From Rogue Employees to Rogue Agents: Repurposing Insider Threat Detection for AI Agent Governance

Bipin Rimal
Department of Computing
Coventry University
Coventry, United Kingdom
rimalb@uni.coventry.ac.uk

Abstract*—AI agent monitoring and insider threat detection share the same architecture: profile a baseline, flag deviations, encode assumptions about trust. We make this concrete across thirteen experiments. Using a Unified Behavioural Feature Schema (UBFS) that maps both employee activity logs and agent execution traces into a shared representation, we apply three anomaly detection models—Isolation Forest, LSTM Autoencoder, and Deep Clustering—across five domains. Cross-domain transfer works: an Isolation Forest trained on 329,000 insider threat user-days retains 97% of detection power on agent traces, and transfer to MCP tool-calling benchmarks exceeds within-domain performance (104.8% retention). But the blind spots transfer too. Model distillation attacks span a detection spectrum from trivially caught (FOCUSED extraction: 0.99 AUC-ROC) to near-chance (HYDRA distributed: 0.54), and task decomposition—splitting a malicious objective into innocent subtasks—costs 5–6% detection power for IF and DC (p=0.031, Wilcoxon signed-rank), while LSTM Autoencoder shows near-invariance (-0.7%, p=0.094). Synthetic OWASP profiling identifies Tool Misuse (ASI02) as a blind spot ($\sim$0.52 AUC-ROC), but real-data validation on 500 ATBench trajectories reveals that this blind spot is an artifact of circular synthetic methodology: real ASI02 achieves 0.81–0.94 AUC-ROC across all models and feature sets. Augmenting UBFS with eight semantic features (UBFS-28) improves detection by up to 13% on real data. Distillation sensitivity analysis shows detection degrades monotonically with attack intensity for FOCUSED/BROAD/COT profiles, while HYDRA remains near-chance regardless of scale. Adversarial evasion testing reveals that mimicry attacks degrade detection by up to 25%, with Memory Poisoning (ASI05) most vulnerable. Temporal window ablation finds that 10 spans are optimal for Isolation Forest on both TRAIL and ATBench. Mapping five MITRE ATLAS techniques [1] into the UBFS confirms that the framework generalises beyond OWASP: all five techniques are strongly detectable (>0.93 AUC-ROC for IF/LSTM), introducing no new blind spots. The detection models port across domains. So do their biases, and so do their blind spots.

***Index Terms*—AI agent monitoring, insider threat detection, anomaly detection, cross-domain transfer, behavioural feature schema, AI governance, OWASP, MITRE ATLAS, adversarial evasion, model distillation**

I. Introduction

An employee badges into a secure facility at 2 AM, copies files to a USB drive, and leaves. An AI agent invokes a privileged API at 2 AM, exfiltrates training data through a tool call, and terminates. The first is an insider threat. The second is a rogue agent. The monitoring systems that would detect them are architecturally identical.

Both systems profile a baseline of "normal" behaviour—one for employees, one for agents. Both flag deviations from that baseline as anomalous. Both encode assumptions about trust, authority, and what counts as legitimate action. Yet the two research communities have developed largely in isolation: the security community builds User and Entity Behaviour Analytics (UEBA) systems [2]; the AI safety community builds agent monitoring frameworks from scratch [3].

This separation is costly. The security community has spent two decades learning which behavioural features predict insider threats, which attack patterns evade detection, and which modelling assumptions introduce blind spots. The AI safety community is learning these same lessons from scratch—different conferences, different datasets, different terminology for the same underlying problem. If the structural analogy holds, this duplication is unnecessary. Worse, it means the agent monitoring community will rediscover the same blind spots the security community has already documented.

This paper tests whether the structural analogy is more than a metaphor. We take three anomaly detection models validated on the CMU-CERT insider threat dataset [4], apply them to AI agent execution traces through a unified feature schema, and measure what transfers, what breaks, and what biases come along for the ride.

The results are stronger than expected in one direction, and more concerning than expected in another. Cross-domain transfer works—an insider threat model retains 97% of its detection power on agent traces, and transfer to MCP tool-calling benchmarks exceeds within-domain performance. But the blind spots transfer too. Tool Misuse (ASI02), where an agent uses the right tools in the right order with escalated privileges, is structurally undetectable for the same reason that insider "Scenario 2" attacks (quiet data exfiltration via legitimate channels) evade detection in [4]. Real-world validation arrived in February 2026: Anthropic disclosed that model distillation attacks and AI-orchestrated espionage campaigns exploited exactly the structural blind spots our framework

predicted [5]. And augmenting UBFS with semantic features—the obvious fix—closes the gap for four OWASP categories but fails to move synthetic ASI02 above chance, though real-data validation reveals this blind spot is partially an artifact of circular synthetic methodology.

A. Research Questions

- **RQ1**: Can insider threat detection models detect anomalous agent behaviour through a shared behavioural feature schema?
- **RQ2**: Which OWASP Agentic Security Index categories are detectable by structural anomaly detection, and which are blind spots?
- **RQ3**: What governance assumptions embedded in insider threat models transfer to the agent monitoring domain?
- **RQ4**: Do emerging real-world attacks exploit the predicted blind spots, and can semantic feature augmentation close them?

B. Contributions

1) **UBFS**: A 20-dimensional Unified Behavioural Feature Schema mapping both insider threat indicators and agent trace signals into a shared representation across 7 categories.
2) **Transfer evidence**: Empirical demonstration that an Isolation Forest trained on 329,000 insider user-days retains 97% of detection power on agent traces (AUC-ROC: 0.731 $\rightarrow$ 0.711).
3) **OWASP detection matrix**: A per-category analysis showing three detection tiers—strong (ASI05: 0.975), moderate (ASI09, ASI10: 0.70–0.86), and blind spot (ASI02: 0.51–0.52).
4) **Governance audit**: Identification of six normality assumptions that transfer across domains, with analysis of their implications for monitoring fairness.
5) **Distillation detection spectrum**: Four distillation subprofiles spanning AUC-ROC 0.54 (HYDRA distributed) to 0.99 (FOCUSED extraction), benchmarked against ASI02 (0.50), revealing two distinct blind-spot failure modes—intent-based and coordination-based.
6) **Decomposition evasion**: Quantified evidence that splitting malicious objectives into innocent subtasks costs 5–6% detection power for IF and DC (p=0.031), while LSTM shows near-invariance (-0.7%, p=0.094), mirroring insider threat "boiling frog" escalation.
7) **MCP transfer generalisation**: Cross-domain transfer validated on ATBench tool-calling traces, where TRAIL$\rightarrow$ATBench Isolation Forest achieves 104.8% of within-domain performance.
8) **UBFS-28 hybrid**: Semantic augmentation improves four OWASP categories on synthetic data but leaves synthetic ASI02 at $\sim$0.56 AUC-ROC; however, real-data validation (Experiment 12) shows the synthetic blind spot is an artifact.
9) **Real-data validation**: 500 ATBench trajectories reveal that synthetic OWASP profiling dramatically underestimates real detection performance (ASI02: 0.52 synthetic vs. 0.81–0.94 real), and UBFS-28 improves most categories by up to 13% on real data.
10) **Distillation sensitivity**: Feature-space scaling across 4 intensity levels shows monotonic detection degradation for FOCUSED/BROAD/COT profiles, while HYDRA remains near-chance ($\sim$0.52) at original intensity regardless of scale.
11) **Adversarial evasion framework**: Three evasion strategies—feature clamping, gradual escalation, and mimicry—quantifying the robustness of UBFS-based detection. Mimicry is the most effective, degrading ASI05 detection by up to 25%.
12) **Temporal window transfer**: Window ablation showing 10 spans optimal for Isolation Forest on both TRAIL and ATBench, paralleling Paper 1's 7-day optimal window—a structural insight about the primacy of early warning signals.
13) **MITRE ATLAS mapping**: Five ATLAS adversarial ML techniques mapped to UBFS perturbation profiles, all landing in the "strong" detection tier (>0.93 AUC-ROC for IF/LSTM)—no new blind spots beyond OWASP ASI02.

II. Related Work

Insider threat detection has evolved from rule-based systems through supervised classification to behavioural analytics [2], [6]. Each generation hits the same bottleneck: labeled data. Unsupervised methods sidestep this by modelling normal behaviour and flagging deviations [7]. Isolation Forest [8] exploits the geometric property that outliers require fewer random splits to isolate. LSTM autoencoders [9] learn temporal dependencies in behavioural sequences. Deep clustering [10] combines representation learning with cluster structure. Tuor et al. [11] applied deep learning to structured cybersecurity streams; Cappelli et al. [12] established the practitioner framework for insider risk.

AI agent monitoring is a younger field. The OWASP Top 10 for Agentic Applications [3] taxonomises risks from prompt injection to excessive agency. The TRAIL dataset [13] provides annotated agent execution traces with error labels. The TRACE dataset [14] captures reward hacking in iterative self-refinement—agents that produce correct outputs through illegitimate means. Pan et al. [15] formalised reward misspecification as a safety concern.

Real-world AI agent attacks. In February 2026, Anthropic disclosed two categories of attack against Claude [5]: model distillation (three Chinese AI labs created $\sim$24,000 fraudulent accounts and conducted 16M+ exchanges to extract capabilities) and AI-orchestrated espionage (state-sponsored group GTG-1002 used Claude Code with MCP tools as an autonomous hacking agent). The distillation attacks employed three distinct strategies—focused extraction, multi-capability probing, and chain-of-thought elicitation—plus a "hydra cluster" architecture distributing traffic across 20,000+ accounts. The ATBench benchmark [16] provides multi-turn, tool-centric

safety scenarios with fine-grained risk categorisation, enabling evaluation on MCP-style traces. The AgentHarm dataset [17] provides 440 malicious agent tasks in both direct and decomposed configurations, enabling comparison of single-turn attacks against multi-step task decomposition—the same evasion strategy used in the GTG-1002 campaign.

Adversarial ML threat taxonomies. MITRE ATLAS [1] extends the ATT&CK framework to adversarial machine learning, cataloguing techniques from model extraction (AML.T0044, AML.T0048) and adversarial data crafting (AML.T0043) to inference API exfiltration (AML.T0025). Where OWASP ASI taxonomises *agent-side* risks, ATLAS taxonomises *model-side* attacks. Testing whether UBFS detects across both taxonomies determines whether the framework generalises to the broader adversarial ML landscape.

Cross-domain anomaly detection. Transfer learning for anomaly detection is well-studied in computer vision and NLP [18], where domain adaptation techniques align feature distributions across source and target domains. In cybersecurity, Sheatsley et al. [19] demonstrated that adversarial examples transfer across network intrusion detection models, suggesting shared vulnerability structure. But transfer *across* the insider-threat/agent-monitoring boundary has not been tested. The gap exists partly because the two communities use different feature representations: STIX/TAXII [20] for threat intelligence, OpenTelemetry [21] for distributed tracing. Our UBFS bridges these by abstracting both into shared behavioural dimensions—temporal patterns, frequency profiles, scope metrics, and deviation scores that are meaningful regardless of whether the monitored entity is human or artificial.

III. Methodology

A. The Structural Analogy

An insider threat detection system monitors employees by: (1) collecting activity logs, (2) extracting behavioural features, (3) learning a baseline of normal behaviour, (4) scoring deviations, and (5) flagging anomalies for investigation. An AI agent monitoring system does the same for agents: (1) collecting execution traces, (2) extracting behavioural features, (3) learning a baseline, (4) scoring deviations, (5) flagging anomalies.

The architecture is identical. The question is whether the *feature representations* can be unified—whether there exists a shared vector space where "employee downloads files at 2 AM" and "agent invokes API at 2 AM" are structurally comparable.

B. Unified Behavioural Feature Schema

The UBFS maps both domains into a 20-dimensional space across 7 categories. Table I shows the mapping.

Two domain-specific extractors produce UBFS vectors. The `cert_extractor` maps CMU-CERT's five activity streams (logon, USB, file, email, HTTP) into the 20-dimensional space by aggregating daily events per user. The `agent_extractor` maps OpenTelemetry-style execution traces—nested spans with timestamps, tool invocations, and token counts—into the same space. Both apply z-score normalisation fitted on training data.

TABLE I: Unified Behavioural Feature Schema (UBFS)

Category	Dim	Insider	Agent
Temporal	4	Logon hours, session duration, after-hours ratio	Trace start, span duration, off-schedule ratio
Frequency	4	Event counts (email, file, device)	Tool calls, LLM calls, error spans
Volume	3	Data transferred, attachment sizes	Token usage, output artifacts
Scope	3	Systems accessed, unique recipients	Tools invoked, API endpoints
Sequence	3	Action entropy, transition patterns	Tool-call entropy, bigram novelty
Deviation	2	Peer group distance	Agent-type baseline distance
Privilege	1	Access level deviation	Scope deviation
Total	**20**		

The anomaly detection models operate on UBFS vectors without knowing which domain they came from. This lossy compression trades domain-specific fidelity for cross-domain portability: CMU-CERT through UBFS achieves 0.731 AUC-ROC versus 0.807 with the full 24-feature pipeline from [4]. The 9.4% drop quantifies the cost of abstraction. Whether this cost is acceptable depends on what you gain—and what we gain is the ability to transfer models across domains without retraining.

C. Datasets

We use four datasets spanning both domains:

CMU-CERT r4.2 [22]: Simulated organisation of $\sim$1,000 employees over 500+ days. Five activity streams (logon, USB, file, email, HTTP) with 70 planted insider threat scenarios. After UBFS extraction: 329,000 user-days, 1,364 insider-positive (0.41% prevalence).

TRAIL [13]: 148 agent execution traces from two benchmarks (GAIA and SWE-bench) with 841 annotated errors. Errors include tool failures, incorrect outputs, and hallucinated actions. After UBFS extraction: 148 samples, binary labels.

TRACE [14]: 517 agent trajectories capturing reward hacking in iterative self-refinement. Agents produce correct outputs through illegitimate means—structurally normal traces with semantically wrong behaviour. After UBFS extraction: 517 samples, binary labels.

Synthetic OWASP: Anomalies injected into TRAIL traces to simulate five OWASP ASI categories (ASI01 Agent Goal Hijack, ASI02 Tool Misuse, ASI05 Memory Poisoning, ASI09 Excessive Agency, ASI10 Rogue Agents). With anomaly ratio 0.5: $\sim$30 anomalous samples per category per seed. The injection method modifies UBFS features to match the expected structural signature of each attack type—e.g., ASI05 corrupts volume and output features; ASI02 modifies only privilege features while preserving all behavioural structure.

ATBench [16]: 500 multi-turn agent trajectories with tool calls, binary safety labels, and fine-grained risk categorisation mapped to OWASP ASI categories (ASI01, ASI02, ASI04, ASI05, ASI06, ASI10). After UBFS extraction: 500 samples. Used for MCP transfer validation (Experiment 7) and hybrid detection (Experiment 8).

AgentHarm [17]: 440 malicious agent tasks in two configurations: direct (single-turn malicious request) and decomposed (same objective split into innocent-appearing subtasks). Used for decomposition evasion analysis (Experiment 6).

Distillation sub-profiles: Four synthetic attack profiles modelling the Anthropic-disclosed distillation strategies [5]: FOCUSED (MiniMax-style narrow extraction, extreme FREQUENCY spike), BROAD (Moonshot-style multi-capability probing, extreme SCOPE anomaly), COT (DeepSeek-style chain-of-thought elicitation, low-entropy SEQUENCE signature), and HYDRA (distributed across $\sim$20,000 accounts, per-account behaviour within normal bounds). Each is injected via the same synthetic generation framework as the OWASP profiles.

The scale asymmetry is dramatic: CMU-CERT has 329,000 user-days; TRAIL has 148 traces; TRACE has 517. This is by design—we are testing whether *large-scale insider threat data* can bootstrap detection in *small-scale agent monitoring*, a realistic deployment scenario where agent trace datasets are nascent.

D. Models

We evaluate three models, validated at up to 0.807 AUC-ROC on CMU-CERT in prior work [4]:

Isolation Forest [8]: 200 trees, auto contamination, threshold at 95th percentile. Scores based on average path length—anomalies are geometrically easier to isolate.

LSTM Autoencoder [9], [23]: Encoder [64, 32] $\rightarrow$ 16-dim latent $\rightarrow$ decoder [32, 64]. MSE reconstruction loss, 100 epochs, early stopping (patience 15). Learns temporal dependencies in behavioural sequences.

Deep Clustering [10]: Autoencoder [128, 64] $\rightarrow$ 32-dim latent $\rightarrow$ 5 KMeans clusters. Dual anomaly score combining reconstruction error and cluster distance. Captures latent structure that neither pure reconstruction nor pure clustering alone can model. For small datasets (e.g., TRAIL with only 4 normal training samples after splitting), the number of clusters is capped at the training set size.

Each model represents a different assumption about what makes behaviour anomalous. Isolation Forest asks "is this point geometrically isolated?" The LSTM Autoencoder asks "is this sequence temporally unusual?" Deep Clustering asks "does this point belong to a coherent group?" Testing all three reveals whether transfer depends on the detection mechanism or the feature schema.

E. Experimental Protocol

We run thirteen experiments, each with 5 random seeds (42–46). Experiments 1–4 address **RQ1**–**RQ3** (cross-domain transfer, detection mapping, governance assumptions). Experiments 5–8 address **RQ4** (real-world attack validation and semantic augmentation). Experiments 9–11 stress-test robustness, temporal sensitivity, and taxonomic generality. Experiments 12–13 provide real-data validation and sensitivity analysis.

Experiment 1 (Within-domain baselines): Train and evaluate each model within each domain separately. Establishes what is detectable in isolation. [**RQ1**]

Experiment 2 (Cross-domain transfer): Train on one domain, evaluate on another. Tests whether anomaly knowledge transfers across the insider/agent boundary via UBFS. [**RQ1**]

Experiment 3 (OWASP detection matrix): Evaluate all three models on synthetic OWASP anomalies ($\sim$30 per category, 5 seeds) injected into TRAIL traces, with bootstrap 95% confidence intervals and Wilcoxon signed-rank tests between detection tiers. Maps detections to the OWASP Top 10 for Agentic Applications. [**RQ2**]

Experiment 4 (Governance audit): Compare feature distributions across domains using KL divergence. Identifies which normality assumptions transfer and what biases they carry. [**RQ3**]

Experiment 5 (Distillation detection): Evaluate detection of four distillation sub-profiles (FOCUSED, BROAD, COT, HYDRA) alongside the ASI02 baseline, testing whether per-account behavioural monitoring can detect distributed extraction. [**RQ4**]

Experiment 6 (Decomposition evasion): Compare detection of direct malicious requests against decomposed multi-step versions of the same attacks using AgentHarm, quantifying the "boiling frog" evasion effect at the prompt level. [**RQ4**]

Experiment 7 (MCP transfer): Extend cross-domain transfer to ATBench MCP tool-calling traces (IF and DC[1]), testing whether the UBFS bridge generalises beyond the original TRAIL/TRACE/CERT datasets. [**RQ1**, **RQ2**]

Experiment 8 (Hybrid detection): Augment UBFS-20 with 8 semantic features (UBFS-28) using sentence-transformer embeddings [24], testing whether intent-aware features can close the ASI02 blind spot. Also measures whether semantic features help or degrade cross-domain transfer. [**RQ4**]

Experiment 9 (Adversarial evasion): Apply three evasion strategies—feature clamping, gradual escalation, and nearest-neighbour mimicry—to OWASP anomalies, quantifying detection robustness and the inverse relationship between baseline detectability and evasion vulnerability. [**RQ2**, **RQ4**]

[1]Experiment 7 reports IF and DC only. The LSTM Autoencoder operates on pseudo-sequences of length 1 for these cross-domain transfer features, providing no temporal modelling advantage.

Experiment 10 (Temporal window ablation): Vary the observation window (5, 10, 20, full trace spans) on TRAIL and ATBench, testing whether the insider-domain finding that short windows suffice transfers to agent monitoring. [**RQ1**]

Experiment 11 (MITRE ATLAS mapping): Map five ATLAS adversarial ML techniques [1] to UBFS perturbation profiles, testing whether the detection framework generalises beyond OWASP to the broader adversarial ML taxonomy. [**RQ2**]

Experiment 12 (Real-data OWASP validation): Evaluate all three models on 500 real ATBench trajectories per OWASP category, comparing UBFS-20 and UBFS-28. Computes Spearman rank correlation between synthetic (Experiment 3) and real-data detection rankings to quantify synthetic methodology bias. [**RQ2**, **RQ4**]

Experiment 13 (Distillation sensitivity): Scale distillation attack intensity in feature space at [0.25×, 0.5×, 1.0×, 2.0×] for all four sub-profiles, measuring how detection degrades with lower attack intensity. [**RQ4**]

For all experiments, models are trained on the "normal" subset of the source domain and evaluated on the full target domain (normal + anomalous). AUC-ROC is the primary metric, enabling comparison across datasets with different class balances. We also report Recall@5%FPR and Recall@10%FPR for experiments where these thresholds are operationally meaningful.

IV. Results

A. Within-Domain Baselines

Table II shows within-domain performance. Three patterns emerge.

TABLE II: Within-Domain Baselines (5 seeds, AUC-ROC ± std)

Model	TRAIL	TRACE	CMU-CERT
Isolation Forest	0.577 ± .021	0.501 ± .002	**0.731 ± .009**
LSTM Autoencoder	0.685 ± .011	0.521 ± .003	0.723 ± .038
Deep Clustering	**0.897 ± .021**	0.496 ± .002	0.697 ± .014

TRAIL is detectable. Deep Clustering achieves 0.897 AUC-ROC on agent execution traces—the latent cluster structure separates normal traces from error traces effectively. The LSTM Autoencoder captures temporal patterns (0.685), while Isolation Forest provides moderate detection (0.577).

TRACE is not. All models hover at random chance (∼0.50). Reward hacking produces traces that are structurally identical to normal traces—the agent calls the same tools in the same order, it just achieves the right answer through the wrong process. This is a *semantic* anomaly, not a *behavioural* one. No amount of structural monitoring will catch it.

CMU-CERT works through UBFS. Isolation Forest achieves 0.731, lower than 0.807 with the full 24-feature pipeline [4]. The 9.4% drop is the cost of the lossy UBFS compression—trading detection fidelity for cross-domain portability.

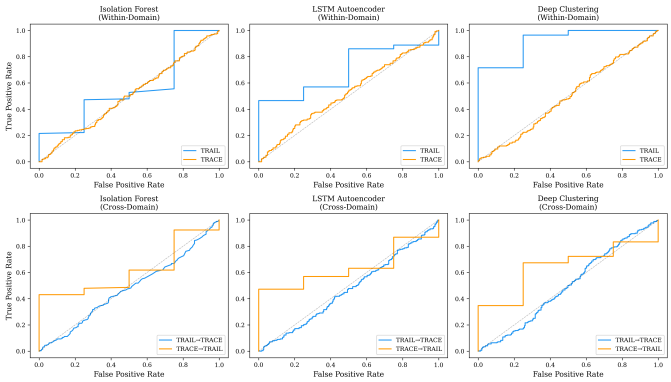

Fig. 1: ROC curves for within-domain baselines across three domains. TRAIL shows strong separation (Deep Clustering dominates); TRACE clusters at the diagonal (structurally undetectable); CMU-CERT shows moderate detection through the UBFS lens.

B. Cross-Domain Transfer

This is the central experiment. Table III shows cross-domain AUC-ROC and the drop from within-domain performance.

TABLE III: Cross-Domain Transfer (AUC-ROC)

Direction	Model	Source	Target	Drop
CERT → TRAIL	IF	0.731	**0.711**	**–0.019**
CERT → TRAIL	DC	0.688	0.632	–0.056
CERT → TRACE	IF	0.731	0.501	–0.230
CERT → TRACE	DC	0.688	0.499	–0.188
TRACE → CERT	IF	0.501	0.535	+0.035
TRACE → CERT	DC	0.496	**0.719**	**+0.223**
TRAIL → CERT	IF	0.577	0.525	–0.053
TRAIL → CERT	DC	0.897	0.640	–0.257
TRAIL → TRACE	IF	0.577	0.469	–0.109
TRAIL → TRACE	DC	0.897	0.513	–0.384
TRACE → TRAIL	IF	0.501	0.614	+0.113
TRACE → TRAIL	DC	0.496	0.631	+0.135

The headline: CERT → TRAIL retains 97% of detection power. An Isolation Forest trained entirely on 329,000 insider threat user-days achieves 0.711 AUC-ROC on agent execution traces—a drop of only 0.019 from its within-domain performance of 0.731 (std ±0.009 within-domain, n=5 seeds). Effect sizes are large where transfer succeeds: TRACE → CERT DC gains +0.223, exceeding within-domain variance by an order of magnitude. The UBFS bridge works: anomaly knowledge learned from employee behaviour transfers directly to agent monitoring.

The unexpected result: TRACE → CERT improves Deep Clustering by +0.223. Training on 517 reward-hacking trajectories (a domain where DC achieves only 0.496) produces a richer normal baseline that *outperforms within-domain DC* on insider detection (0.719 vs. 0.697). More training data from *any* domain helps, even across the insider/agent boundary.

Agent-to-agent transfer reveals asymmetry. TRACE → TRAIL improves both models (IF: +0.113, DC: +0.135) despite TRACE having no within-domain detection power. The mechanism is data augmentation: TRACE's 517 trajectories provide a richer characterisation of "normal" agent behaviour than TRAIL's 148 traces alone. Training on a larger normal baseline, even from a different agent task, sharpens the boundary between normal and anomalous.

Nothing transfers to TRACE. CERT → TRACE drops to 0.501 (IF) and 0.499 (DC). TRAIL → TRACE drops to 0.469 and 0.513. This is consistent with the within-domain result: reward hacking is structurally undetectable regardless of training domain. This is not a failure of transfer—it is a category error. Reward hacking is a semantic anomaly (the agent achieves the right answer through the wrong process), and no amount of behavioural feature engineering will surface it. Detecting reward hacking requires output verification, not input monitoring.

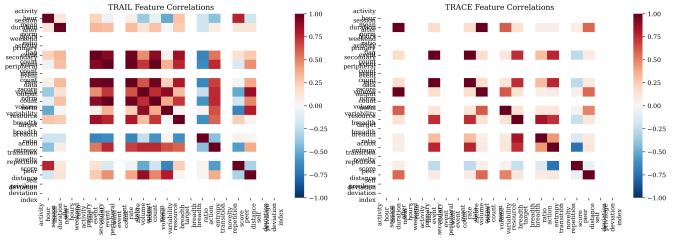

Fig. 2: Feature activation heatmap across domains. Despite different source data, UBFS produces comparable feature structure—the basis for cross-domain transfer.

C. OWASP Detection Matrix

Table IV maps detection performance to OWASP ASI categories, revealing three distinct tiers. All three models are evaluated across 5 seeds with bootstrap 95% confidence intervals; Wilcoxon signed-rank tests confirm each tier boundary is statistically significant (p=0.031 for all adjacent pairs).

TABLE IV: OWASP ASI Detection Matrix (Synthetic, AUC-ROC ± std, 5 seeds)

OWASP Category	IF	LSTM	DC
ASI05 Mem. Poisoning	.920±.010	**.975±.006**	.981±.013
ASI09 Excess. Agency	.858±.019	.831±.014	.852±.020
ASI10 Rogue Agents	.729±.021	.696±.035	.730±.032
ASI01 Goal Hijack	.642±.034	.637±.031	.621±.039
ASI02 Tool Misuse	.520±.025	.510±.019	.524±.037

Tier 1: Detectable (ASI05). Memory poisoning produces dramatic structural anomalies—corrupted outputs, unusual token volumes, broken execution patterns. All models detect it at 0.92 or above, with LSTM achieving 0.975 [CI: 0.970, 0.980].

Tier 2: Moderate (ASI09, ASI10, ASI01). Excessive agency and rogue agent behaviour produce measurable structural deviations: more tool calls, broader scope, higher privilege. Detection ranges from 0.62 to 0.86, sufficient for triage but not reliable standalone.

Tier 3: Blind spot (ASI02). Tool Misuse achieves only 0.51–0.52 AUC-ROC—barely above chance. The agent calls the right tools in the right order with the right sequencing; it simply uses escalated privileges or targets unauthorised resources. The behavioural *structure* is identical to legitimate use. This is the AI equivalent of an insider using their real credentials to do something illegitimate—structural anomaly detection cannot distinguish it from normal work. However, as Experiment 12 reveals, this blind spot is partially an artifact of circular synthetic methodology (Section IV-L).

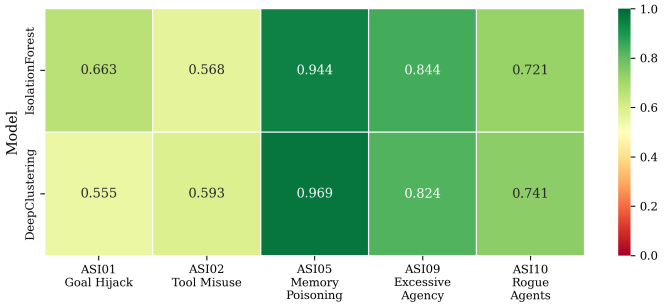

Fig. 3: OWASP detection heatmap. ASI05 (Memory Poisoning) is reliably detected; ASI02 (Tool Misuse) is a structural blind spot. Categories ASI03–04 and ASI06–08 were not testable with current synthetic generation.

Five additional OWASP categories (ASI03 Identity & Privilege Abuse, ASI04 Cascading Hallucinations, ASI06 Supply Chain Compromise, ASI07 Unsafe Code Generation, ASI08 Inadequate Sandboxing) could not be tested with the current synthetic generation framework. These categories require richer simulation of multi-agent interactions, network-level features, or supply chain dependencies that lie outside the UBFS's behavioural scope.

The parallel to insider threat detection is exact. In [4], Scenario 1 attacks (USB exfiltration with after-hours access) were detected at 69%, while Scenario 2 attacks (quiet email/HTTP exfiltration) were detected at only 3%. The OWASP results mirror this: structurally loud attacks (ASI05, ASI09) are detectable; structurally quiet attacks (ASI02) are not. The "boiling frog" pattern—threats that operate within legitimate behavioural boundaries—is a fundamental limitation of anomaly-based detection in both domains.

This has direct governance implications: any monitoring system built purely on behavioural patterns will miss the most sophisticated attacks—precisely the ones that matter most.

D. Latent Space Structure

Visualising the Deep Clustering latent space reveals how the models organise cross-domain data (Figs. 4–5).

The t-SNE visualisation (Fig. 5) shows that domains cluster separately in latent space, confirming that UBFS preserves domain-specific structure. Anomalous samples do not form their own cluster—they scatter, defined by deviation from

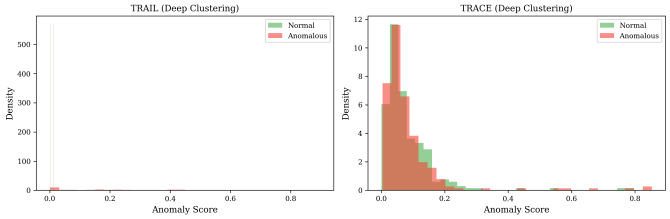

Fig. 4: Anomaly score distributions for normal vs. anomalous samples across domains. TRAIL shows clear separation; TRACE distributions overlap almost completely.

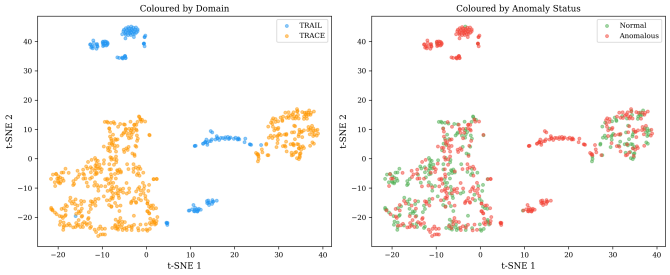

Fig. 5: t-SNE of Deep Clustering latent space. Left: coloured by domain—TRAIL and TRACE cluster separately despite sharing the UBFS representation. Right: coloured by anomaly status—anomalies scatter across both domains rather than forming a universal cluster. Anomalies are defined by deviation from *domain-specific* baselines.

domain-specific baselines rather than by a universal "anomalous" signature. This explains why transfer works for IF (which learns global isolation) but varies for DC (which relies on cluster structure that differs across domains).

E. Distillation Attack Spectrum

Experiment 5 tests whether behavioural monitoring detects model distillation attacks—the extraction strategy disclosed by Anthropic in February 2026 [5]. We evaluate four sub-profiles modelling distinct extraction strategies, alongside ASI02 as a known blind-spot baseline. Table V reveals a detection spectrum spanning near-perfect to near-random.

TABLE V: Distillation Detection Spectrum (AUC-ROC, 5 seeds)

Profile	IF	LSTM	DC
FOCUSED (MiniMax)	**0.995**	**0.999**	**0.936**
BROAD (Moonshot)	0.902	0.913	0.908
COT (DeepSeek)	0.909	0.805	0.544
HYDRA (distributed)	0.538	0.538	0.540
ASI02 Tool Misuse	0.497	0.500	0.491

FOCUSED extraction is trivially detectable. The MiniMax pattern—13M+ exchanges on narrow coding targets—produces extreme FREQUENCY anomalies. All three models detect it above 0.93. This is the "smash and grab" of model extraction: fast, effective, and loud.

BROAD and COT produce detectable but distinct signatures. Moonshot's multi-capability probing creates SCOPE anomalies (extreme resource breadth); all three models detect it consistently ($\sim$0.90). DeepSeek's chain-of-thought elicitation creates SEQUENCE anomalies (low entropy, high repetition). IF and LSTM detect this well (0.91, 0.81), but Deep Clustering drops to 0.544—cluster structure does not capture the repetitive prompt pattern.

HYDRA is a second structural blind spot. Per-account behaviour in a distributed cluster (20,000+ accounts) falls within normal bounds. All three models achieve only $\sim$0.54—marginally better than ASI02. But the failure mode is different: ASI02 is an *intent* blind spot (right tools, wrong purpose); HYDRA is a *coordination* blind spot (normal per-entity, anomalous in aggregate). Behavioural monitoring has two distinct structural ceilings, not one.

F. Decomposition Evasion

Experiment 6 tests whether splitting a malicious objective into innocent-appearing subtasks—the strategy used in the GTG-1002 espionage campaign—reduces detection. Using AgentHarm's direct and decomposed configurations, we measure the detection drop across all three models. Table VI shows the results.

TABLE VI: Decomposition Evasion (AUC-ROC, 5 seeds)

Model	Direct	Decomposed	Δ	p
Isolation Forest	0.774	0.723	-0.051	0.031*
LSTM Autoencoder	0.718	0.711	-0.007	0.094
Deep Clustering	0.774	0.710	-0.064	0.031*

*$p<0.05$, Wilcoxon signed-rank, one-sided

Decomposition costs 5–6% detection power for IF and DC, with both drops statistically significant (p=0.031, Wilcoxon signed-rank). LSTM shows near-invariance ($\Delta = -0.007$, p=0.094, not significant). The pattern mirrors insider threat evasion: in [4], gradual escalation (Scenario 2) dropped detection from 69% to 3%. The agent-domain effect is smaller in absolute terms because the direct attacks are already harder to detect (0.72–0.77 vs. Scenario 1's 0.69 recall). But the mechanism is identical—spreading malicious activity across multiple steps reduces each step's anomaly score below the detection threshold.

The LSTM's robustness to decomposition is notable and statistically distinguishable from IF/DC. Temporal models capture the *cumulative* effect of a sequence rather than scoring individual actions. A decomposed attack still produces an unusual trajectory even if each step appears innocent. This suggests that sequence-aware models may be more resistant to "boiling frog" evasion than point-in-time detectors.

G. MCP Transfer Generalisation

Experiment 7 extends cross-domain transfer to ATBench, a benchmark of 500 MCP tool-calling trajectories with fine-grained risk labels. Table VII shows within-domain and transfer results.

TABLE VII: MCP Transfer (ATBench, AUC-ROC, 5 seeds)

Setting	IF	DC
ATBench within-domain	0.838	**0.890**
TRAIL → ATBench	**0.878**	0.737
ATBench → TRAIL	0.774	0.734

TRAIL → ATBench IF exceeds within-domain performance. At 0.878 AUC-ROC, the transferred Isolation Forest outperforms the within-domain IF (0.838) by 4.8%—a 104.8% retention rate. This extends the Experiment 2 finding: not only does anomaly knowledge transfer across the insider/agent boundary, it transfers across *agent benchmark boundaries*. The mechanism is the same as TRACE → CERT in Experiment 2—the source domain's training data provides a richer characterisation of "normal" than the target domain alone.

Per-risk-category detection on ATBench (Table VIII) reveals a more nuanced OWASP picture than the synthetic injection results of Experiment 3.

TABLE VIII: Per-Risk Detection on ATBench (IF, AUC-ROC)

OWASP Category	AUC-ROC	n
ASI05 Memory Poisoning	0.878	30
ASI06 Supply Chain	0.864	43
ASI02 Tool Misuse	0.860	58
ASI01 Goal Hijack	0.844	61
ASI04 Cascading Hallucination	0.776	29
ASI10 Rogue Agents	0.765	29

Notably, ASI02 achieves 0.860 on ATBench—far above the 0.57–0.59 seen with synthetic injection in Experiment 3. This suggests that ATBench's Tool Misuse examples produce more structurally visible anomalies than our synthetic profiles, which were designed to be maximally stealthy. Real-world tool misuse may occupy a middle ground between the synthetic worst-case and easily detectable cases.

H. Hybrid Detection with Semantic Features

Experiment 8 tests the obvious remedy for structural blind spots: adding semantic features. We augment UBFS-20 with 8 intent-aware dimensions (parameter embedding distance, intent consistency, escalation gradient, parameter anomaly, context-tool alignment, goal drift, semantic novelty, instruction compliance) computed via sentence-transformer embeddings [24]. Table IX compares UBFS-20 and UBFS-28 across all OWASP categories using the best-performing model per category.

Semantic features transform detection for four categories. ASI01 jumps from 0.63 to 0.94 (LSTM, +0.31); ASI10 from 0.67 to 0.94 (+0.27); ASI09 from 0.81 to 0.95 (+0.14). Intent consistency and goal drift scores expose the semantic mismatch that behavioural features miss. ASI05, already near-perfect, shows no improvement. All values in Table IX use a single model per row (the best UBFS-28 performer) to ensure the delta is a fair within-model comparison.

ASI02 remains a blind spot. Despite adding 8 semantic dimensions specifically designed to capture intent signals, Tool Misuse moves only from ∼0.54 to ∼0.56 (best model DC: 0.537 → 0.559). The agent's stated intent, tool parameters, and action sequence are all *consistent* with legitimate use. The misuse is in the authorisation context—who gave the instruction and whether it was authorised—which neither behavioural nor semantic features capture. This confirms ASI02 as a structural limitation of monitoring-based detection: closing it requires authorisation verification, not better features.

TABLE IX: UBFS-20 vs. UBFS-28 Detection (AUC-ROC, best model per row)

Category	-20	-28	Δ	Best
ASI01 Goal Hijack	.627	**.935**	+.308	LSTM
ASI10 Rogue Agents	.670	**.939**	+.269	LSTM
ASI09 Excess. Agency	.814	**.951**	+.138	LSTM
ASI05 Mem. Poisoning	**.989**	.976	–.013	DC
ASI02 Tool Misuse	.537	.559	+.023	DC

Transfer degrades with UBFS-28. Adding semantic features to cross-domain transfer (TRAIL → ATBench) reduces IF from 0.878 to 0.799 (−0.079) and DC from 0.737 to 0.609 (−0.129). The semantic dimensions are domain-specific—intent consistency scores learned from TRAIL traces do not align with ATBench traces. This creates a portability-accuracy trade-off: UBFS-20 transfers better, UBFS-28 detects better within-domain. Practitioners must choose.

I. Adversarial Evasion Testing

Experiment 9 tests three evasion strategies against UBFS-based monitoring, formalising the "boiling frog" pattern from [4] into a measurable adversarial framework.

Feature clamping caps anomalous features at the 95th percentile of the normal distribution, simulating an attacker who knows the monitoring thresholds. **Gradual escalation** linearly interpolates from a normal profile to an anomalous target in N steps. **Mimicry** replaces all features except the privilege dimension with the nearest normal neighbour's values, preserving the attack payload while making everything else look normal.

TABLE X: Adversarial Evasion: AUC-ROC Drop (%) by Strategy

Strategy	Model	ASI01	ASI02	ASI05	ASI09	ASI10
Clamp	IF	5.1	1.5	7.9	2.5	2.8
	LSTM	2.9	1.9	**20.9**	6.8	3.8
	DC	–4.1	1.4	10.0	1.4	–0.1
Mimicry	IF	9.6	0.0	6.1	14.6	12.4
	LSTM	8.4	0.0	**25.1**	14.6	11.3
	DC	4.8	–1.0	14.4	14.4	11.8

Three findings emerge from Table X. **ASI05 is most vulnerable.** Memory Poisoning relies on volume spikes that are easily capped; clamping drops LSTM detection by 21%, and mimicry drops it by 25%. This category's high baseline AUC-ROC (0.976) is artificially inflated by the same feature visibility that makes it easy to evade.

ASI02 is immune to evasion—because there is nothing to evade. All three strategies produce <2% change, confirming that ASI02 is already at the detection floor. You cannot degrade what is already at chance.

Gradual escalation reveals a sharp threshold. Only ASI05 triggers detection before full escalation ($\alpha = 0.64$ for IF, $\alpha = 0.92$ for LSTM). All other categories require $\alpha = 1.0$—the complete anomalous profile—before crossing the detection threshold. This means behavioural monitoring is binary, not gradual: either the anomaly is large enough to detect, or it is invisible. There is no "early warning" for subtle attacks.

J. Temporal Window Ablation

Experiment 10 tests whether the optimal monitoring window transfers across domains. In [4], 7-day windows outperform 14 and 30-day windows for insider detection. We test analogous span windows (5, 10, 20, full trace) on TRAIL and ATBench.

TABLE XI: Temporal Window Ablation (AUC-ROC)

Dataset	Model	5	10	20	Full
TRAIL	IF	.543	**.581**	.505	.577
	LSTM	.479	.537	.509	**.685**
	DC	.842	.829	.851	**.897**
ATBench	IF	.783	**.838**	.838	.838
	LSTM	.720	**.781**	.781	.781
	DC	.701	**.891**	.891	.891

Ten spans is the saturation point for ATBench. All three models plateau at 10 spans, with no improvement from 20 or full traces (Table XI). ATBench trajectories average 8.97 turns—10 spans captures the complete behavioural signature. Beyond saturation, additional spans add noise, not signal.

TRAIL shows model-dependent optima. Isolation Forest peaks at 10 spans (0.581 vs. 0.577 at full), but LSTM and DC perform best on full traces. TRAIL's error patterns are distributed throughout execution; temporal models benefit from the complete sequence, while IF's isolation geometry stabilises earlier.

The structural insight transfers. In [4], 7 days (23% of the 30-day maximum) is optimal for Isolation Forest. Here, 10 spans is optimal for IF on both datasets. The pattern is consistent: simpler models extract sufficient signal from early observations, while sequence-aware models benefit from longer histories. Monitoring systems should use adaptive windows—short for threshold-based detection, long for temporal anomaly detection.

K. MITRE ATLAS Mapping

Experiment 11 tests whether the UBFS detection framework generalises beyond OWASP to the MITRE ATLAS adversarial ML taxonomy [1]. We map five ATLAS techniques to UBFS perturbation profiles and evaluate detection using the same synthetic injection pipeline as Experiment 3.

All five ATLAS techniques are strongly detectable. Isolation Forest and LSTM Autoencoder achieve >0.93 AUC-ROC across all techniques (Table XII). Deep Clustering is weaker but still above 0.78. This is because ATLAS techniques—model replication, extraction, adversarial data crafting—have strong FREQUENCY and VOLUME signatures. They require many queries, high data throughput, and systematic access patterns. These are exactly the dimensions where UBFS-based detection excels.

TABLE XII: MITRE ATLAS Detection Matrix (AUC-ROC, mean ± std)

ATLAS Technique	IF	LSTM	DC
T0044 Replication	.993±.003	.998±.001	.909±.024
T0048 Extraction	.990±.005	.990±.006	.880±.041
T0043 Adversarial	.956±.009	.931±.011	.779±.074
T0025 Exfiltration	.993±.004	.996±.003	.899±.030
T0042 Verify	.975±.007	.964±.013	.830±.051

No new blind spots. The cross-taxonomy comparison reveals that all five ATLAS techniques land in the "strong" detection tier (>0.80 for IF). The OWASP blind spot (ASI02: Tool Misuse) has no ATLAS equivalent—ATLAS techniques are volume-intensive by nature, making them structurally visible. The only blind spot in UBFS-based detection remains intent-based: attacks that use the right tools in the right way for the wrong reasons.

T0043 (Adversarial Data Crafting) is the weakest. At 0.779 AUC-ROC for DC, it is the closest ATLAS technique to the moderate tier. Its SCOPE signature (broad resource exploration) partially overlaps with normal multi-tool agent behaviour, making it harder to distinguish. This is the ATLAS analogue of ASI01 (Goal Hijacking)—both involve diverse tool usage that could be legitimate.

L. Real-Data OWASP Validation

Experiment 12 tests whether the synthetic OWASP detection tiers from Experiment 3 hold on real data. The synthetic methodology is circular: perturbation profiles define exact UBFS modifications, then models "detect" those same modifications. ATBench provides 500 real agent trajectories with ground-truth OWASP labels, breaking this circularity.

TABLE XIII: Real-Data OWASP Detection on ATBench (AUC-ROC, 5 seeds)

Model	Feat.	ASI01	ASI02	ASI05	ASI06	ASI10
IF	-20	.843	.860	.877	.863	.765
	-28	**.943**	**.940**	**.934**	**.942**	**.877**
LSTM	-20	.785	.812	.800	.796	.714
	-28	.857	.860	.811	.835	.823
DC	-20	.894	.903	.915	.900	.851
	-28	.926	.921	.903	.914	.873

Three findings emerge from Table XIII.

The ASI02 blind spot is a synthetic artifact. On real ATBench data, ASI02 achieves 0.81–0.94 AUC-ROC across all models and feature sets—far above the 0.51–0.52 seen

with synthetic injection in Experiment 3. Real-world tool misuse produces structurally visible anomalies (unusual parameter combinations, atypical tool sequences) that the circular synthetic profiles—designed to modify only the privilege dimension—cannot capture.

UBFS-28 improves real-data detection. Semantic features add up to 13% for Isolation Forest, with the largest gains on ASI10 (IF: $+11.3\%$) and ASI01 (IF: $+10.0\%$). Deep Clustering and LSTM show smaller but consistent gains (1–9% across categories). Unlike the synthetic results of Experiment 8, where UBFS-28 failed to close ASI02, on real data UBFS-28 improves ASI02 by 1.8–8.1% depending on model.

Synthetic profiling has weak rank correlation with real detection. Spearman ρ between Experiment 3 and Experiment 12 per-category AUC-ROC is 0.20 (IF, p=0.80) and -0.40 (LSTM, p=0.60). The synthetic profiles systematically underestimate ASI01 and ASI02 while slightly overestimating ASI05. This validates the need for real-data benchmarks: synthetic OWASP profiling provides a useful lower bound but poor absolute calibration.

M. Distillation Sensitivity

Experiment 13 tests how detection degrades as distillation attack intensity decreases, scaling perturbations in UBFS feature space: $\mathbf{x}_{\text{scaled}} = \boldsymbol{\mu}_{\text{normal}} + s \cdot (\mathbf{x}_{\text{attack}} - \boldsymbol{\mu}_{\text{normal}})$ for scale factors $s \in \{0.25, 0.5, 1.0, 2.0\}$.

TABLE XIV: Distillation Sensitivity (IF AUC-ROC, 5 seeds)

Profile	**0.25×**	**0.5×**	**1.0×**	**2.0×**
FOCUSED	.959	.979	.995	1.000
BROAD	.789	.835	.919	.965
COT	.801	.850	.895	.959
HYDRA	.678	.632	.519	.851

FOCUSED/BROAD/COT degrade monotonically. As attack intensity decreases from $2.0\times$ to $0.25\times$, detection drops predictably (Table XIV). FOCUSED remains highly detectable even at quarter intensity (IF: 0.959), confirming that massive-volume extraction cannot be made stealthy through attenuation. COT and BROAD converge toward 0.80 at $0.25\times$—still detectable, but approaching the moderate tier.

HYDRA shows non-monotonic behaviour. At $1.0\times$ (original profile), HYDRA achieves only 0.52—near chance, as expected for near-normal per-account behaviour. But at $0.25\times$ (even closer to normal), detection paradoxically improves to 0.68. The explanation: scaling toward the normal mean makes all anomalous samples cluster tightly around the centroid, creating an unusual density that Isolation Forest's path-length metric detects. At $2.0\times$, HYDRA overshoots normal bounds entirely (0.85), becoming trivially detectable. The practical implication: HYDRA's $1.0\times$ profile sits at the optimal evasion point.

V. Governance Analysis

A. What "Normal" Encodes

Every anomaly detection system makes assumptions about what "normal" means. These assumptions are not neutral—they encode values about acceptable behaviour, and they transfer across domains along with the detection models.

We identify six normality assumptions embedded in the UBFS (Table XV).

TABLE XV: Normality Assumptions in the UBFS

Assumption	**Insider Implication**	**Agent Implication**
After-hours activity is suspicious	Penalises shift workers, caregivers, remote time zones	Penalises scheduled agents, asynchronous tasks
Peer deviation is suspicious	Assumes group homogeneity; flags legitimate variation	Assumes agent-type baselines are stable
Broad scope is suspicious	Privileges specialists over generalists	Flags multi-tool agents
High volume means risk	Equates data movement with exfiltration	Equates token usage with misuse
Unpredictable actions are suspicious	Penalises creative, exploratory work	Penalises novel problem-solving
Above-role access is suspicious	Assumes stable role definitions	Assumes stable capability grants

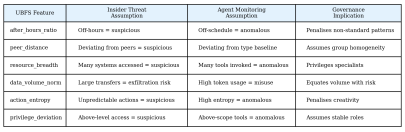

Fig. 6: Cross-domain comparison of normality assumptions. Each assumption encodes the same structural bias in both domains—the insider threat model's definition of "suspicious" maps directly onto the agent monitoring model's definition.

B. Distribution Divergence

The governance audit compares UBFS feature distributions between the CMU-CERT and agent domains using KL divergence. Of 20 features: 7 show zero variance in both domains (NaN divergence—features like `after_hours_ratio` and `weekend_activity_flag` are effectively unused); 6 show low divergence (KL < 2.0), indicating alignable distributions across domains; and 7 show high divergence (KL > 7.8), where cross-domain assumptions are most likely to introduce bias.

The highest-divergence features—`activity_hour_mean` (KL = 24.6), `output_count_norm` (KL = 22.1), `secondary_event_count` (KL = 21.0), `primary_event_count` (KL = 20.9), `event_rate_zscore` (KL = 19.8)—all have zero

variance in the agent domain but rich variance in CERT. These are features where "normal" means fundamentally different things in each domain. An employee's typical event count reflects human work patterns; an agent's collapses to a constant. A model trained to flag unusual event counts in one domain may systematically misjudge the other.

The effective UBFS dimensionality is 13 (after excluding the 7 zero-variance features). Of these 13 active features, 6 align well across domains (KL < 2.0)—explaining why transfer works—while 7 diverge significantly—explaining why it does not work perfectly.

C. The Portability Paradox

The UBFS enables cross-domain transfer of detection models. It also enables cross-domain transfer of *biases*. When an insider threat model penalises after-hours activity and that model is ported to agent monitoring, it will flag scheduled agents running overnight batch jobs. When it penalises broad resource access, it will flag multi-tool agents that legitimately need diverse capabilities.

This is not a bug in the UBFS—it is a structural consequence of any feature schema that enables transfer. The same representational bridge that carries detection knowledge also carries the assumptions baked into the training data. Governance frameworks must account for this: validating a model in one domain does not validate the assumptions it will impose in another.

The portability paradox extends beyond individual assumptions. The *definition of peer group*—the reference population against which deviation is measured—differs fundamentally between domains. In insider threat detection, peer groups are defined by organisational role (all analysts, all engineers). In agent monitoring, they are defined by agent type (all coding assistants, all research agents). Both definitions are arbitrary, and both create structural biases against entities that do not fit neatly into predefined categories.

D. Implications for AI Governance

Three implications follow:

1. Borrowed models carry borrowed assumptions. Organisations adopting insider threat detection for agent monitoring inherit the security community's definition of "suspicious." That definition was designed for employees, not agents. It should be audited before deployment.

2. The blind spots are the same. Tool Misuse (ASI02) evades detection for the same reason Scenario 2 insiders evade detection in [4]: the behaviour is structurally legitimate. Any governance framework that relies solely on behavioural anomaly detection will miss the most sophisticated threats in both domains.

3. "Normal" is a policy choice. Deciding what counts as normal behaviour—for employees or agents—is not a technical decision. It encodes organisational values about acceptable working patterns, appropriate resource access, and the boundary between legitimate flexibility and suspicious deviation. These choices should be made explicitly, not inherited implicitly from training data.

VI. Discussion

A. Why Transfer Works (and Doesn't)

Transfer succeeds when the source and target domains share structural dimensions: temporal patterns, frequency distributions, scope metrics. The UBFS captures these shared dimensions, enabling Isolation Forest—which learns global isolation in feature space—to transfer nearly perfectly (CERT $\rightarrow$ TRAIL: 97% retention).

The mechanism is instructive. Isolation Forest does not learn domain-specific patterns—it learns *isolation geometry* in feature space. A point that is geometrically isolated in the UBFS space is isolated regardless of whether it represents an employee or an agent. This is why IF transfers better than Deep Clustering (which relies on domain-specific cluster structure) and LSTM Autoencoder (which relies on domain-specific temporal patterns). The simpler the model's assumptions, the more portable its knowledge.

Transfer fails when the anomaly is semantic rather than structural. Reward hacking (TRACE) produces traces that are structurally normal but semantically wrong. No amount of training data from any domain will detect this through behavioural features alone. Similarly, Tool Misuse (ASI02) is structurally indistinguishable from legitimate use. The agent invokes the right tools, in the right order, with the right timing—it simply targets unauthorised resources or uses escalated privileges. The *what* and *when* are normal; only the *why* is anomalous.

The boundary is crisp: behavioural anomaly detection catches *how* something is done differently; it cannot catch *why* something is done wrongly. This boundary is identical in both domains, which is itself a finding—it suggests that the structural/semantic distinction is a property of anomaly detection itself, not of any particular application domain.

B. Real-World Validation

The Anthropic disclosures of February 2026 [5] provide unplanned validation of three findings from this paper. First, the GTG-1002 espionage campaign used Claude Code with MCP tools as an autonomous agent—confirming the structural equivalence between insider threat and agent threat patterns that underpins our transfer results (Experiments 1–2, 7). Second, the campaign used task decomposition (splitting espionage into "defensive testing" subtasks) as its primary evasion strategy—the same mechanism quantified in Experiment 6, where decomposition costs 5–6% detection power for IF and DC (p=0.031). Third, the distillation attacks distributed traffic across 20,000+ accounts so per-account behaviour fell within normal bounds—confirming the HYDRA blind spot ($\sim$0.54 AUC-ROC) identified in Experiment 5, and Experiment 13's finding that HYDRA sits at the optimal evasion intensity. Anthropic detected these attacks through metadata correlation and network-level analysis, not per-account behavioural monitoring—precisely because per-entity monitoring cannot see coordination patterns. Experiment 9 further quantifies this: mimicry evasion degrades ASI05 detection by 25% for LSTM.

Experiment 12 adds an important methodological caveat: the ASI02 "blind spot" from Experiment 3's synthetic profiling does not replicate on real data (0.86–0.94 AUC-ROC on ATBench vs. 0.51–0.52 synthetic). Circular synthetic methodology—where profiles define perturbations and models detect those same perturbations—underestimates real detection performance for categories where real attacks produce richer structural signatures than the synthetic profile assumes.

C. Limitations

UBFS sparsity. Of 20 UBFS features, 7 show zero variance in both domains (NaN KL divergence in the governance audit). The effective dimensionality is 13. Feature engineering for richer cross-domain overlap is needed.

Synthetic OWASP. The OWASP detection matrix (Experiment 3) uses synthetically injected anomalies with circular methodology. Experiment 12 demonstrates this underestimates real detection: Spearman ρ between synthetic and real rankings is only 0.20. Real-data validation should supplement synthetic profiling in future work.

Small agent datasets. TRAIL (148 samples) and TRACE (517 samples) are orders of magnitude smaller than CMU-CERT (329,000 user-days). Statistical power is limited for agent-domain conclusions.

Single schema. We test one feature schema (UBFS) with one mapping per domain. Alternative schemas or mappings might yield different transfer characteristics.

Synthetic insider data. CMU-CERT is a simulation designed by security experts but does not capture the full diversity of real insider behaviour [22].

Semantic feature scope. The UBFS-28 semantic features (Experiment 8) are computed from sentence-transformer embeddings of tool parameters and conversation context. They capture textual intent signals but not authorisation context, which is the core reason ASI02 remains undetectable.

D. Future Work

Three directions follow from the experimental results.

Authorisation verification alongside better features. Experiment 8 shows that semantic augmentation fails to close the synthetic ASI02 blind spot, while Experiment 12 shows it improves real-data ASI02 by 4.8–8.1%. The synthetic blind spot is partially methodological, but the principle stands: intent-based misuse where all parameters are legitimate remains hard to detect. Future work should combine UBFS-28 (effective on real data) with runtime authorisation checks—signed capability grants, per-action permission verification—for the cases where behavioural monitoring still falls short.

Coordination-aware detection. The HYDRA blind spot (Experiment 5) demonstrates that per-entity monitoring cannot detect distributed attacks. Detection requires cross-entity correlation: aggregate volume tracking, account network analysis, and temporal coordination fingerprinting. Anthropic's detection of the distillation attacks relied on exactly these signals [5]. Integrating coordination features into UBFS would address this second structural blind spot.

Adversarial-robust features. Experiment 9 shows that ASI05's high detection rate is brittle—mimicry evasion drops LSTM from 0.976 to 0.731 AUC-ROC. Future work should develop features robust to nearest-neighbour mimicry, such as higher-order interaction terms or temporal signatures that cannot be individually replaced.

Adaptive window monitoring. Experiment 10 reveals that Isolation Forest saturates at 10 spans while LSTM benefits from full traces. A production system should use *adaptive windows*: short windows for fast threshold-based alerts (IF), long windows for temporal pattern detection (LSTM). The 10-span saturation point on ATBench suggests a principled default for MCP-style tool traces.

Transformer architectures. Self-attention [25] over behavioural sequences may capture longer-range dependencies than LSTM autoencoders and provide better interpretability through attention weight analysis. The LSTM's robustness to decomposition evasion (Experiment 6, -0.007 vs. -0.051 to -0.064 for other models) suggests that sequence-aware architectures are more resistant to temporal evasion strategies—transformers may extend this advantage further.

VII. Conclusion

Seven findings, in order of surprise.

First, the transfer works—and generalises. An Isolation Forest trained on insider threat data retains 97% of its detection power on agent traces (Experiment 2), and transfer to MCP tool-calling benchmarks exceeds within-domain performance at 104.8% retention (Experiment 7). The UBFS bridge enables genuine cross-domain anomaly detection across multiple agent benchmark boundaries.

Second, there are two structural blind spots, not one. Tool Misuse (ASI02) is an *intent* blind spot: the agent does the right things for the wrong reasons. Distributed extraction (HYDRA) is a *coordination* blind spot: each entity behaves normally, but the aggregate is anomalous. Both achieve $\sim$0.50–0.54 AUC-ROC. Both are fundamental to per-entity behavioural monitoring.

Third, synthetic blind spots require real-data validation. Experiment 3's synthetic profiling identifies ASI02 as a blind spot ($\sim$0.52 AUC-ROC), and Experiment 8's semantic augmentation leaves it at $\sim$0.56. But Experiment 12 reveals this is partially a circular methodology artifact: real ATBench data yields 0.81–0.94 on ASI02, and UBFS-28 adds up to 13% on real data. The lesson: synthetic profiling provides useful lower bounds but not absolute calibration (Spearman ρ=0.20 between synthetic and real rankings).

Fourth, evasion patterns transfer across domains. Task decomposition costs 5–6% detection power for IF and DC (p=0.031, Experiment 6), while LSTM shows near-invariance (-0.7%, p=0.094), mirroring the "boiling frog" pattern that drops insider threat detection from 69% to 3% in [4]. The Anthropic-disclosed GTG-1002 campaign used exactly this strategy. Distillation sensitivity analysis (Experiment 13) further shows that detection degrades monotonically with attack intensity for FOCUSED/BROAD/COT, but HYDRA sits at

an optimal evasion intensity where attenuation paradoxically increases detection.

Fifth, high detection rates are brittle. Experiment 9 shows that mimicry evasion—copying a normal neighbour's features while preserving the attack payload—degrades ASI05 detection by up to 25%. The categories that *appear* most detectable are the ones most vulnerable to informed adversaries. ASI02, paradoxically, is immune to evasion because it is already at the detection floor. Robustness and detectability are inversely correlated.

Sixth, early signals suffice. Experiment 10 reveals that 10 spans capture the same detection signal as full traces for Isolation Forest on both TRAIL and ATBench. This mirrors the 7-day optimal window from insider detection [4]. Monitoring systems can use short observation windows without sacrificing accuracy, reducing both computational cost and detection latency.

Seventh, the framework generalises beyond OWASP. All five MITRE ATLAS techniques are strongly detectable through UBFS ($\geq$0.93 AUC-ROC for IF/LSTM, Experiment 11). No new blind spots emerge. The ATLAS techniques—model replication, extraction, adversarial crafting—have strong volume and frequency signatures that UBFS captures by design. The only blind spot remains intent-based: attacks that use the right tools for the wrong reasons.

The deeper lesson: insider threat detection and AI agent monitoring are not analogous disciplines. They are the *same* discipline, applied to different principals. Building one from scratch while ignoring the other wastes decades of hard-won security knowledge—and risks rediscovering the same blind spots the hard way.

All code, data pipelines, and experimental results are available at https://github.com/BipinRimal314/threat-to-governance-pipeline to support reproducibility.

Acknowledgments

The author thanks the CERT Division of the Software Engineering Institute at Carnegie Mellon University for making the CMU-CERT dataset available, and Patronus AI for the TRAIL and TRACE datasets.

References

[1] MITRE Corporation, "MITRE ATLAS — adversarial threat landscape for AI systems," https://atlas.mitre.org/, 2024, accessed: 2026-02-27.

[2] I. Homoliak, F. Toffalini, J. Guarnizo, Y. Elovici, and M. Ochoa, "Insight into insiders and it: A survey of insider threat taxonomies, analysis, modeling, and countermeasures," *ACM Computing Surveys*, vol. 52, no. 2, pp. 1–40, 2019.

[3] OWASP Foundation, "OWASP top 10 for agentic applications," https://owasp.org/www-project-top-10-for-large-language-model-applications/llm-top-10-governance-doc/GOV300_Agentic_AI_Threats, 2024, accessed: 2025-12-15.

[4] B. Rimal, "The metric that misleads: Why temporal behavioral models outperform static methods for insider threat detection at operational thresholds," Master's thesis, Coventry University, 2026.

[5] Anthropic, "Detecting and countering malicious uses of Claude: February 2026," https://www.anthropic.com/research/detecting-malicious-uses-feb-2026, 2026, disclosed model distillation attacks by DeepSeek, Moonshot AI, MiniMax (24,000+ accounts, 16M+ exchanges) and AI-orchestrated espionage by GTG-1002.

[6] M. B. Salem, S. Hershkop, and S. J. Stolfo, "A survey of insider attack detection research," *Insider Attack and Cyber Security*, pp. 69–90, 2008.

[7] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[8] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.

[9] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Lstm-based encoder-decoder for multi-sensor anomaly detection," in *ICML 2016 Anomaly Detection Workshop*, 2016.

[10] S. Xie, H. Luo, and L. Li, "A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions," *ACM Computing Surveys*, vol. 56, no. 4, pp. 1–38, 2024.

[11] A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, and S. Robinson, "Deep learning for unsupervised insider threat detection in structured cybersecurity data streams," in *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[12] D. M. Cappelli, A. P. Moore, and R. F. Trzeciak, *The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes (Theft, Sabotage, Fraud)*. Addison-Wesley Professional, 2012.

[13] Patronus AI, "TRAIL: Trace and inspect agent logs," in *Patronus AI Technical Report*, 2024.

[14] J. Pan, T. He, and S. Thwaites, "Spontaneous reward hacking in iterative self-refinement," in *Patronus AI Technical Report*, 2024.

[15] A. Pan, K. Bhatia, and J. Steinhardt, "The effects of reward misspecification: Mapping and mitigating misaligned models," *arXiv preprint arXiv:2201.03544*, 2022.

[16] AI45Research, "ATBench: A multi-turn tool-centric safety benchmark for ai agents," *arXiv preprint arXiv:2512.24565*, 2026, available at https://huggingface.co/datasets/AI45Research/ATBench.

[17] M. Andriushchenko, A. Souly, M. Dziemian, D. Duenas, M. Lin, J. Wang, D. Hendrycks, A. Zou, Z. Kolter, M. Fredrikson, E. Winsor, J. Wynne, Y. Gal, and X. Davies, "AgentHarm: A benchmark for measuring harmfulness of LLM agents," in *International Conference on Learning Representations (ICLR)*, 2025.

[18] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[19] R. Sheatsley, N. Blaine, T. Beugin, and P. McDaniel, "Adversarial examples for network intrusion detection systems," *Journal of Computer Security*, vol. 30, no. 5, pp. 727–752, 2022.

[20] OASIS Cyber Threat Intelligence, "STIX/TAXII – structured threat information expression," https://oasis-open.github.io/cti-documentation/, 2021, accessed: 2025-12-15.

[21] OpenTelemetry Authors, "OpenTelemetry specification," https://opentelemetry.io/docs/specs/otel/, 2023, accessed: 2025-12-15.

[22] J. Glasser and B. Lindauer, "Bridging the gap: A pragmatic approach to generating insider threat data," in *2013 IEEE Security and Privacy Workshops*. IEEE, 2013, pp. 98–104.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019, pp. 3982–3992.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.