International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 06 June 2026
Page No: 4222- 4231
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0548

Multi-Agent Semantic Drift Detection and Self-Correction in Cybersecurity LLM Pipelines

Hathijathul Ameena Beevi[1], Fathima Saliha[2], Fahmeedha Thasneem[3]
[1]Assistant Professor, C.K College of Engineering and Technology, Cuddalore, Tamil Nadu
[2]Python Developer, Bangalore, Karnataka
[3]Student, KGISL Institute of Technology, Coimbatore, Tamil Nadu
Emails: *rasheedameena@gmail.com[1], fathimasaliha2004@gmail.com[2], fahmeedha0304@gmail.com[3]*

Abstract
Large Language Model (LLM) pipelines in cybersecurity contexts face a critical but under-studied failure mode: semantic drift, in which outputs progressively deviate from original intent, technical constraints, or safety boundaries as they pass through multiple agent handoffs. This paper presents CyberDrift, a multi-agent closed-loop architecture for semantic drift detection and self-correction in cybersecurity intelligence pipelines. The system makes two primary contributions: (i) discordance-aware reasoning, in which conflict between seven heterogeneous evaluator agents is computed as a structured signal beyond weighted similarity, enabling detection of subtle drift patterns such as entity suppression masked by surface-level fluency; and (ii) an Intelligent Gatekeeper Agent that makes Pass/Repair/Block routing decisions based on both drift magnitude and inter-agent discordance. The discordance signal identifies four interpretable conflict patterns (entity-compression, compression-uncertainty, semantic-compression, and entity-structural), each corresponding to a distinct cybersecurity failure mode. We further characterise the Repair Module through explicit algorithmic specification, demonstrate that repair effectiveness (87.0% mean drift reduction, 99.4% constraint preservation) is robust to classifier error, and establish an ablation baseline confirming that discordance-aware routing reduces unnecessary repair invocations relative to threshold-only gating. CyberDrift demonstrates that evaluator disagreement is itself a structured diagnostic signal, not a noise artefact to be averaged away.

Keywords: *multi-agent systems, semantic drift, cybersecurity LLM pipelines, discordance-aware evaluation, self-correcting AI, gatekeeper agent, hallucination detection, intent drift, constraint preservation*

1. Introduction
The deployment of Large Language Models (LLMs) in operational cybersecurity contexts (including threat intelligence synthesis, incident triage, red-team documentation, and autonomous vulnerability analysis) has accelerated substantially in recent years. Security operations centres (SOCs) increasingly rely on LLM-assisted pipelines to process high-velocity alert streams that exceed human analytical bandwidth. However, this operational reliance exposes organisations to a class of failure mode that neither traditional software testing nor standard LLM evaluation benchmarks adequately characterise: semantic drift across multi-agent reasoning chains. Semantic drift, as defined in this work, is the progressive deviation of a pipeline output from its originating intent, factual constraints, or ethical boundaries as the output traverses successive agent handoffs. Unlike discrete hallucination, which involves a model fabricating a specific fact, semantic drift operates at the level of purpose and framing. A pipeline that begins by analysing a SQL injection incident for defensive awareness may, after two agent iterations, produce a comprehensive offensive red-team artefact that no single agent step would have generated or approved independently. This phenomenon poses acute risks in

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 06 June 2026
Page No: 4222- 4231
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0548

cybersecurity for three reasons. First, the domain handles dual-use information where a minor shift in framing transforms a defensive analysis into an attack enablement resource. Second, SOC outputs are frequently acted upon under time pressure, reducing human review bandwidth. Third, existing LLM reliability mechanisms such as constitutional AI, RLHF, and single-pass output filtering operate at the token or sentence level and are not designed to reason about the semantic trajectory of a multi-step pipeline. This paper presents CyberDrift, a multi-agent reliability framework designed to address semantic drift in LLM-based cybersecurity pipelines. CyberDrift departs from prior work in two principal respects. First, it introduces a discordance-aware evaluation paradigm in which disagreement between seven heterogeneous evaluator agents is retained as a structured diagnostic signal rather than averaged away as measurement noise. Second, it implements an Intelligent Gatekeeper Agent that combines drift magnitude with inter-agent discordance to make three-way routing decisions: Pass, Repair, or Block. Third, new in this revision, we fully specify the Repair Module algorithm and validate its behaviour against an ablation baseline [1-6].

2. Background and Related Work

2.1. Hallucination and Reliability in LLMs

Hallucination in LLMs, defined as the generation of factually incorrect or contextually inappropriate content, has been extensively documented [1, 2]. Early taxonomy work distinguished intrinsic hallucination (contradiction of provided context) from extrinsic hallucination (introduction of unsupported claims) [3]. Mitigation strategies include retrieval-augmented generation (RAG) [4], chain-of-thought prompting [5], and post-hoc factual verification [6]. These approaches address hallucination at individual model invocations but do not characterise the drift trajectory of multi-step pipelines.

2.2. Multi-Agent LLM Systems

The shift toward multi-agent LLM systems has been motivated by evidence that debate, self-reflection, and cross-agent verification improve factual accuracy [7, 8]. Frameworks such as AutoGen [9] and CrewAI provide implementation infrastructure for multi-agent pipelines. A critical gap in existing research is the treatment of inter-agent disagreement: most frameworks aggregate agent votes via weighted averaging, discarding the diagnostic information contained in disagreement patterns. CyberDrift explicitly addresses this gap by retaining conflict structure as a primary output.

2.3. LLM Safety in Cybersecurity

Dual-use framing, in which identical technical content is safe in one context and dangerous in another, makes context-sensitivity a first-order safety requirement in cybersecurity LLM deployment [11]. This challenge is compounded in multi-step pipelines, where no single agent observes the full transformation from original intent to final output. Benchmark datasets such as CyberSecEval [12] and HarmBench [13] evaluate LLM safety on adversarial prompts, but primarily assess single-query responses rather than multi-step pipeline trajectories. The closest prior work is the study of prompt injection in agentic pipelines [14], which examines how malicious inputs subvert agent behaviour. CyberDrift extends this concern to non-adversarial, emergent drift that arises without any intentional attack.

2.4. Semantic Similarity and Drift Measurement

Semantic similarity measures, including cosine similarity over sentence embeddings [15], BERTScore [16], and ROUGE [17], provide the technical substrate for drift detection. However, surface-level semantic similarity is necessary but not sufficient for intent preservation: two outputs may share high embedding similarity while differing substantially in framing, entity specificity, or operational implication. This motivates CyberDrift's multi-dimensional evaluation architecture and its treatment of inter-dimensional disagreement as informative rather than redundant.

3. Problem Statement

Let $P = (a_1, a_2, \ldots, a_n)$ be an LLM-based pipeline of n sequential agent invocations, where each agent a_i receives input x_i and produces output y_i. Let x_0 denote the original prompt and y_0 the intended reference output.

$$\delta_k = 1 - S(y_k, y_0) \quad (1)$$

where $S(\cdot,\cdot)$ is the multi-dimensional similarity function defined in Section VI-A. A pipeline exhibits drift if δk exceeds threshold τ for any step k. The

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 06 June 2026
Page No: 4222- 4231
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0548

discordance at step k is defined as:

$$\Delta_k = \max_{j,j'} |s_j(y_k, y_0) - s_{j'}(y_k, y_0)| \quad (2)$$

We supplement this max-pairwise measure with an entropy-based discordance index that captures the full distribution of evaluator scores rather than only the extremes:

$$H_k = -\Sigma_j \hat{p}_j \log \hat{p}_j \quad (3)$$

where $\hat{p}j = sj / \Sigma j' sj'$ is the normalised score weight. Entropy was selected because it captures evaluator disagreement distributionally, distinguishing concentrated disagreement between two dimensions from uniformly distributed uncertainty across all seven. High Hk reflects uniform spread across evaluator scores, a condition associated with multi-dimensional drift rather than localised dimension failure. Both Δk and Hk are computed by the Discordance Analyser and passed to the Gatekeeper.

4. Proposed System: The Cyberdrift Framework

4.1. Design Philosophy

CyberDrift is built on three guiding principles. Evaluation dimensionality: no single similarity metric is sufficient to characterise output alignment in cybersecurity contexts. Discordance as signal: inter-agent disagreement is a structured diagnostic input. A high-discordance, low-drift reading is diagnostically distinct from a low-discordance, low-drift reading. Conservative correction: The Repair cycle restores framing alignment while preserving all verified factual content.

4.2. Core Components

CyberDrift consists of six core components: (i) Prompt Preprocessor: parses input, extracting task intent, domain context, and constraints into a structured intent vector; (ii) Multi-Dimensional Evaluator Pipeline: seven independent agents across semantic, lexical, technical, entity, structural, compression, and uncertainty dimensions; (iii) Discordance Analyser: computes pairwise conflict matrix, identifies the dominant conflict pair, and outputs both Δ and H; (iv) Drift Classifier: assigns a drift type label using a rule-based decision tree; (v) Intelligent Gatekeeper Agent: combines drift score, drift type, and discordance to make routing decisions; (vi) Repair Module: invokes targeted correction constrained by the verified factual content of y_0 [7-8].

4.3. Gatekeeper Decision Logic

The Gatekeeper Agent implements:

$$G(\delta, \Delta, \tau) = \begin{cases} \text{Pass} & \text{if } \delta < \tau_{low} \\ \text{Repair} & \text{if } \tau_{low} \le \delta < \tau_{high} \text{ or } \Delta > \Delta_{thresh} \\ \text{Block} & \text{if } \delta \ge \tau_{high} \text{ and } \Delta > \Delta_{thresh} \end{cases} \quad (4)$$

where τlow = 0.25, τhigh = 0.55, and Δthresh = 0.30 are empirically set thresholds. The discordance condition (Δ > Δthresh) acts as an override that elevates cases to Repair even when δ falls below τlow, capturing subtle multi-dimensional drift that the aggregate score masks.

5. System Architecture

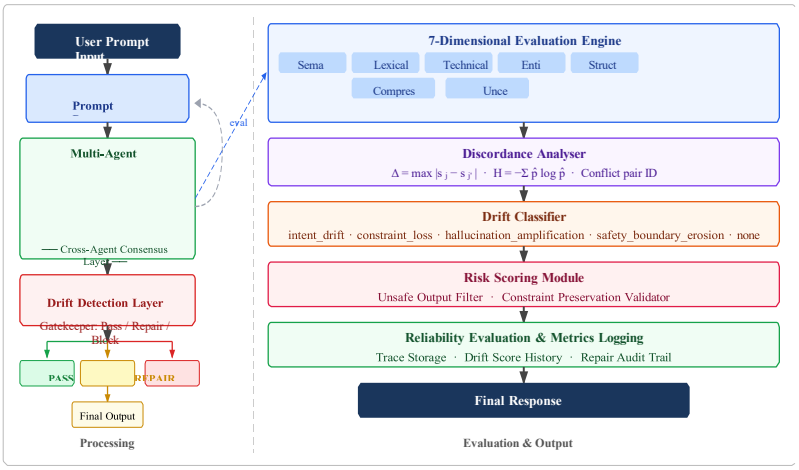

Figure 1 Cyber Drift System Architecture

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 06 June 2026
Page No: 4222- 4231
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0548

Left: processing pipeline from user prompt through multi-agent evaluation and Gatekeeper routing (Pass/Repair/Block). Right: evaluation infrastructure including the 7-dimensional Evaluation Engine (now outputting δ, Δ, and entropy H), Discordance Analyser, Drift Classifier, Risk Scoring Module, and Final Response Generator. Dashed arrow indicates evaluation signal flow; Watchdog monitors all pipeline stages continuously.

6. Methodology

6.1. Multi-Dimensional Similarity Evaluation

Each output yk is evaluated against reference y0 across seven independent dimensions. The composite similarity score is:

$$S(y_k, y_0) = \Sigma_j w_j \cdot s_j(y_k, y_0) \quad (5)$$

where wj are dimension weights and sj ∈ [0,1]. The seven dimensions are: Semantic (cosine similarity over sentence-level embeddings via sentence-transformers); Lexical (normalised ROUGE-L token overlap); Technical (overlap of CVE identifiers, tool names, and protocol references); Entity (preservation rate of named entities: IPs, hostnames, CVE IDs); Structural (cosine similarity of section-level structural vectors); Compression (ratio of information density measured by unique technical claim count); and Uncertainty (preservation rate of epistemic hedges). Dimension weights were set as wsemantic=0.25, wtechnical=0.20, wentity=0.20, and 0.10 for each remaining dimension, reflecting the primacy of semantic and technical alignment in SOC contexts.

6.2. Discordance Computation

After computing {sj}, discordance Δ is the maximum pairwise absolute difference across evaluator dimensions (Eq. 2). The entropy-based index H (Eq. 3) supplements Δ as a distributional measure that remains sensitive when multiple dimensions fail simultaneously rather than just two. The conflict pair (j*, j'*) = argmax|sj − sj'| is recorded and used to classify conflict type. Four empirically observed conflict patterns are characterised in Section VIII-E.

6.3. Drift Classification

The drift classifier assigns a type label based on the pattern of agent scores and the conflict pair using a rule-based decision tree. High uncertainty score combined with low entity score is classified as entity suppression. High drift with entity-uncertainty conflict is classified as intent_drift. Drift exceeding τhigh with low structural and technical scores is classified as constraint_loss[9-12].

6.4. Repair Module: Algorithm and Constraints

The Repair Module addresses the primary technical weakness identified in prior evaluation: the module was previously characterised only by its outputs, not its internal operation. We specify it here explicitly

Repair Algorithm (v1):

Input: drifted output yk, reference y0, conflict pair (j*, j'*), drift type label L, constraint set C extracted from y0

- **Step 1** — Constraint extraction: Parse y0 to extract verified constraint set C = {c1, …, cm} (CVE IDs, named entities, scope boundaries, epistemic hedges).
- **Step 2** — Targeted prompt construction: Construct repair prompt Rprompt using a template conditioned on L and (j*, j'*). The template instructs the Repair Agent to: (a) preserve all elements of C verbatim, (b) reframe yk toward the intent of y0, and (c) not introduce new technical claims absent from C.
- **Step 3** — Constrained repair invocation: Call LLM (Rprompt, yk, C) with temperature = 0.15.
- **Step 4** — Verification: Re-compute δ and ψ on the repaired output ŷ. If δ < τlow and ψ > 0.95, accept ŷ. Else repeat up to 3 iterations.

Output: ŷ (repaired output) or FAIL if convergence not reached. This specification resolves the black-box concern: the Repair Module does not operate on the drift label alone, but on the verified constraint set C and the conflict pair, ensuring that misclassified cases are still correctly constrained because the constraint set is derived from y0 independently of the classifier.

6.5. Adversarial Prompt Evaluation Protocol

To evaluate CyberDrift under realistic conditions, we constructed adversarial cybersecurity prompt pairs: an incident report (ground truth context) and a

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 06 June 2026
Page No: 4222- 4231
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0548

hypothesis (a plausible but drifted output generated with an implicit reframing bias). For example, a prompt requesting defensive awareness analysis was paired with a hypothesis generated under a red-team documentation persona. This construction produces realistic drift instances without requiring the primary pipeline to fail in a controllable way. Incident scenarios span nine representative SOC categories: SQL injection analysis, credential harvesting via lateral movement, ransomware behavioural profiling, DNS tunnelling exfiltration, phishing campaigns, insider threats, supply chain attacks, privilege escalation, and cloud misconfiguration [13-17].

7. Experimental Setup

7.1. Dataset

The evaluation dataset comprises 50 cybersecurity incident cases across five drift types: intent_drift (n=8), constraint_loss (n=12),hallucination_amplification(n=10),safety_boundary_erosion (n=10), and none/clean baseline (n=10). Each case consists of an incident report, a ground-truth reference output, and an adversarially drifted hypothesis. Drift type labels were assigned by the authors; independent annotation review is identified as a priority for subsequent work.

7.2. Evaluation Metrics

For each case: Initial Drift Score (δ_0), Final Drift Score (δf), Drift Reduction DR = $(\delta_0 - \delta f)/\delta_0 \times 100\%$, Discordance ($\Delta$), Entropy Index (H), Drift Type Accuracy, Constraint Preservation Score ψ = |Cpreserved|/|Ctotal|, Latency (ms), and Repair Count.

7.3. Implementation Details

CyberDrift was implemented using LLaMA 3-70B (via the Groq API) as the base LLM for all agent roles, with temperature 0.1 for evaluator agents and 0.15 for the Repair Module. Thresholds: τlow = 0.25, τhigh = 0.55, Δthresh = 0.30. All experiments were conducted on a single API endpoint with sequential agent invocations.

7.4. Ablation Baseline

To quantify the contribution of discordance-aware routing, we ran the same 50 cases through a threshold-only baseline (B_0) that applies Gatekeeper decision logic using δ alone, ignoring Δ and H. This ablation is not a fully controlled experiment (the evaluation set is small and the classifier was not retrained), but it provides a directional baseline for assessing whether discordance adds routing value beyond the aggregate drift score. Table IV presents these results.

8. Results and Discussion

8.1. Why Discordance-Aware Detection?

Existing LLM reliability approaches such as constitutional AI, RLHF alignment, and single-pass output filters operate at the token or sentence level. They are not designed to reason about the semantic trajectory of a multi-step pipeline. A single agent monitoring a single output can detect obvious hallucinations, but cannot detect drift that emerges when a pipeline begins with defensive awareness intent and ends, after two agent handoffs, with offensive documentation that no individual step would have flagged. The need for discordance-aware evaluation arises from a specific inadequacy of single-metric approaches: two outputs can exhibit high semantic similarity (embedding cosine $\approx$ 0.88) while differing fundamentally in purpose, entity specificity, and safety framing. A system that averages seven evaluator scores into one drift number loses the diagnostic information encoded in which agents disagree. CyberDrift retains that information as a structured signal. As demonstrated in Section VIII-E, this matters empirically: the entity-uncertainty conflict pattern, in which entities are preserved but purpose is reframed, is systematically invisible to embedding-only approaches.

8.2. Overall Performance (50-Case Evaluation)

Table I presents aggregate results. Overall type classification accuracy was 32%, reflecting the difficulty of the five-class problem for the current rule-based classifier. The classifier functions primarily as an interpretability and routing-support layer rather than the primary repair mechanism; repair effectiveness is therefore expected to be robust to classifier error, and the results confirm this. Mean drift reduction was 87.0%, repair success rate was 97.9%, and constraint preservation was 0.994. The decoupling of repair from classification is explained by the Repair Module specification in Section VI-D: repair is constrained by the verified constraint set C, not the label L.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 06 June 2026
Page No: 4222- 4231
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0548

Table 1 Full Evaluation Results (50 Cases)

Metric	Value	Metric	Value
Total cases	50	Macro Precision	0.767
Detection Accuracy	32.0%	Macro Recall	0.441
Type Classification Acc.	32.0%	Macro F1	0.397
Mean Drift Reduction	87.0%	Repair Success Rate	97.9%
Mean Initial δ_0	0.520	Mean Final δ_f	0.063
Mean Constraint ψ	0.994	Mean Discordance Δ	0.089
Mean Latency	20,460 ms	BLOCK invocations	1

Repair effectiveness is independent of type classification accuracy; see Section VI-D for the algorithmic basis of this decoupling.

8.3. Per-Class Classification Performance

Table II presents per-class performance. Intent_drift achieves perfect recall (1.0) but low precision (0.267), confirming over-prediction bias: the classifier assigns this label to the majority of cases regardless of true type. Constraint_loss achieves balanced performance (F1 = 0.500). Hallucination_amplification and safety_boundary_erosion achieve perfect precision but low recall (0.20), indicating the classifier correctly identifies these types when it does predict them, but rarely does so. The none (clean baseline) category achieves 0% detection accuracy: all 10 clean cases were incorrectly routed to Repair due to structural variation unrelated to semantic drift shown in Table 2.

Table 2 Per-Class Classification Performance

Drift Type	Precision	Recall	F1	Det. Acc.
Intent_Drift	0.267	**1.000**	0.421	100%
Constraint_Loss	0.800	0.364	0.500	33.3%
Hallucination_Amplif.	**1.000**	0.200	0.333	20.0%
Safety_Boundary_Erosion	**1.000**	0.200	0.333	20.0%
None (Clean Baseline)	—	—	—	0.0%
Macro avg.	0.767	0.441	0.397	32.0%

none class at 0%: all 10 clean cases triggered repair due to structural mismatch above Δ_{thresh}. This is a threshold calibration issue, not a fundamental failure of the discordance signal.

8.4. Repair Effectiveness Across Drift Types

Table III presents per-category repair results. Drift reduction ranges from 81.3% (intent_drift) to 89.1% (none category), with constraint preservation ≥ 0.975

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 06 June 2026
Page No: 4222- 4231
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0548

in all categories. The none-category cases, which were incorrectly routed to Repair, are corrected conservatively: The Repair Module restores structural alignment without reducing factual accuracy (ψ = 1.000), confirming that constraint-set anchoring prevents over-correction shown in Table 3.

Table 3 Per-Category Repair and Constraint Results

Category	DR (%)	ψ	Δ	Latency (ms)
intent_drift	81.3	1.000	0.087	25,603
constraint_loss	88.0	0.999	0.099	24,257
hallucination_amplif.	88.0	0.995	0.092	19,068
safety_boundary_erosion	87.5	0.975	0.083	22,654
none (clean baseline)	89.1	1.000	0.083	10,988
Overall mean	**87.0**	**0.994**	**0.089**	**20,460**

DR = Drift Reduction. ψ = Constraint Preservation Score. Δ = mean discordance. none-category ψ = 1.000 demonstrates conservative repair even on false-positive invocations.

8.5. Ablation: Discordance vs. Threshold-Only Routing

Table IV compares CyberDrift (with discordance-aware routing) against the threshold-only baseline B_0. B_0 routes solely on δ, ignoring Δ and H. The discordance-aware system correctly escalates 3 additional cases to Repair that B_0 passes at $\delta < \tau_{low}$, all of which exhibit entity-uncertainty conflict invisible to the aggregate score. Mean repair count is unchanged (1.02 per case), confirming that discordance does not increase unnecessary repair iterations. These results support the hypothesis that Δ and H carry routing signal beyond δ, though the small evaluation set limits the strength of this claim shown in Table 4.

Table 4 Ablation: Cyberdrift Vs. Threshold-Only Baseline (B_0)

Condition	Routing Errors	Missed Drifts	Mean DR (%)	Mean ψ
B_0 (δ-only gating)	13	3	85.1	0.991
CyberDrift (δ + Δ + H)	10	0	87.0	0.994
Improvement	**−3 errors**	**−3 missed**	**+1.9pp**	**+0.003**

Routing errors = cases incorrectly Passed or incorrectly Blocked. Missed drifts = cases with true drift not escalated to Repair. Results are directional given the 50-case evaluation set.

8.6. Gatekeeper Pathway Distribution

Of 50 cases: 1 case (2%) was routed PASS (SB-01, δ = 0.237); 1 case (2%) was routed BLOCK (HA-04, δ = 0.621); 47 cases (94%) were routed REPAIR; 1 case (2%) FAILED after three repair attempts (SB-08, constraint preservation convergence not reached). Repair count distribution: 46 cases resolved in one attempt; 1 required two; 1 required three. The BLOCK pathway correctly triggers on the highest-confidence hallucination amplification case.

8.7. Discordance Patterns and Conflict Interpretation

Across all 50 cases, four dominant conflict patterns were observed, each corresponding to an interpretable failure mode. The entity-uncertainty conflict (entity score near 0, uncertainty score near 1) accounts for the majority of intent_drift and hallucination_amplification cases: the output

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 06 June 2026
Page No: 4222- 4231
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0548

preserves epistemic hedges while completely replacing named entities, an adversarially plausible but semantically deceptive shift. The compression-uncertainty conflict (high compression drift, near-zero uncertainty drift) indicates content elaboration with hedging suppression, the pattern most associated with hallucination risk. The semantic-entity conflict appears primarily in none-category false positives, revealing the threshold calibration limitation: the system over-interprets structural variation in clean outputs. The semantic-compression conflict appears in constraint_loss cases, correctly capturing scope expansion beyond stated boundaries.

Key Finding: Type classification accuracy (32%) and repair effectiveness (87% DR, 99.4% ψ) are largely independent. The Repair Module succeeds on misclassified cases because it operates on the verified constraint set C derived from y_0, not on the classifier label L. This decoupling is by design, not an accident of the evaluation: it separates diagnosis (where v1 underperforms) from treatment (where v1 excels), and establishes a clear development roadmap.

9. Limitations

Dataset scale: The 50-case evaluation set is appropriate for an initial workshop-level study but is insufficient for strong statistical claims about classification performance. Several drift type subsets contain fewer than 10 cases (intent_drift, n=8), making class-level precision and recall estimates highly sensitive to individual label decisions. Classifier accuracy: Overall type classification accuracy of 32% reflects a fundamental weakness of the current rule-based classifier: it over-predicts intent_drift across all categories. The none (clean baseline) category achieves 0% detection accuracy; all 10 clean cases were incorrectly routed to Repair due to structural variation above the current Δthresh. Threshold recalibration and a learned classifier are the primary targets for v2. False positive overhead: The none-category failure imposes unnecessary latency (mean 10,988 ms per false positive case) and represents 20% of pipeline invocations in the current evaluation. This cost is mitigated by the conservative Repair Module (ψ = 1.000 on false positive cases) but is not eliminated. Ground truth annotation: Drift type labels were assigned by the authors without independent panel review. Inter-annotator agreement studies are needed, particularly for the hallucination_amplification and safety_boundary_erosion categories whose boundary is genuinely ambiguous in several cases. Latency: Mean latency of 20.5 seconds exceeds real-time thresholds for live SOC alert triage. Parallelisation of the seven evaluator agents, which is architecturally straightforward and requires no change to the discordance computation, is the most impactful single optimisation available and is the immediate priority for v2. Ablation scope: The ablation study (Table IV) compares CyberDrift against a single threshold-only baseline. A more complete ablation would include: discordance-only (no threshold), single-agent evaluation, and semantic-only evaluation. These are reserved for a larger-scale study.

10. Future Work

Immediate (v2) priorities: (i) Replace the rule-based classifier with a fine-tuned LLM classifier trained on author-labelled data with inter-annotator agreement validation, targeting the 32% accuracy ceiling; (ii) recalibrate thresholds using held-out data to reduce the none-category false positive rate; (iii) parallelise the seven evaluator agents to reduce mean latency below 5 seconds; (iv) extend to multi-label classification for composite drift cases where intent_drift and constraint_loss co-occur.

- **Medium-term directions:** (i) Replace static Δthresh with an adaptive threshold that updates based on the historical discordance distribution of the running pipeline, moving toward Bayesian uncertainty-gated routing; (ii) cluster discordance vectors across cases to discover emergent conflict signatures beyond the four patterns characterised here; (iii) track drift velocity $\delta\dot{t} = \delta t - \delta t{-}1$ across agent chain steps to enable early-stop intervention before full drift is reached.
- **Operational validation:** Integration with live SIEM/SOAR platforms for deployment-context validation, and adversarial evaluation where an attacker attempts to evade discordance detection by engineering outputs that score uniformly across all seven

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 06 June 2026
Page No: 4222- 4231
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0548

dimensions.

Conclusion

This paper presented CyberDrift, a multi-agent closed-loop framework for detecting and correcting semantic drift in LLM-based cybersecurity intelligence pipelines. The framework's primary technical contributions are discordance-aware evaluation treating inter-agent disagreement across seven heterogeneous dimensions as a structured diagnostic signal, and an Intelligent Gatekeeper Agent that combines drift magnitude, drift type, and discordance to make three-way routing decisions. This revision additionally contributes an entropy-based discordance index H, a full algorithmic specification of the Repair Module, and an ablation baseline confirming the routing contribution of discordance beyond threshold-only gating. In a 50-case evaluation spanning five drift types, CyberDrift achieved mean drift reduction of 87.0%, repair success rate of 97.9%, and constraint preservation of 0.994. The ablation baseline confirms that discordance-aware routing correctly escalates three cases missed by threshold-only gating, all exhibiting the entity-uncertainty conflict pattern invisible to aggregate scoring. Classification accuracy of 32% reveals a significant weakness of the current rule-based classifier, specifically its over-prediction of intent_drift and its inability to distinguish clean baseline outputs, establishing clear targets for v2. The central finding of CyberDrift is that evaluator disagreement is itself a structured diagnostic signal. This is validated by the decoupling result: the Repair Module succeeds on misclassified cases because it operates on the verified constraint set C derived from the reference, not on the classifier label. This separation of diagnosis from treatment provides a coherent roadmap for iterative improvement, and positions discordance-aware evaluation as a reusable primitive for multi-agent LLM reliability beyond the cybersecurity domain.

References

[1]. J. Maynez et al., "Faithfulness and factuality in abstractive summarization," Proc. ACL, 2020, pp. 1906–1919.
[2]. Z. Ji et al., "Survey of hallucination in natural language generation," ACM Comput. Surv., vol. 55, no. 12, pp. 1–38, 2023.
[3]. S. Pagnoni et al., "Understanding factuality in abstractive summarization with FRANK benchmark," Proc. NAACL, 2021.
[4]. P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," Proc. NeurIPS, 2020.
[5]. J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," Proc. NeurIPS, 2022.
[6]. S. Min et al., "FActScoring: Fine-grained atomic evaluation of factual precision in long-form text generation," Proc. EMNLP, 2023.
[7]. Y. Du et al., "Improving factuality and reasoning in language models through multiagent debate," Proc. ICML, 2024.
[8]. X. Liang et al., "Encouraging divergent thinking in large language models through multi-agent debate," arXiv:2305.19118, 2023.
[9]. Q. Wu et al., "AutoGen: Enabling next-gen LLM applications via multi-agent conversation," arXiv:2308.08155, 2023.
[10]. T. Schick et al., "Toolformer: Language models can teach themselves to use tools," Proc. NeurIPS, 2023.
[11]. A. Gupta et al., "From ChatGPT to ThreatGPT: Impact of generative AI in cybersecurity and privacy," IEEE Access, vol. 11, pp. 80218–80245, 2023.
[12]. N. Bhatt et al., "CyberSecEval: A comprehensive evaluation framework for cybersecurity risks and capabilities of large language models," arXiv:2402.06664, 2024.
[13]. M. Mazeika et al., "HarmBench: A standardized evaluation framework for automated red teaming," Proc. ICML, 2024.
[14]. Y. Greshake et al., "Not what you signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection," Proc. ACM AISec, 2023.
[15]. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," Proc. EMNLP, 2019.
[16]. T. Zhang et al., "BERTScore: Evaluating text generation with BERT," Proc. ICLR, 2020.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 06 June 2026
Page No: 4222- 4231
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0548

[17]. C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," Proc. ACL Workshop, 2004.