Calibrated Trust in AI for Security Operations: A Conceptual Framework for Analyst–AI Collaboration

Israt Jahan Chowdhury * and Md Abu Yousuf Tanvir

Posted Date: 23 December 2025

doi: 10.20944/preprints202512.2112.v1

Keywords: trust in AI; security operations center; human–AI collaboration; explainable AI; uncertainty; automation bias; incident response; operational ML risk

 doi:10.20944/preprints202512.2112.v1

Article

Calibrated Trust in AI for Security Operations: A Conceptual Framework for Analyst–AI Collaboration Without Blind Automation

Israt Jahan Chowdhury * and Md Abu Yousuf Tanvir

Ontario Tech University, Canada
* Correspondence: isratjahan.chowdhury@ontariotechu.net

Abstract

AI is now embedded in Security Operations Centers (SOCs) through SIEM analytics, UEBA, EDR detections, phishing classifiers, and automated response playbooks. Yet operational value depends less on raw model accuracy and more on *calibrated reliance*—the analyst understands when to trust the system, when to challenge it, and how to recover when the model fails. This paper presents a conceptual framework for *Analyst–AI trust calibration* in SOC workflows. We unify insights from human–automation interaction, explainable and uncertainty-aware machine learning, and incident response practice to propose: (i) a failure-mode taxonomy for AI in SOC, (ii) an interaction model that ties explanations, uncertainty, and provenance to specific analyst decisions, (iii) a lightweight metric set for "trust calibration" (not just accuracy), and (iv) design principles and governance controls that reduce automation bias while preserving speed. The result is a practical blueprint that can be extended later with datasets, experiments, and user studies, but is immediately useful as a structured paper for secure deployment planning.

Keywords: trust in AI; security operations center; human–AI collaboration; explainable AI; uncertainty; automation bias; incident response; operational ML risk

1. Introduction

Security operations increasingly rely on AI-driven detections: anomaly detection in network telemetry, sequence modeling of authentication events, supervised classifiers for phishing and malware, and behavioral analytics that rank users, endpoints, and processes by risk. Commercial tools package these capabilities under SIEM/UEBA/EDR banners, but the analyst experiences them as a stream of alerts, scores, explanations, and recommended actions. In practice, the critical question is not "Is the model accurate?" but "*Can the analyst rely on it in this context?*"

A consistent body of evidence from human factors research shows that people can *over-trust* automation (automation bias) or *under-trust* it (disuse), especially under time pressure and cognitive load [1,2]. In SOC work, this becomes high stakes: over-trust can trigger unnecessary containment, service disruption, or escalation fatigue; under-trust can miss active intrusions. Because SOC environments are adversarial and non-stationary, models face concept drift, distribution shift, data-quality breaks, and deliberate evasion [5]. Trust must therefore be *calibrated* and continuously maintained.

Contributions. This paper contributes a conceptual framework that:

- defines *Analyst–AI trust calibration* for SOC decision points, grounded in human–automation interaction [1] and responsible AI guidance [3,4];
- proposes a taxonomy of failure modes and "trust ruptures" specific to SOC deployments;
- introduces an interaction model that maps *uncertainty* + *provenance* + *explanation* to analyst actions (triage, investigation, response, and post-incident learning);
- provides a metric set and evaluation blueprint that can be used *before* running full experiments, as a design and documentation scaffold.

 doi:10.20944/preprints202512.2112.v1

2. Background: Trust, Calibration, and Automation Bias

2.1. Trust as a Calibrated, Task-Specific Relationship

In automation research, trust is commonly framed as the operator's willingness to rely on an automated aid under uncertainty [1]. Crucially, trust is not a personality trait to be "maximized": it is *task- and context-dependent*. In SOC settings, trust should be *bounded* by the system's demonstrated competence, the evidence it provides, and the operational cost of errors. Miscalibration arises when perceived reliability diverges from actual reliability.

2.2. Why SOC Conditions Amplify Miscalibration

SOCs are vulnerable to trust miscalibration because:

- **Alert overload** encourages shortcut decisions and deference to scores;
- **Asymmetric costs** (false negatives can be catastrophic; false positives erode attention);
- **Adversarial pressure** enables evasion and poisoning attempts;
- **Non-stationarity** causes drift in user behavior, infrastructure, and attacker tactics [5];
- **Tool heterogeneity** means analysts fuse evidence from SIEM, EDR, NDR, IAM logs, and threat intel feeds, each with different quality.

2.3. Responsible AI Expectations in Security Tooling

High-level AI governance principles (transparency, accountability, robustness, human oversight) appear across policy and ethics work [3,4]. In SOC tooling, these principles must translate into concrete controls: auditability of detections, explanation appropriate to the decision, explicit uncertainty, and safe automation boundaries.

3. Problem Framing: Trust Failures in SOC AI

We define a **trust rupture** as a moment when the Analyst–AI relationship becomes unreliable for decision-making due to model error, missing context, misleading explanation, or compromised data. Table 1 summarizes common SOC-relevant failure modes.

Table 1. Failure-mode taxonomy for AI in SOC workflows (conceptual).

Category	What it looks like	Trust impact / typical consequence
Data quality break	Missing fields, time skew, duplicated events, inconsistent enrichment	False spikes/drops; analysts lose faith in the system; silent false negatives
Distribution shift	New software rollouts, IAM policy changes, remote work patterns	Score meaning changes; explanation becomes misleading; increased false positives
Adversarial evasion	Living-off-the-land, benign-appearing behavior, log manipulation	Over-trust leads to missed intrusion; model confidence becomes uninformative
Label leakage / proxy learning	Model learns environment-specific shortcuts (e.g., "admin hosts = malicious")	High offline accuracy but poor generalization; brittle trust
Pipeline / deployment mismatch	Training features differ from production features; version skew	Unexpected behavior; hard-to-debug incidents; trust collapse after surprises
Automation bias	Analyst defers to AI recommendation despite contrary evidence	Premature containment/escalation; reduced investigative rigor [2]

4. Threat Model: Adversaries Targeting Analyst–AI Trust

Unlike many enterprise AI deployments, SOC AI operates against adaptive adversaries. Attackers do not merely attempt to evade a detector; they can also attempt to *manipulate trust* by shaping what analysts see and believe. A practical trust framework must therefore include an explicit threat model.

 doi:10.20944/preprints202512.2112.v1

4.1. Adversary Goals

We consider three broad attacker goals:

- **Evasion:** perform malicious actions while keeping AI scores low and explanations benign.
- **Disruption:** create alert storms and false positives to exhaust analysts and reduce trust.
- **Trust hijacking:** induce analysts to over-rely on AI recommendations (or to ignore them) at critical moments.

4.2. Attack Surfaces

Trust-related attack surfaces include:

- **Telemetry manipulation:** log tampering, time skew, or selective suppression of events.
- **Feature and pipeline fragility:** exploiting brittle proxies the model learned (environment-specific shortcuts).
- **Poisoning and feedback loops:** influencing labels through staged "benign" outcomes or noisy tickets.
- **Explanation gaming:** triggering feature patterns that yield comforting explanations even when behavior is malicious.

Table 2. Adversarial strategies that influence trust (conceptual) and corresponding defenses.

Strategy	How it breaks trust	Defensive control
Alert flooding	Analysts learn to ignore alerts; disuse becomes rational	Rate-limit, deduplicate, and surface "new pattern" alerts separately
Low-and-slow evasion	Model sees weak signals; confidence appears high due to proxy cues	Uncertainty triggers, abstention, and cross-source corroboration
Telemetry gaps	False negatives; analysts assume "no news is good news"	Integrity checks, missing-data flags, fallback rules
Poisoned feedback	Retraining reinforces attacker-shaped labels	Label provenance, gated updates, anomaly checks on training data
Explanation mimicry	Explanations appear benign even for malicious activity	Adversarial testing of explanations; multi-view evidence packages

This threat model reinforces a key design rule: *trust signals must be robust to manipulation*. Provenance metadata, multi-source corroboration, and conservative automation boundaries are not just governance preferences—they are security requirements.

5. Analyst–AI Trust Interaction Model

This section proposes a workflow-aligned interaction model: trust is calibrated at each decision point by presenting the right *evidence package* (signals, uncertainty, provenance, and explanation), and by constraining automation to safe operating envelopes.

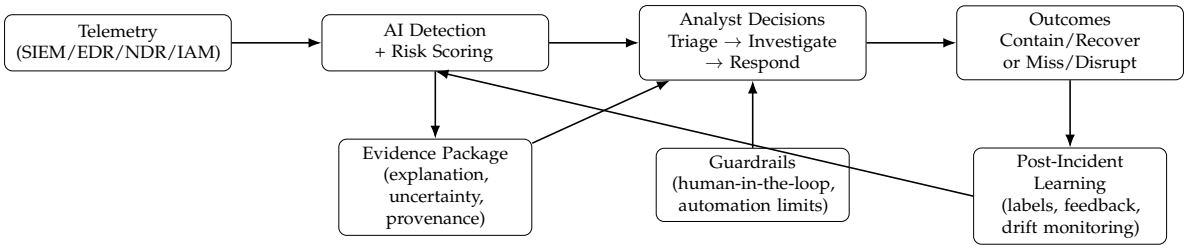

Figure 1. Analyst–AI trust interaction model for SOC workflows. The key design object is the *evidence package* that enables calibrated reliance at each decision point.

 doi:10.20944/preprints202512.2112.v1

5.1. Evidence Package: What the Analyst Needs at the Moment of Decision

The "evidence package" is not a generic explanation blob. It is a structured bundle tailored to a decision:

- **Local explanation:** which signals/features drove the score (appropriate abstraction level);
- **Uncertainty:** confidence intervals, calibration cues, or "insufficient evidence" flags;
- **Provenance:** data sources, enrichment lineage, model version, and time window;
- **Counter-evidence prompts:** checks the analyst should perform ("verify process tree", "confirm sign-in location");
- **Action safety:** recommended next steps with blast-radius hints (e.g., isolate host vs. observe).

This aligns with the view that explanations should support *actionable understanding* rather than post-hoc justification [6].

5.2. Guardrails: Preventing Blind Automation

Guardrails operationalize human oversight:

- require **explicit analyst confirmation** for high-impact actions;
- implement **rate limits** on automated containment to avoid cascading outages;
- ensure **fallback modes** (rule-based or conservative thresholds) during telemetry outages;
- record **audit trails** for every automated suggestion and analyst override.

6. SOC Workflow Mapping: Where Trust Decisions Actually Happen

To make "trust" operational, we identify concrete decision points where the analyst either relies on the AI output, seeks additional evidence, or overrides it. A SOC pipeline can be simplified into four stages: *triage*, *investigation*, *response*, and *post-incident learning*. Each stage has distinct time budgets, evidence needs, and error costs.

6.1. Triage: Fast Ranking Under Uncertainty

Triage is dominated by speed and prioritization: the analyst decides whether an alert is benign, needs enrichment, or warrants escalation. Trust failures here often manifest as *alert fatigue* (too many false positives) or *silent miss* (model suppresses important events). Therefore, triage explanations must be short and diagnostic: "why this is high-risk" and "what single check can falsify it".

6.2. Investigation: Causal Reconstruction and Hypothesis Testing

Investigation requires stitching together evidence across sources: process lineage, network flows, identity events, asset context, and threat intelligence. The AI should support hypothesis generation and evidence navigation (e.g., link analysis, sequence summaries), but must not replace adversarial reasoning. At this stage, provenance and time windows are critical: analysts need to know *what data the model actually saw* and what might be missing.

6.3. Response: High-Impact Actions with Blast Radius

Response actions (isolation, account disablement, block rules) can disrupt business operations. Trust must be conservative: if uncertainty is high, recommend observation or containment with limited scope. Automation at this stage should be tiered and rate-limited, with explicit approval for high-impact actions.

6.4. Post-Incident Learning: Closing the Loop Without Contaminating Labels

After containment and recovery, SOC teams create tickets, reports, and lessons learned. These artifacts are tempting to treat as "labels" for retraining. However, labels can be noisy or biased (e.g., only obvious incidents get labeled). A trust-aware pipeline must track label provenance and quality, and avoid feedback loops that reinforce wrong patterns.

 doi:10.20944/preprints202512.2112.v1

Table 3. Evidence package requirements by SOC decision point (conceptual checklist).

Stage	Minimum evidence package	Automation boundary
Triage	Top drivers, quick falsification check, uncertainty flag, source list	Assist/recommend; avoid auto-close unless confidence is high
Investigation	Provenance, timeline, correlated entities, explanation at multiple depths	Assist; semi-automated enrichment and graphing are safe
Response	Impact estimate, blast radius, alternative actions, required confirmations	Automate only low-risk actions; approvals for high-impact steps
Learning	Outcome summary, label confidence, drift notes, model/version references	No autonomous retraining; gated updates and audits

7. Trust Calibration Metrics (Beyond Accuracy)

Classic metrics (precision/recall) are necessary but insufficient. Trust calibration is about whether confidence and explanations *track reality* under operational conditions. We propose a lightweight metric set usable in planning and early validation:

- **Calibration error:** do predicted probabilities match empirical outcomes? (e.g., ECE) [7];
- **Selective prediction utility:** performance when the model is allowed to abstain under uncertainty;
- **Evidence adequacy:** analyst-rated usefulness of explanations for the specific decision (triage vs. response);
- **Override rate and rationale:** frequency and reasons analysts reject recommendations (signal of misfit);
- **Time-to-decision and error cost:** speed improvements without increasing harmful errors.

8. Design Principles for SOC-Ready Trustworthy AI

We translate principles into implementable guidance:

8.1. Principle 1: Make Uncertainty Explicit and Actionable

Instead of a single score, provide:

- confidence bands or calibrated probabilities;
- uncertainty categories (“data missing”, “out-of-distribution”, “conflicting signals”);
- abstention pathways (route to manual triage or request more telemetry).

8.2. Principle 2: Match Explanation to the Analyst Task

Triage needs short, high-signal cues; investigation needs deep provenance (process trees, sequence context), while post-incident learning needs aggregated patterns and root-cause factors. Model-agnostic explainability methods (e.g., LIME/SHAP) are useful but must be applied carefully to avoid false confidence [6].

8.3. Principle 3: Engineer for Drift and Operational ML Risk

SOCs should treat ML as a production system subject to technical debt [5]:

- drift monitors on input distributions and alert rates;
- canary deployments and rollback plans;
- model cards / change logs accessible to analysts;
- periodic red-team evaluation for evasion resilience.

 doi:10.20944/preprints202512.2112.v1

8.4. Principle 4: Encode Safe Automation Boundtaries

Use tiered automation:

- **Tier 0 (assist):** summarize evidence and propose queries;
- **Tier 1 (recommend):** suggest actions with impact estimates;
- **Tier 2 (automate):** only for low-risk actions with tight constraints.

9. Blueprint for an Evaluation (Without Running Experiments Yet)

Even as a conceptual draft, a credible paper should state how claims can be validated. We propose a three-layer evaluation plan:

1. **Offline validation**: use benchmark datasets (e.g., NSL-KDD, UNSW-NB15, CICIDS2017) to establish baseline detection behavior [8–10];
2. **Operational simulation**: replay real SOC log sequences with injected drift and telemetry faults to test robustness and guardrails;
3. **Human-in-the-loop study**: measure calibration (ECE), decision speed, and harmful error rate under time pressure [2,7].

10. Worked Examples (Conceptual)

10.1. Phishing-to-Account-Takeover

Consider an email phishing alert that links to suspicious OAuth consent. A trust-calibrated system would:

- show provenance (email gateway verdict + identity logs + device posture);
- provide uncertainty flags if mailbox telemetry is incomplete or delayed;
- explain top signals (unusual sender infrastructure, new device, abnormal token scope);
- recommend safe actions (password reset, token revocation, conditional access) with explicit confirmation.

The key is that the analyst can quickly test the model's hypothesis: "Is there corroborating sign-in risk from the IdP?" "Did the user actually approve consent?"

10.2. Endpoint Anomaly During Patch Windows

Suppose an endpoint model flags suspicious PowerShell activity on many hosts shortly after a legitimate software deployment. A naive system creates an alert flood and destroys trust. A calibrated system recognizes distribution shift cues: synchronized time window, shared parent process, signed installer lineage, and known change ticket. The evidence package should surface these contextual anchors and may lower severity with a prominent "change window likely" tag, while still preserving a path to investigate outliers (e.g., hosts that executed additional network beacons). This example illustrates why provenance and environment context are first-class trust signals.

10.3. Insider-like Behavior with Ambiguous Intent

UEBA systems often flag "impossible travel," abnormal access, or unusual data downloads. These events can be malicious or benign (travel, VPN, new project). For ambiguous cases, uncertainty should trigger *graduated response*: additional verification (manager confirmation, MFA re-check), step-up authentication, or temporary scope reduction rather than immediate account lockout. Here, calibrated trust means the system supports proportionality: it helps the analyst respond safely even when the model cannot know intent.

11. Governance and Documentation

To align with trustworthy AI expectations [3,4], SOC deployments should publish internal documentation (even if not public):

- model purpose, limitations, and known failure modes;

 doi:10.20944/preprints202512.2112.v1

- data lineage and retention constraints;
- incident playbooks describing when to override AI;
- audit policy for automated actions.

12. Related Work and Positioning

Our framing connects three lines of work.

Human–automation interaction. Research on misuse/disuse of automation highlights how operators form mental models of automated aids and how interface cues influence reliance [1,2]. In SOC contexts, these effects are amplified by workload and time pressure, making "trust calibration" a more practical goal than simply increasing adoption.

Interpretable and uncertainty-aware ML. Interpretable ML argues for explanations that support understanding and accountability [6]. Calibration work shows that modern neural networks can be poorly calibrated even when accurate, motivating explicit calibration metrics and uncertainty mechanisms [7]. For SOC systems, uncertainty must be *actionable* (e.g., abstain, request telemetry) rather than a numeric decoration.

Operational ML and security datasets. ML-in-production introduces hidden technical debt, monitoring needs, and dependency risk [5]. In security analytics, benchmark datasets are widely used for baseline experimentation, but their limitations and environment mismatch reinforce the need for deployment-aware evaluation and documentation [8–10].

13. Trust Repair: What to Do After the Model Is Wrong

Trust is dynamic. In SOCs, a single high-profile failure can cause long-lasting disuse. We propose a "trust repair" loop that treats failures as operational incidents:

1. **Detect the rupture:** spikes in override rate, sudden alert distribution changes, or analyst feedback indicating "nonsense" explanations.
2. **Triage the root cause:** separate data-quality breaks from model drift and from attacker evasion. Provenance metadata accelerates this.
3. **Mitigate safely:** tighten automation boundaries, enable conservative thresholds, or fall back to rules while maintaining visibility.
4. **Communicate clearly:** publish a short internal incident note describing what happened, scope, and temporary guidance for analysts.
5. **Verify the fix:** run a regression suite, replay recent traffic, and validate calibration before restoring automation.

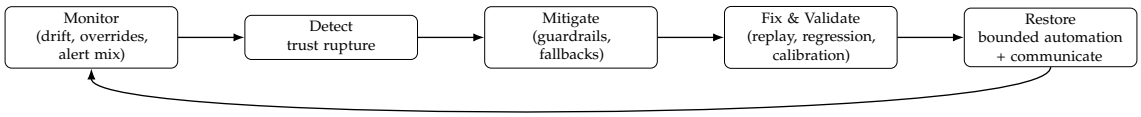

Figure 2. Trust repair loop after a model failure. Treating trust ruptures like incidents helps prevent long-term disuse and unsafe over-corrections.

14. Practical Implementation Checklist (SOC-Ready)

For teams adopting AI in SOCs, the following checklist turns the framework into concrete actions:

14.1. Interface and Analyst Experience

- Provide a one-line "why" plus a one-line "what to check next" for every high-severity alert.
- Expose uncertainty categories (missing telemetry, out-of-distribution, conflicting signals) instead of only a confidence score.
- Show model/version and feature availability so analysts can reason about what the system could have seen.
- Record analyst overrides with short reason codes to create a measurable feedback channel.

 doi:10.20944/preprints202512.2112.v1

14.2. Engineering and Monitoring

- Track input distribution drift and alert-rate drift; alert on sudden shifts.
- Maintain a regression suite of representative incidents and benign activity; replay before every model update.
- Implement canary releases and rollbacks for model updates, the same way you would for high-risk code.
- Maintain secure logging and integrity checks to reduce the chance of telemetry manipulation.

14.3. Governance and Documentation

- Publish a "model card" for SOC users: purpose, limitations, failure modes, and safe usage guidance.
- Define which actions can be automated at each confidence/uncertainty level and enforce this in tooling.
- Ensure auditability: who saw what, what the model recommended, what the analyst did, and why.

15. Scope for Future Empirical Work

This working draft can be extended with: (i) calibration and abstention experiments under synthetic drift, (ii) red-team evaluations for evasion resilience, (iii) SOC analyst user studies measuring decision quality and workload, and (iv) comparisons between explanation designs (minimal cues vs. deep provenance views).

16. Limitations

This paper is conceptual: it does not report new experimental results or user-study statistics. Nevertheless, the framework is designed to be testable, extensible, and immediately useful for engineering and governance planning. Future work should include empirical measurement of calibration and decision outcomes across SOC tiers and tool stacks, and should evaluate resilience against adversarial evasion.

17. Conclusion

Trustworthy AI in security operations is not achieved by maximizing automation, but by building systems that enable *calibrated reliance*. We provided a SOC-specific failure taxonomy, an Analyst–AI interaction model, a practical metric set, and implementation principles with governance controls. These components provide a structured foundation for calibrated Analyst, AI collaboration in SOCs and can be empirically validated and extended in future work.

References

1. Raja Parasuraman and Victor Riley. "Humans and automation: Use, misuse, disuse, abuse." *Human Factors*, 39(2):230–253, 1997.
2. Kathleen L. Mosier and Linda J. Skitka. "Human decision makers and automated decision aids: Made for each other?" In *Automation and Human Performance: Theory and Applications*, 1996.
3. Luciano Floridi et al. "AI4People—An ethical framework for a good AI society." *Minds and Machines*, 28:689–707, 2018.
4. European Commission High-Level Expert Group on AI. *Ethics Guidelines for Trustworthy AI*. 2019. https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai
5. D. Sculley, G. Holt, D. Golovin, et al. "Hidden technical debt in machine learning systems." In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems
6. Finale Doshi-Velez and Been Kim. "Towards a rigorous science of interpretable machine learning." arXiv:1702.08608, 2017. https://arxiv.org/abs/1702.08608

 doi:10.20944/preprints202512.2112.v1

7. Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. "On calibration of modern neural networks." In *Proceedings of ICML*, 2017. https://arxiv.org/abs/1706.04599
8. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani. "A detailed analysis of the KDD CUP 99 data set." In *Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009.
9. N. Moustafa and J. Slay. "UNSW-NB15: A comprehensive data set for network intrusion detection systems." In *Proceedings of the Military Communications and Information Systems Conference (MilCIS)*, 2015.
10. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani. "Toward generating a new intrusion detection dataset and intrusion traffic characterization." In *Proceedings of the International Conference on Information Systems Security and Privacy (ICISSP)*, 2018.