recall have a monotonically counter-balanced relationship. To express this rigorously we include it as a proposition and prove it in the Appendix.

Theorem 3 (**Deterministic taxonomy idealization**). Assume a lossless taxonomy such that every child label implies its ancestor: $Y_n = P \Rightarrow Y_m = P$ for all ancestors $m \prec n$ (no noise). Then for any child Y_n gated by parent Y_m via AND,

$$\mathrm{Prec}_{C_n^{\mathrm{op}}} \geq \mathrm{Prec}_{D_m}, \qquad \mathrm{Rec}_{C_n^{\mathrm{op}}} \leq \mathrm{Rec}_{D_m}.$$

Theorem 4 (**Stochastic taxonomy, robustness**). If $\mathbb{P}(Y_m = P \mid C_n = P) \geq 1 - \epsilon$ for all thresholds in the action sets, then

$$\mathrm{Prec}_{C_n^{\mathrm{op}}} \geq \mathrm{Prec}_{D_m} - O(\epsilon), \qquad \mathrm{Rec}_{C_n^{\mathrm{op}}} \leq \mathrm{Rec}_{D_m} + O(\epsilon).$$

Proof. Proof available in appendix Section A.3. □

Theorem 5 (**Existence of pure-strategy equilibrium**). Let $\{\Lambda_D\}_{D=1}^N$ be nonempty, compact, convex intervals. For each D, define the utility

$$U_D(\lambda_D;\lambda_{-D}) = a_D\,\mathrm{Prec}_{D^{\mathrm{op}}}(\lambda_D;\lambda_{-D}) + b_D\,\mathrm{Rec}_{D^{\mathrm{op}}}(\lambda_D;\lambda_{-D}), \quad a_D, b_D \geq 0,\ (a_D, b_D) \neq (0,0).$$

Assume (A1)–(A3) and that for every fixed λ_{-D}, $U_D(\cdot;\lambda_{-D})$ is quasi-concave on Λ_D and U_D is continuous on $\prod_D \Lambda_D$.

Then this framework is a noncooperative game with an equilibrium point [18, 19, 20] where appropriate thresholds can be selected where no detection can be improved without other detections wanting to change their thresholds for improvement.

Proof. Proof available in appendix Section A.4. □

We also want to make a small note here that in scenarios where we're not working with precision and recall, but instead SNR, then if we are trying to maximize SNR, and the idealized version of SNR as a function of the threshold is quasi-concave, then the equilibrium point exists as well.

4 Simulation Study

4.1 Design of Experiments

4.2 Simulation Framework

The simulation experiments are designed to illustrate and validate the theoretical properties of *taxonomy-conditioned detection networks* introduced in the paper—particularly the continuity, monotonicity, and equilibrium results for detection thresholds under operational gating.

4.3 Graph and Taxonomy Structure

We represent the detection network as a *directed acyclic graph* (DAG) whose nodes correspond to detectors and whose edges reflect *operational conditioning* relationships. Parent–child relationships are instantiated according to a given taxonomy of cyber incidents, and each child's operational firing condition is determined by a gate type:

- **AND gate**: detector fires only if its internal rule is positive *and* all parents in the **AND** set fire.
- **NOT gate**: detector fires only if its internal rule is positive *and* all parents in the **NOT** set do not fire.

The **AND** and **NOT** sets are indicated by an indicator taking values of "+" and "-", respectively, in the code. An illustration of the DAG for the simulations is provided in Figure 6; with 10 edges corresponding to 8 **AND** gates and 2 **NOT** gates.

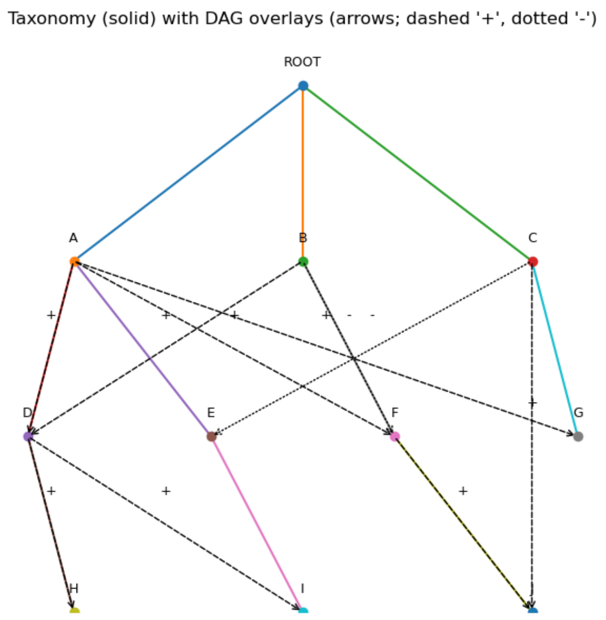
Taxonomy (solid) with DAG overlays (arrows; dashed '+', dotted '-')
ROOT
A
B
C
D
E
F
G
H
I
J
+
-

Figure 6: The taxonomy structure for the simulation study, of depth 4 leading off from a root node at the top. There are 12 total nodes in the tree, indexed by letters A-J and the ROOT node.

These gate semantics are enforced in the code base using the DAG utilities, outlined in `PerformanceGraphsClasss.py` in the code-base in Section 5, with cycle prevention and topological ordering. We don't model the gates directly as binary, but model them as *influence scores* from $[0, 1]$, where values on the boundary indicate hard rules. For the simulations the influence scores are set to .2 for both **AND** and **NOT** gates across all nodes. We also introduce a strength parameter indicating the interconnectedness of all edges in the network $\gamma > 0$, which in our simulations is generated from a $2 * \text{Beta}(1, 1)$ for all edges in the graph, seeded appropriately. This indicates high interdependence.

Specifically, γ is incorporating into the neighbor calculations for utility as an additive term in the utility function weighting the relationship of neighbors in the Moral Graph of the DAG ([16].

4.4 Generative Model for Detector Scores

Each detector is constructed out of a `TaxonomyNode` class object outlined in `TaxonomyNodeClass..py`, in the same linked code-base, and is endowed with a latent score distribution for positive and negative cases. In the current code base, these scores are indicated by a prevalence parameter between $[0, 1]$. This control the type 1 and 2 error for each λ value for the associated node. The *base rate* (prevalence of positives) is fixed per detector. Precision and recall are computed to be functions defined as a function of the base sensitivity and specificity at λ (which in this scenario comes from:

```
def clamp(x: float, lo: float = 0.0, hi: float = 1.0) -> float:
    return max(lo, min(hi, x))

def base_sensitivity(self, lam: float) -> float:
    return math.sqrt(clamp(1.0 - lam))

def base_specificity(self, lam: float) -> float:
    return math.sqrt(clamp(lam))
```

We then incpororate the DAG component, and a child in the DAG can be influenced by it's parents in the DAG as:

```
# --- Effective sensitivity/specificity with parental influence ---
    def effective_s_and_t(self, lam: float, neighbor_s: Dict[str, float], neighbor_t: Dict[str, float])
        s0 = self.base_sensitivity(lam)
        t0 = self.base_specificity(lam)

        if not self.in_edges:
            return s0, t0

        pos_par_s = [neighbor_s[nm] for nm, sgn in self.in_edges.items()\
                      if sgn > 0 and nm in neighbor_s]
        pos_par_t = [neighbor_t[nm] for nm, sgn in self.in_edges.items()\
                      if sgn > 0 and nm in neighbor_t]
        neg_par_s = [neighbor_s[nm] for nm, sgn in self.in_edges.items()\
                      if sgn < 0 and nm in neighbor_s]
        neg_par_t = [neighbor_t[nm] for nm, sgn in self.in_edges.items()\
                      if sgn < 0 and nm in neighbor_t]

        # Means centered around 0.5 to create monotone shifts consistent with Theorem 2 / Prop 1
        def centered_mean(xs):
            if not xs: return 0.0
            return sum(x - 0.5 for x in xs) / len(xs)

        s = s0 + self.pos_weight * centered_mean(pos_par_s) -\
              self.neg_weight * centered_mean(neg_par_s)
        t = t0 + self.pos_weight * centered_mean(pos_par_t) -\
              self.neg_weight * centered_mean(neg_par_t)

        return clamp(s), clamp(t)
```

The baseline sensitivity and specificity of a node are modified based on the ancestors/children in the Taxonomy.

This setup makes it possible to control:

- **Signal strength** between positives and negatives (affecting achievable precision/recall).

- **Interconnectedness** in the DAG (how much a parent's firing rate impacts a child's base rate through gating).
- **Noise** in parent–child taxonomy relationships (for stochastic-taxonomy variants).

4.5 Threshold Optimization Regimes

The simulations compare three optimization regimes for threshold selection:

1. **Single-pass staged optimization**: Thresholds are optimized in topological order of the DAG, with each node's threshold chosen to maximize its own utility
$$U_D = a_D \cdot \mathrm{Precision}_D + b_D \cdot \mathrm{Recall}_D$$
given its parents' fixed thresholds.
2. **Forward–backward sweep until convergence**: A local optimization procedure sweeps forward and backward through the DAG repeatedly, updating each node's threshold to its current best response until the threshold vector converges within a small tolerance.
3. **Global (equilibrium) search**: A coarse grid search over all thresholds in the action sets, selecting the profile that maximizes the joint objective (or meets the equilibrium conditions). In practice, this is tractable only for small networks due to combinatorial growth.

These methods correspond directly to the theoretical constructs: (1) single-pass staged optimization models a one-shot myopic strategy; (2) sweeps approximate best-response dynamics; (3) the global search stands in for the pure-strategy Nash equilibrium guaranteed by the existence theorem.

4.6 Evaluation Metrics

The primary outcome is each node's utility value under the three regimes, plotted side-by-side in **boxplots** across repeated random instantiations of the generative model. The simulations also record precision and recall separately to verify the monotonicity and continuity predictions:

- **Continuity**: sweeping thresholds produces smooth changes in measured metrics, consistent with Theorem 1.
- **Monotonicity**: for AND gates, increasing parent precision tends to increase child precision (and decrease recall), matching Theorem 2 and Proposition 1.
- **Hierarchy effect**: when gating respects the taxonomy structure, children under their parents show higher precision and lower recall in the deterministic-taxonomy limit (Theorem 3).

4.7 Repetition and Randomization

For each scenario, the simulation is repeated multiple times (with independent random seeds for score generation) to capture variability due to stochastic sampling. Boxplots aggregate these replicates, showing the distribution of utility improvements from local methods toward equilibrium.

4.8 Implementation Notes

- All DAG operations, gating, and staged/sweep solvers are implemented in `PerformanceGraphsClass.py`.
- Node attributes, including base rates and score distributions, are defined in `TaxonomyNodeClass.py`.
- Visualization code (boxplots, convergence traces) is contained in the main script or notebook.
- Parameters such as base rate, signal strength, and noise can be varied to explore sensitivity, though in the current code these are fixed in the main demonstration.

4.9 Nash Equilibrium Improvements

To illustrate the Nash equilibrium result in a simple scenario, assume that all paths downward from this tree at any depth (including only the root node) are local detection subsets indicating an attack strategy. This is in line with the outline previously presented; in the sense that modifying any particular attack strategy for optimal performance might affect the other attack strategy detections. We illustrate the three scenarios outlined in Section 4.5, first comparing regime (1) to (3), then comparing (2) to (3). The results are outlined in Figure 7 for comparison between single pass frozen

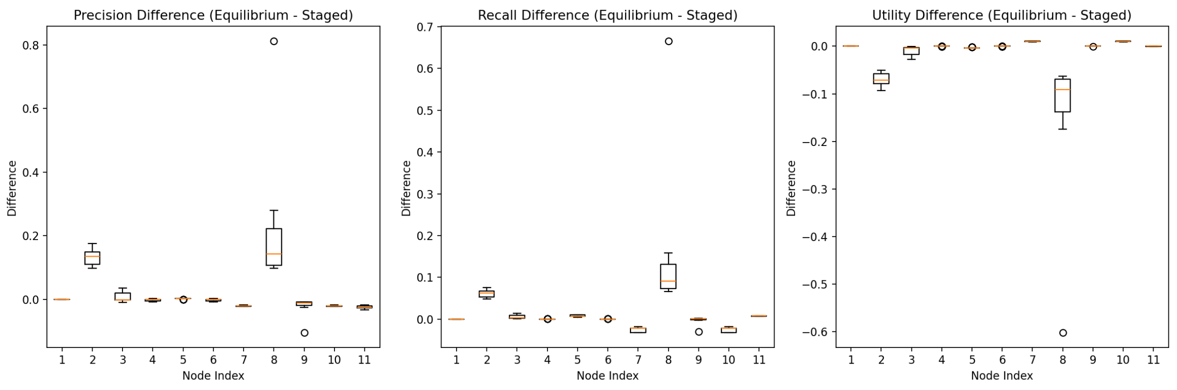

Figure 7: Boxplots showing the difference in Nash equilibrium and staged single pass frozen optimization. Precision, Recall, and Utility are shown for for $a = .4$ and $b = .6$. Values generated across 20 replications and varying values for the interconnectedness parameter in our model between [0,2] randomly generated uniformly.

optimization and the nash equilibrium, and in Figure 8 for the iterated forward-backward non-frozen pass. In our setting the criteria for convergence of the nonfrozen solver was maximum iteration of 1000 and a maximum $\Delta\lambda$ of 10^{-6}.

We can see that the equilibrium improves performance in some nodes dramatically, at a small trade off with other nodes for the single pass in Figure 7. Figure 8 illustrates another interesting dynamic. It appears that as we run he forward-backward pass over multiple iterations it slows diminishes the difference between the equilibrium and the single pass, converging to the Nash equilibrium. This is consistent with broader dynamic system theory which shows that under all actors behaving individually in their best interests the system converges to the Nash equilibrium given enough time [21].

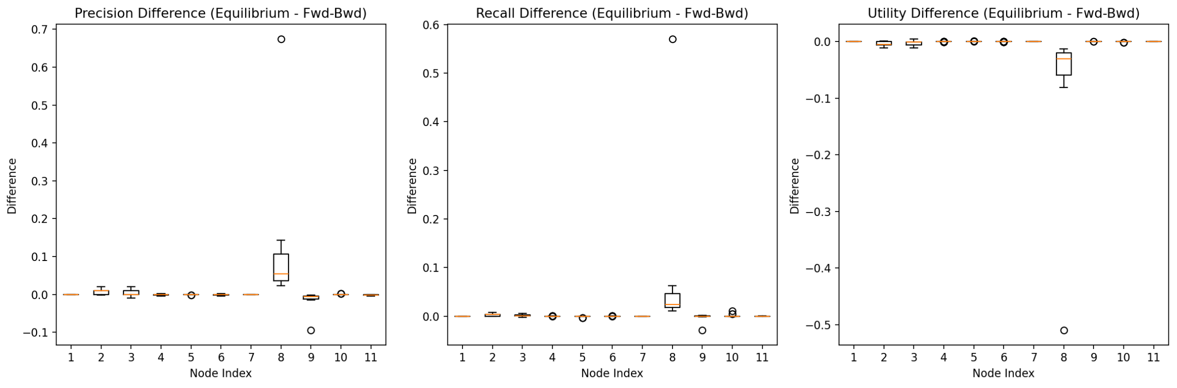

Figure 8: Boxplots showing the difference in Nash equilibrium and staged forward backward non-frozen optimization. Precision, Recall, and Utility are shown for for $a = .4$ and $b = .6$. Values generated across 20 replications and varying values for the interconnectedness parameter in our model between [0,2] randomly generated uniformly.

4.10 Illustration of the Results Through Simulations

We proceed to illustrate the other theorems with simulations. Figure 9 illustrates the smoothness of the precision surface with respect to the thresholding parameters. We can see that $Precision(H|D)(\lambda_H, \lambda_D)$ is continuous both in λ_H and λ_D. Additionally, this theorem illustrates Theorem 2. As $\lambda_D \to 1$ Precision$(D)(\lambda_D)$ increases, and so does Precision$(H|D)(\lambda_H, \lambda_D)$ across all values of λ_H. Heading in the opposite direction, as $\lambda_D \to 0$, we get the result for recall in Theorem 2 as well. The inverse relationship between precision and recall gives us the converse result for recall outlined in the same theorem.

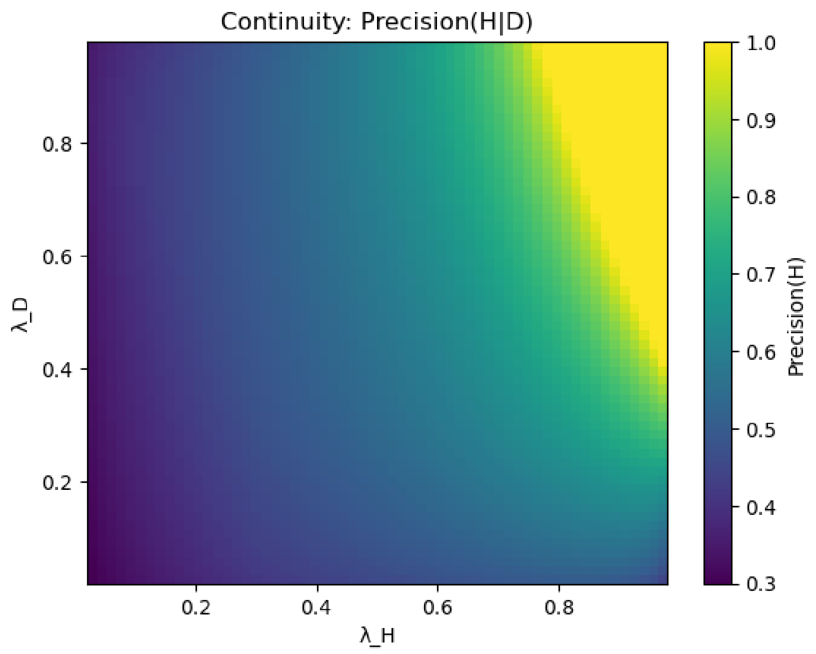

Figure 9: A figure illustrating the smoothness of a synthetic detection with respect to thresholding parameters

Figure 10 illustrates the deterministic taxonomy descendant conditioning outlined in Theorem 3.

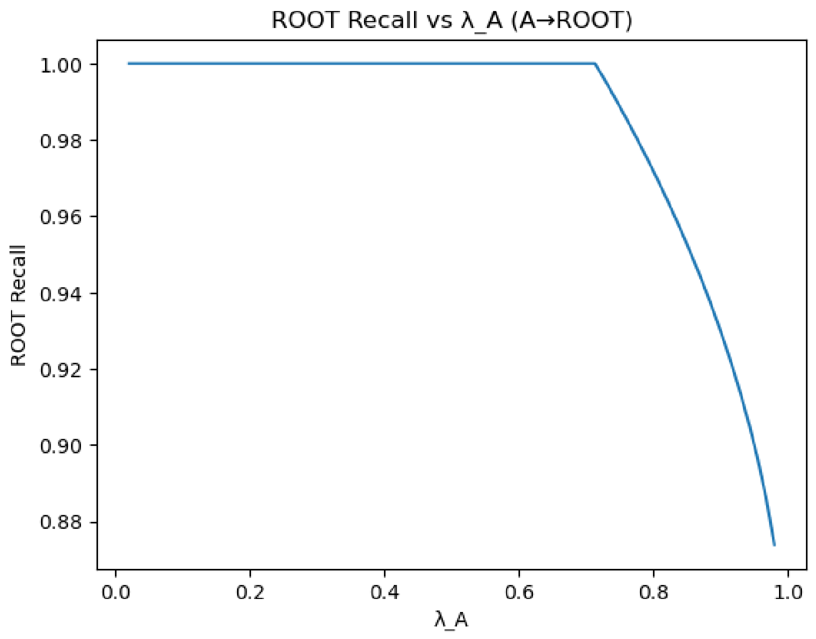

Figure 10: A figure illustrating Theorem 3. As the Precision of the child in the taxonomy (which the parent in the taxonomy is conditioned upon) increases, the recall of the parent in the taxonomy (which is the child in the DAG) goes down monotonically. The inverse is true in the opposite direction.

5 Discussion

This work has introduced a framework for integrating cyber incident taxonomies into networks of detections in order to optimize their joint performance under precision–recall trade-offs. By viewing each detection as a noncooperative agent with its own performance objectives, we linked the tuning of detection thresholds to game-theoretic equilibrium analysis.

Our key theoretical contributions are:

- *Smoothness and Monotonicity under gating* We formally established conditions under which precision and recall vary continuously with detection thresholds, and how gating relationships (e.g., AND, NOT) induce monotonic effects across parent–child detection pairs.
- *Taxonomy-Structured Performance Bounds* For deterministic taxonomies, conditioning a parent in the taxonomy on a descentant in the taxonomy results in changes in precision or recall; depending on the gating. We generalized this to stochastic taxonomies, quantifying robustness to noise.

  Existence of Nash Equilibrium in Threshold Selection We proved that under mild continuity and quasi-concavity conditions, there exists a pure-strategy Nash equilibrium in the threshold configuration space. This equilibrium represents a stable configuration where no detector can unilaterally improve its utility.

  From a practical perspective, the equilibrium result suggests that: Real-world detection networks can be tuned iteratively to approach optimal joint performance without requiring central coordination. In simulation, forward–backward non-frozen optimization approximates equilibrium performance over time, aligning with convergence results from dynamic systems theory
- *The simulation results illustrate that:*
  1. Single-pass local optimization underperforms compared to the global equilibrium.
  2. Iterative forward–backward optimization converges toward the equilibrium, validating the theoretical predictions.
  3. The shape of the precision–recall trade-off surfaces can guide practitioners in selecting threshold adjustments most likely to yield performance gains.
- *Future Directions*
  1. Extending the framework to multi-class taxonomies and multi-objective utility functions.
  2. Incorporating cost-sensitive metrics (e.g., weighted false positives/negatives).
  3. Exploring learning-based approaches for equilibrium estimation in large-scale detection graphs.

We believe that, by unifying taxonomy-based detection design with game-theoretic optimization, this work opens the door for more adaptive and resilient cyber defense systems that operate efficiently even in interconnected and dynamic environments.

Supplementary information

Code is available at the following github repository: `https://github.com/rswarnick1/Performance_Graphs`.

6 Acknowledgements

We would like to thank the Microsoft Security Research Leadership Team for permission to publish. Code was generated with the assistance of GPT5.

A Appendix

A.1 Proof of Theorem 1

Statement 1 (**Continuity/semi-continuity under gating**). Under (A1)–(A3) and fixed λ_{-D}, the map $\lambda_D \mapsto \mathrm{Rec}_{D^{\mathrm{op}}}(\lambda_D\,;\,\lambda_{-D})$ is continuous on Λ_D. Moreover, $\lambda_D \mapsto \mathrm{Prec}_{D^{\mathrm{op}}}(\lambda_D\,;\,\lambda_{-D})$ is upper semicontinuous on Λ_D, and continuous wherever $\mathbb{P}(D^{\mathrm{op}}_{\lambda} = P)$ stays bounded away from 0.

Proof. Both recall and the numerator of precision are expectations of bounded indicator functions whose arguments depend continuously on λ_D; the denominator inherits upper semicontinuity from dominated convergence. The ratio of a continuous (respectively upper semicontinuous) numerator and a denominator bounded away from 0 is continuous (respectively upper semicontinuous); on the constraint set where the denominator is positive, upper semicontinuity follows by standard closure arguments.

■. □

A.2 Proof of Theorem 2 and Proposition 1

Statement 2 (**Monotonicity of precision/recall under AND gating**). Fix a detector D with parents $\mathcal{P}(D)$ under AND gating. Suppose that for every parent $C \in \mathcal{P}(D)$ we move λ_C to weakly increase Prec_C (with recall not increasing). Then for fixed λ_D, $\mathrm{Prec}_{D^{\mathrm{op}}}$ weakly increases and $\mathrm{Rec}_{D^{\mathrm{op}}}$ weakly decreases.
Statement 3 (**Monotonicity under NOT gating**). Under NOT gating, if Prec_C weakly increases (with recall not increasing) for each parent C, then $\mathrm{Prec}_{D^{\mathrm{op}}}$ weakly *decreases* and $\mathrm{Rec}_{D^{\mathrm{op}}}$ weakly *increases*.

Proof. Treat the operational event as set intersections/unions and use inclusion relations of conditioning events; apply isotonicity of conditional probabilities ([17]) with respect to set containment, keeping λ_D fixed. ■ □

A.3 Proof of Theorem 3 and Theorem 4

Statement 4 (**Deterministic taxonomy idealization**). Assume a lossless taxonomy such that every child label implies its ancestor: $Y_n = P \Rightarrow Y_m = P$ for all ancestors $m \prec n$ (no noise). Then for any child D_n gated by parent D_m via AND,

$$\mathrm{Prec}_{C_n^{\mathrm{op}}} \geq \mathrm{Prec}_{D_m}, \qquad \mathrm{Rec}_{C_n^{\mathrm{op}}} \leq \mathrm{Rec}_{D_M}.$$

Statement 5 (**Stochastic taxonomy, robustness**). If $\mathbb{P}(Y_m = P \mid C_n = P) \geq 1 - \epsilon$ for all thresholds in the action sets, then

$$\mathrm{Precision}_{C_n^{\mathrm{op}}} \geq \mathrm{Precision}_{C_m} - O(\epsilon), \qquad \mathrm{Recall}_{C_n^{\mathrm{op}}} \leq \mathrm{Recall}_{C_m} + O(\epsilon).$$

We use the subscript n for thresholds, detections, and signals to indicate $\epsilon_1, \ldots, \epsilon_n$, and similarly m for $\epsilon_1, \ldots, \epsilon_m$.

Note that the event of a $Y_n = P$ in T_n implies an event of a P for Y_m in T_m with probability 1 .

Note from equation 5 that:

$$Precision(D_n) = \lim_{N \to \infty} \frac{TP(D_n)(\lambda_n)}{TP(D_n)(\lambda_n) + FP(D_n)(\lambda_n)} \tag{10}$$

we then have that:

$$Precision(D_n)(\lambda_n) > \frac{P(D_n = P, Y_n = P | Y_m = P, D_m = P)(\lambda_n, \lambda_m)}{P(D_n = P | D_m = P, Y_m = P)(\lambda_n, \lambda_m)} \tag{11}$$

$$Precision(D_n)(\lambda_n) > P(Y_n = P | Y_m = P, D_m = P, D_n = P)(\lambda_n, \lambda_m) \tag{12}$$

But note that $_m = P \Rightarrow Y_n = P$ with probability $1 - \epsilon$ logically, as T_n stochastically contains T_m with probability $1 - \epsilon$. This means that the probably on the right hand side of inequality 12 is at least $1 - \epsilon$. Using the law of total probability gives us $Precision(D_n)($

This gives us the correct lower bound for Theorem 4. Taking the limit as $\epsilon \to 0$ gives us the idealized case for Theorem3.

A.4 Proof of Theorem 5

Statement 6 (**Existence of pure-strategy equilibrium**). Let $\{\Lambda_D\}_{D=1}^{N}$ be nonempty, compact, convex intervals. For each D, define the utility

$$U_D(\lambda_D; \lambda_{-D}) = a_D \, \mathrm{Prec}_{D^{\mathrm{op}}}(\lambda_D; \lambda_{-D}) + b_D \, \mathrm{Rec}_{D^{\mathrm{op}}}(\lambda_D; \lambda_{-D}), \quad a_D, b_D \geq 0, \ (a_D, b_D) \neq (0, 0).$$

Assume (A1)–(A3) and that for every fixed λ_{-D}, $U_D(\cdot; \lambda_{-D})$ is quasi-concave on Λ_D and U_D is continuous on $\prod_D \Lambda_D$.

Then this framework is a noncooperative game with an equilibrium point [18, 19, 20] where appropriate thresholds can be selected where no detection can be improved without other detections wanting to change their thresholds for improvement.

Proof. We show that the detection threshold optimization problem is a noncooperative game satisfying the hypotheses of the Glicksberg extension of the Kakutani fixed-point theorem.

1. *Players and Strategy Sets*

   Each Detector D is a player.
   - The strategy set for D is Λ_D, which is nonempty, compact, and convex by assumption.
2. *Utility Functions*

   By (A1)-(A3), Precision_D^{op} is upper semi-continuous, and Recall_D^{op} is continuous on λ_D for fixed λ_{-D}
   - Therefore $U_D(\circ; \lambda_{-D)}$ is continuous and quasi concave on Λ_D.
3. *Best Response Correspondence*

   For each fixed λ_{-D}, define the best response set:

$$BR_D(\lambda_{-D}) = \arg\max_{\lambda_D \in \Lambda_D} U_D(\lambda_D; \lambda_{-D}) \tag{13}$$

   - Quasi-concavity of U_D ensure that $BR_D(\lambda_{-D})$ is convex valued and nonempty.
   - Continuity ensures that BR_D is upper hemicontinuous.
4. *Product Correspondence*

   Define $BR(\lambda) = \prod_{D \in \mathcal{D}} BR_D(\lambda_{-D})$
   - BR maps Λ into itself with nonempty, convex values and is upper hemicontinuous.
5. *Existence of Equilibrium*

   By Kakutani's fixed point theorem, there exists $\lambda^* \in \Lambda$ such that:

$$\lambda^* \in BR(\lambda^*) \tag{14}$$

   This λ^* is a pure strategy Nash equilibrium. [18, 19, 20].

■ □

References

[1] Daniel Ramsbrock, Robin Berthier, and Michel Cukier. Profiling attacker behavior following ssh compromises. In *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07)*, pages 119–124, 2007.

[2] Estefania Belen Vergara Cobos and Selcen Cakir. A review of the economic costs of cyber incidents, 2024.

[3] Hamed Taherdoost. Insights into cybercrime detection and response: A review of time factor. *Information*, 15(5), 2024.

[4] Ankur Mahida. Real-time incident response and remediation-a review paper. *Journal of Artificial Intelligence & Cloud Computing*, pages 1–3, 04 2023.

[5] Giovanni Rabitti, Amir Khorrami Chokami, Patrick Coyle, and Ruben D. Cohen. A taxonomy of cyber risk taxonomies. *Risk Analysis*, 45(2):376–386, 2025.

[6] Vasileios Mavroeidis and Siri Bromander. Cyber threat intelligence model: An evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence. *CoRR*, abs/2103.03530, 2021.

[7] A. Bruce Carlsen and Paul B. Crilly. *Communications Systems: An Introduction to Signals and Noise in Electrical Communication, Fifth Edition*. McGraw Hill, Avenue of the Americas, New York, NY, 10020, 1910.

[8] Cyril Cleverdon. *The Cranfield tests on index language devices*, page 47–59. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[9] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, 2nd edition, 1979.

[10] G. Canebk, Taskaya Temizel, and S. Sagiroglu. Ptopi: A comprehensive review, analysis, and knowledge representation of binary classification performance measures/metrics. *SN COMPUT. SCI.*, 4, 2023.

[11] The MITRE Corporation. MITRE ATT&CK®. `https://attack.mitre.org/`, 2025. August 9, 2025.

[12] John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, 1950.

[13] Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. Deep metric learning to rank. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6 2019.

[14] Judea Pearl. Probabilistic reasoning in intelligent systems: Networks of plausible inference. *Morgan Kaufmann Publishers*, 1988.

[15] René Thomas. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42(3):563–585, 1973.

[16] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press, 2009.

[17] David Angeli and Eduardo D. Sontag. Monotone control systems. *IEEE Transactions on Automatic Control*, 48(10):1684–1698, 2003.

[18] Gerard Debreu. A social equilibrium existence theorem. *Proceedings of the National Academy of Sciences*, 38(10):886–893, 1952.

[19] Irving L Glicksberg. A further generalization of the kakutani fixed point theorem, with application to nash equilibrium points. *Proceedings of the American Mathematical Society*, 3(1):170–174, 1952.

[20] Ky Fan. Minimax theorems. *Proceedings of the National Academy of Sciences*, 39(1):42–47, 1952.

[21] Zibo Xu. Convergence of best-response dynamics in extensive-form games. *Journal of Economic Theory*, 162:21–54, 2016.