

Article

Insider Threat Detection and Evidence-Grounded Triage Using a Heterogeneous Dual-Branch Network and a Constrained LLM Agent

Kai Cheng ^{1,*}, Dongkun Li ¹, Weidong Tang ¹, Yi Shu ² and Weizhong Qiang ^{2,*} ¹ State Grid Corporation of China Central China Branch, Wuhan 430077, China; lidk1020@163.com (D.L.); tang-wd@139.com (W.T.)² School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China; yshu9266@gmail.com

* Correspondence: kaychang1988@163.com (K.C.); wzqiang@hust.edu.cn (W.Q.)

Abstract

Insider threats are difficult to detect and investigate because risky behavior can occur through valid accounts, develop slowly, and leave mixed behavioral traces. We study an analyst support architecture that links anomaly detection with evidence-based triage. Rather than treating insider threat analysis as a single classification task, the architecture organizes it as a workflow that combines behavior modeling, evidence retrieval, rule interpretation, and constrained reasoning. For anomaly detection, we build a multi-view representation over subject time windows and design a heterogeneous dual-branch network (HDBN) that fuses semantic and statistical branches through confidence-gated fusion. The detector captures temporal event patterns and statistical deviations while retaining structured alert clues for downstream investigation. For investigation, we construct a dual-track evidence infrastructure for LLM-assisted reasoning: a star schema data mart supports structured fact verification and a rule tree index supports policy and rule interpretation. An intent-aware hybrid retrieval mechanism routes sub-questions to the relevant evidence source and aligns structured facts with unstructured rules. We also design a constrained reasoning agent based on a controlled reasoning and acting workflow. Action-space constraints, evidence-completeness gating, and state recovery are used to limit reasoning drift and unsupported decisions when enterprise evidence is incomplete. On the CERT r4.2 dataset, the HDBN reaches 96.8% recall at the selected operating point. In the constructed investigation tasks, the constrained ReAct agent reaches 91.8% end-to-end triage accuracy compared to an unconstrained LLM agent baseline under the same tool access setting. The results support the feasibility of traceable detection-to-investigation workflows in the evaluated benchmark setting, while deployment on real enterprise data and adversarial scenarios remains to be validated.



Academic Editor: Grigorios Beligiannis

Received: 2 June 2026

Revised: 30 June 2026

Accepted: 9 July 2026

Published: 14 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.**Keywords:** insider threat detection; LLM-assisted investigation; hybrid retrieval; constrained LLM agent

1. Introduction

As enterprise workflows extend beyond conventional intranet boundaries, security monitoring must address identity abuse, privilege misuse, and anomalous insider behavior alongside external intrusions [1,2]. Industry reports and academic surveys emphasize that

insider activity is difficult to detect and assess because it often occurs under legitimate identities, manifests intermittently, and may persist for extended periods [3–5].

Previous research has transitioned from feature-engineered machine learning methods for the detection of numerical deviations to deep models, including LSTM- and Transformer-based architectures for the modeling of sequential behavior [6–8]. However, two critical challenges remain for systems designed for security analysts: First, single-view models may fail to capture both temporal semantic shifts and statistical deviations in heterogeneous logs, which can exacerbate false-positive and false-negative rates at different operating thresholds [9,10]. Second, many detectors return only anomaly scores or class labels. Consequently, they fail to preserve sufficient contextual evidence for triage, leaving a gap between the discovery of anomalies and subsequent investigation.

Recent studies have examined large language models (LLMs) for security analysis and alert triage [11,12], including Chain of Thought prompting [13] and reasoning and acting paradigms [14]. Nevertheless, the reliable application of unconstrained LLM workflows in complex enterprise investigations remains challenging. Executing queries across structured databases and unstructured compliance rules can result in schema linking errors, incomplete evidence retrieval, or unsupported factual claims. Without explicit process control, agents navigating multi-step analytical chains are prone to reasoning drift, where they deviate from the original investigation target, execute redundant tool invocations, or reach premature conclusions based on fragmented or insufficient evidence [15].

This study proposes a method for the analysis of insider threats constructed around a unified detection-to-investigation workflow. This approach formulates the analysis as a structured process of clue extraction, automated evidence acquisition, rule interpretation, and constrained reasoning. A heterogeneous dual-branch anomaly detection network (HDBN) first identifies suspicious subject time windows and generates structured alert clues. A dual-track evidence infrastructure and an intent-aware hybrid retrieval mechanism then supply multi-modal evidence for LLM-assisted investigation. Finally, state and action constraints within the reasoning agent guide the generation of traceable, evidence-grounded triage outputs, explicitly excluding automated enforcement, disciplinary action, or legal attribution.

The primary contributions of this study are summarized as follows:

- Detection-to-investigation formulation with HDBN clue generation. We formulate the analysis of insider threats as a detection-to-investigation workflow and design the HDBN to connect anomaly detection with downstream investigation. The semantic and statistical branches model event sequence patterns, distributional deviations, and relational behavior, while a confidence-gated fusion mechanism facilitates high-recall alert generation and produces structured clue objects that retain anomaly sources, behavioral bases, related objects, and log tracking fields.
- Dual-track evidence acquisition for fact–rule verification. We construct a dual-track evidence infrastructure that organizes structured audit facts in a star schema data mart and policy and rule texts in a rule tree index. Intent-aware routing and fact–rule alignment coordinate the retrieval of facts via Text-to-SQL with vector-based rule retrieval, allowing investigation sub-questions to be grounded in both quantitative audit evidence and applicable rules.
- Constrained ReAct triage as a prototype workflow. We introduce a constrained ReAct agent with action-space masks, evidence-completeness gating, and state recovery to mitigate reasoning drift and unsupported decisions. Through evaluation on the CERT r4.2 dataset, the proposed architecture serves as a robust benchmark and prototype workflow, laying a strong foundation for traceable and evidence-grounded insider threat investigation in enterprise environments.

2. Related Work

Recent surveys have traced the evolution of insider threat analytics from feature-based models to deep learning, graph modeling, and LLM-assisted cybersecurity analysis [5–7,12,16]. The recent literature published between 2024 and 2026 further emphasizes multi-view behavior representation, temporal heterogeneous context, imbalanced data handling, evidence discovery, and agent security evaluation [17–22]. Five lines of research are particularly relevant here: behavioral risk modeling, graph-based enterprise context modeling, deep log representation learning, benchmark-aware evaluation, and LLM-assisted security investigation. These directions position the proposed architecture as a workflow that links anomaly detection, evidence acquisition, and constrained reasoning.

We use *insider threat detection and investigation* to refer to the detection-to-triage workflow studied in this paper rather than to a new threat category. This terminology aligns with operational programs for insider threats: CERT frames the management of insider threats around prevention, detection, and response, while the NICE Insider Threat Analysis work role includes the identification and assessment of cybersecurity insider threat activities and the production of findings that support investigations [1,23]. In this paper, *detection* refers to the model-based identification of suspicious subject time windows, and *investigation* encompasses evidence collection, rule interpretation, and triage for those windows. This scope is narrower than legal attribution or enforcement but broader than mere anomaly classification.

2.1. Behavioral and Graph-Based Insider Threat Detection

Research on insider threats originates from a practical challenge: malicious activities are often executed via legitimate identities and therefore cannot be distinguished from normal enterprise behavior by perimeter controls alone [1,5]. The CERT program formalized representative insider scenarios and synthetic audit data generation, enabling the controlled evaluation of detection algorithms on user behavior traces modeled after enterprise environments [24]. Ontology-based studies structure insider indicators such as role, access, device, and behavioral context, providing a conceptual foundation for feature construction and rule interpretation [2].

Traditional machine learning approaches typically transform audit logs into manually engineered behavioral features. For instance, Hidden Markov models capture transitions among user activities, Bayesian networks are applied to risk prediction, and empirical studies demonstrate that detection performance depends strongly on temporal and behavioral granularity [25–27]. These methods are often interpretable and computationally efficient, but they rely heavily on feature engineering and fixed aggregation windows. Consequently, they may fail to detect attacks that unfold slowly, aggregate weak signals across systems, or mimic routine business operations.

Graph-based methods partially alleviate these limitations by modeling users, hosts, files, and operations as relational structures. Log2vec introduced heterogeneous graph embeddings for the detection of enterprise cyber threats, and subsequent studies on insider threats employed daily activity graphs or graph neural networks to represent access paths, peer relationships, and role deviations [16,28,29]. Attention-based graph networks facilitate the weighting of heterogeneous neighbors within enterprise-entity graphs [30]. Recent temporal heterogeneous and group-aware models further integrate the relations between users and entities, behavior sequences, and group baselines, indicating a shift from isolated user profiles toward contextual enterprise modeling [21,31]. Evidence-oriented approaches such as Ripple2Detect also demonstrate a growing interest in the discovery of multi-step insider evidence rather than the mere assignment of anomaly labels [20]. However, most graph and evidence-mining approaches remain optimized primarily for detection, ranking, or ev-

idence discovery; they rarely specify how alerts are converted into structured investigation targets, verified against relational audit facts, and validated against institutional rules. Furthermore, graph construction and real-time inference can introduce computational overhead and cold-start issues that vary depending on the deployment setting.

2.2. Deep Sequential and Semantic Representation Learning

Deep learning approaches are extensively utilized to model temporal and semantic patterns in security logs. General log anomaly methods such as DeepLog and LogBERT demonstrate that recurrent and Transformer-based models can learn event sequence regularities and detect deviations without extensive manual rule engineering [32,33]. In the context of insider threat detection, unsupervised LSTM autoencoders and LSTM-based anomaly detectors have been employed to model structured cybersecurity streams and user behavior sequences [8,34]. These models enhance the learning of temporal dependencies, but their outputs predominantly consist of anomaly scores or class labels rather than structured evidence objects formatted for subsequent investigation.

Recent models for insider threats incorporate semantic, multi-granularity, image-based, text-aware, and heterogeneous representations. ITDBERT adapts concepts from pre-trained language models for the temporal-semantic representation of insider behavior, SeqA-ITD augments user behavior sequences at multiple time granularities, and RAP-Net captures resource access patterns for the detection of insider threats [35–37]. Recent studies based on statistical-sequential methods, contextual image features, and natural language processing confirm that heterogeneous user traces can be represented from multiple semantic perspectives; nevertheless, these representations are typically evaluated as classifiers rather than as interfaces for investigation [17,18,38,39]. Double-layer detection architectures and log text analysis methods further integrate behavioral patterns with textual or semantic information to improve detection capabilities across heterogeneous inputs [9,10]. Recent graph- and LLM-oriented studies also highlight the potential of temporal heterogeneous graphs and multi-view reasoning over behavioral evidence [31,40].

Despite these advances, most deep models remain heavily detection-centric. They improve recall, precision, or ranking metrics, but they rarely establish a clear interface from the discovery of anomalies to evidence-grounded investigation. For the support of human analysts, the objective is not merely to detect an abnormal window, but also to preserve the behavioral semantics, statistical deviations, and contextual clues required for downstream triage. Our approach bridges this gap by employing a heterogeneous dual-branch network to model semantic associations and statistical deviations and subsequently exporting these clues as structured formats to the investigation agent.

2.3. Datasets, Benchmark Bias, and Evaluation Constraints

Evaluation remains challenging in the research of insider threats because malicious behaviors are rare, privacy-sensitive, and difficult to observe in actual enterprises. The CERT synthetic corpora are consequently widely adopted for reproducible experiments; however, their scenario scripts, simulated organizational roles, and assumptions regarding event generation constrain the direct transferability of benchmark performance to operational deployment environments [5,24]. Studies focusing on data granularity and imbalanced learning further indicate that window size, aggregation level, the treatment of minority classes, and strategies for data generation can materially impact reported performance [19,27,41]. Consequently, classification accuracy alone is insufficient to evaluate an operational system for insider threats.

Benchmark protocols also typically evaluate the outputs of detection rather than the downstream investigation process. In operational triage, analysts must verify whether

a behavioral clue is supported by audit facts, whether institutional rules are applicable, and whether the final judgment is traceable to concrete evidence. Adversarial and adaptive behaviors further complicate this process because insiders may manipulate event sequences to evade learned detectors or to exploit vulnerabilities in log-based models [42]. Accordingly, our evaluation encompasses anomaly detection alongside the factual validity of Text-to-SQL, hybrid retrieval quality, agent triage accuracy, token consumption, and end-to-end latency.

2.4. LLM-Assisted Security Analysis and Knowledge-Grounded Retrieval

LLMs have been investigated for cybersecurity tasks such as log interpretation, alert triage, threat hunting, and domain-specific reasoning [12]. Prompt-based methods facilitate interpretable online log analysis, while multi-agent systems such as Audit-LLM examine the coordination of LLM agents over log evidence for the detection of insider threats [40,43]. Generative agent-based modeling has also been deployed to simulate insider behavior and support the analysis of scenarios [44]. These studies demonstrate the potential of LLMs for summarizing evidence and reasoning across complex security contexts, but they concurrently highlight the risks associated with unsupported decisions, redundant tool invocations, and reasoning drift within extended multi-step workflows.

The ReAct paradigm establishes a foundation for interleaving reasoning and the utilization of external tools, and broader surveys of agent architectures emphasize planning, memory, tool selection, and execution control [14,15]. In security operations, the invocation of tools must depend on the state of the evidence: the agent must query structured facts when quantitative verification is required, retrieve policies or rules when normative interpretation is necessary, and avoid rendering triage decisions before the requisite evidence is complete. General-purpose LLM agents typically fail to enforce these domain-specific state transitions.

Retrieval-augmented generation enhances factual grounding by incorporating external knowledge into the prompts of LLMs [45]. Cybersecurity knowledge graphs and threat intelligence extraction methods, including AttackKG and AttackKG+, organize tactics, techniques, and procedures for downstream reasoning, while graph-based reasoning frameworks investigate the operation of LLMs over structured knowledge [46–48]. At the operational level, systems such as SPYRAPTOR demonstrate the utility of query support for enterprise threat hunting [49]. Furthermore, most general-purpose RAG pipelines emphasize the retrieval of unstructured text, whereas the investigation of insider threats also demands quantitative verification against relational audit data, including counts, times, devices, resources, and cross-system correlations. Therefore, generic retrieval must be explicitly constrained to prevent the fragmented gathering of evidence and the misalignment of reasoning.

Text-to-SQL research bridges natural language queries with relational databases and facilitates the extraction of such structured facts. Benchmark-driven works such as Spider underscore the challenges of cross-domain schema linking, while decomposed prompting and agent-based repair methods such as DIN-SQL and SQLFixAgent strive to enhance SQL validity and semantic accuracy [50–52]. These methods are applicable to security investigation, but a generic Text-to-SQL module remains insufficient for the comprehensive triage problem: factual database evidence must be aligned with unstructured compliance rules, and the reasoning process necessitates an explicit criterion for evidence sufficiency.

2.5. Research Gap and Positioning of This Study

Five distinct gaps motivate this study: First, numerous detectors for insider threats optimize anomaly classification but fail to generate structured clues for subsequent investigation. Second, single-view or single-branch representations frequently emphasize either semantic

context or statistical deviation, rather than integrating both elements in a confidence-aware manner. Third, benchmark evaluations frequently conclude with detection metrics and neglect the measurement of evidence retrieval or triage quality. Fourth, LLM-based security analysis typically approaches retrieval as a single unstructured context acquisition step, whereas the triage of insider threats mandates coordination between structured audit facts and unstructured rules. Fifth, unconstrained agents executing long logical chains are susceptible to reasoning drift and premature judgments when evidence remains incomplete. These gaps do not imply that the prior works are inadequate within their specific contexts; rather, they highlight the necessity for a system interface tailored to the detection-to-investigation workflow studied here.

Table 1 summarizes the primary streams of the literature and their relevance to this study. Prior work has provided fundamental components for behavior modeling, relational representation, benchmark construction, and language model reasoning, but fewer studies have integrated the requirements highlighted in this work: structured clue generation, benchmark-aware evaluation, hybrid fact–rule evidence acquisition, and constrained multi-step reasoning under explicit scope assumptions.

Table 1. Positioning of the proposed study relative to representative related work.

Literature Stream	Positioning and Gap Addressed in This Study
Behavioral risk modeling [25–27]	Transforms anomalous windows into structured investigation clues, avoiding the treatment of risk scores as terminal outputs.
Graph and relational detection [16,20,21,28,29,31]	Connects relational and multi-step evidence signals to downstream fact–rule verification.
Sequential and semantic deep models [10,17,18,32,33,35]	Utilizes the outputs of a dual-branch detector as triage cues instead of relying exclusively on numerical scores.
Benchmark and deployment evaluation [19,24,41,42]	Evaluates evidence validity, reasoning cost, and latency in addition to detection metrics.
LLM agents and RAG [14,40,45]	Restricts reasoning drift through explicit state transitions, evidence-completeness checks, and recovery rules under defined scope assumptions.
Text-to-SQL and security knowledge graphs [46,50–52]	Coordinates structured SQL fact extraction and unstructured policy rule retrieval within a unified automated investigation workflow.
This study	Combines HDBN clues, a star schema data mart, a rule tree index, hybrid retrieval, and a constrained ReAct agent for traceable insider threat triage in the evaluated CERT-based setting.

This work addresses the intersection of these gaps. Compared with stand-alone detection models, it integrates a heterogeneous dual-branch anomaly detector with investigation-oriented clue representation. Compared with general RAG-style pipelines, it separates structured fact verification from policy interpretation through a star schema data mart and a rule tree index. Compared with unconstrained ReAct agents, it introduces intent-aware hybrid retrieval, action-space constraints, evidence-completeness gating, and state recovery. Together, these components constitute a collaborative detection and investigation architecture that maps abnormal behavior windows to traceable, evidence-grounded triage outputs for insider risks within the stated scope and evaluated setting.

3. Methodology

3.1. Overall Architecture and Formal Task Definition

Insider threat detection and investigation form a continuous analytical process that spans anomaly discovery, clue transfer, evidence acquisition, and evidence-based triage.

We model this process as a two-stage collaborative workflow: The detection stage analyzes multi-source logs and outputs structured alert clues. The investigation stage organizes external evidence around these clues and applies constrained multi-step reasoning to produce analyst-facing, evidence-grounded triage decisions. This interface avoids passing isolated anomaly scores to the investigation stage and limits the raw log data exposed to the agent.

Figure 1 illustrates the workflow. After receiving logs and security data, the anomaly detection layer generates structured alert clues. The clue representation layer converts these outputs into investigation targets with explicit priors, guiding evidence acquisition and reasoning in the LLM-assisted investigation layer. In this way, raw events are mapped to evidence-based triage outputs through an explicit intermediate representation.

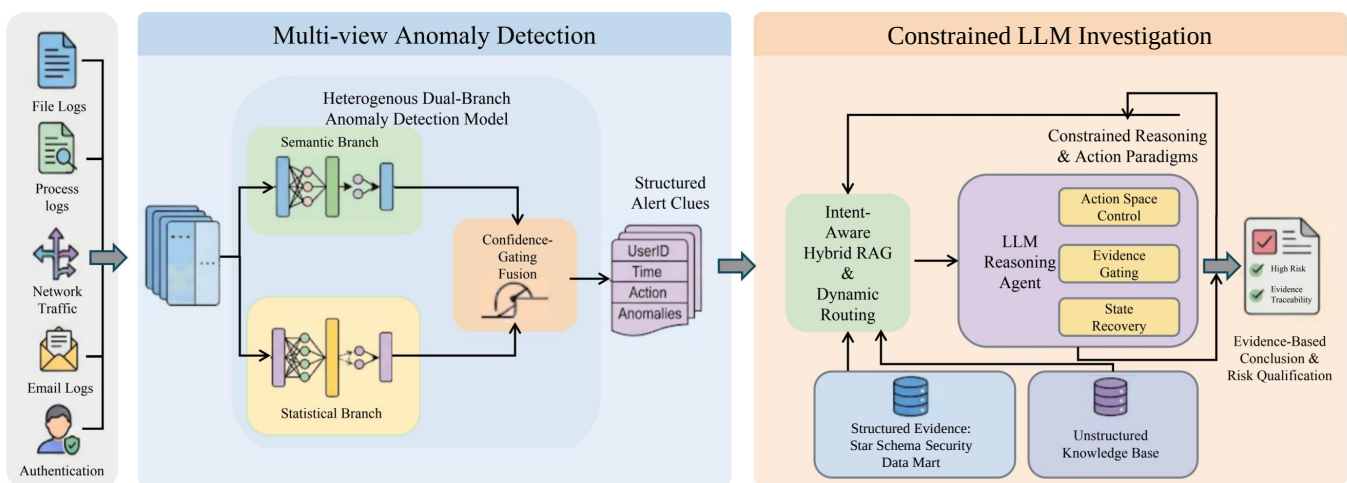


Figure 1. Overall framework of the detection and investigation method.

Formally, the system receives a standardized event set $\mathcal{E} = \{e_1, e_2, \dots, e_N\}$ from heterogeneous logs. Each event e_i is mapped to a six-tuple $(t_i, u_i, o_i, a_i, s_i, z_i)$, representing the timestamp, subject account, object entity, operation, source sensor, and extended attributes, respectively. Because insider threats often appear as temporal behavioral deviations, the event subset $\mathcal{E}_{u,w}$ for subject u within time window $w = [t, t + \Delta]$ is used as the basic modeling unit. The detection layer learns a risk scoring function $f_\theta : \mathcal{E}_{u,w} \rightarrow r_{u,w}$, where $r_{u,w}$ denotes the subject's window-specific anomaly risk score.

The investigation task is then formulated as a constrained evidence-based analysis centered on structured alert clues. Rather than passing only anomaly risk scores, the system integrates the subject identifier, time range, key behavioral basis, related objects, and log tracking fields into a unified clue object:

$$\mathcal{C}_{u,w} = (u, w, r_{u,w}, \gamma_{u,w}, \mathcal{B}_{u,w}, \mathcal{O}_{u,w}, \mathcal{T}_{u,w}). \quad (1)$$

Using this clue as the input anchor, the investigation agent retrieves verifiable facts and rules through the external evidence infrastructure and generates the triage result object:

$$\mathcal{R}_{u,w} = (\mathcal{F}_{u,w}^{fact}, \mathcal{K}_{u,w}^{rule}, \mathcal{D}_{u,w}^{decision}, \mathcal{A}_{u,w}^{action}). \quad (2)$$

The variables in $\mathcal{R}_{u,w}$ represent extracted quantitative evidence, retrieved compliance rules, a qualitative triage decision, and analyst-facing recommended follow-up actions. The formulation maps raw observations to investigation outputs without treating the agent's decision as an automated enforcement decision.

3.2. Threat Model, Scope, and Assumptions

We consider an enterprise insider risk analysis in which suspicious actions are performed through authenticated accounts and are observable in audit logs. The in-scope insider may access systems with legitimate credentials and may try to hide risky behavior within ordinary work patterns but is not assumed to control logging infrastructure, alter historical audit records, poison the rule index, or compromise the LLM service. The defender is assumed to collect normalized audit records, maintain institutional rules or policies as retrievable text, preserve access controls around sensitive evidence, and use the system for analyst support triage rather than real-time access control enforcement or automated punitive action. The evaluation does not cover attacks that leave no observable audit trace, adversarial log tampering, deliberate rule index or retrieval corpus poisoning, prompt injection attacks through retrieved content, sensitive information leakage from prompts or tool outputs, or adaptive evasion against the detector. Those cases require separate integrity controls, privacy controls, and adversarial evaluation.

3.3. Multi-View Behavior Representation and Alert Clue Extraction

Insider threats frequently manifest as temporal semantic inconsistencies or distributional deviations rather than as isolated events. To transform heterogeneous logs into unified model inputs, we construct the event set $\mathcal{E}_{u,w}$ for the subject time window and extract multi-view representations from it. In the statistical view, we calculate individual deviations $d_k^{(ind)}(u, w) = \frac{m_k(u,w) - \mu_k(u)}{\sigma_k(u) + \epsilon}$ and group baseline deviations to account for the effects of roles and workloads. Concatenating these variables with personalized PageRank relational features $PPR_u(v)$, which capture interaction topology [28,29], yields the continuous numerical input $x_{u,w}^{num} = [x_{u,w}^{stat} \oplus x_{u,w}^{rel}]$. To retain sequential event dependencies, we employ a sampling priority $\omega^{(i)} = \gamma \cdot IDF(a_i) + (1 - \gamma) \cdot (1 - PPR_u(o_i))$ for controlled sampling over discrete sequences, which yields the semantic input $\tilde{x}_{u,w}^{sem}$. Together, these paths define the model input $\mathcal{X}_{u,w} = (\tilde{x}_{u,w}^{sem}, x_{u,w}^{num})$.

Utilizing these features, we design a heterogeneous dual-branch anomaly detection model that jointly represents temporal patterns and statistical deviations. A multi-head self-attention semantic branch extracts the sequential features $h_{u,w}^{sem}$, while a residual feed-forward branch extracts the numerical features $h_{u,w}^{num}$. To adapt to different anomaly patterns across windows, an adaptive gating vector $g_{u,w}$ adjusts the weights of the dual branches and generates the fused representation $h_{u,w}^{fus} = g_{u,w} \odot (c_{u,w}^{sem} h_{u,w}^{sem}) + (1 - g_{u,w}) \odot (c_{u,w}^{num} h_{u,w}^{num})$, where $c_{u,w}^{sem}$ and $c_{u,w}^{num}$ denote the branch confidence weights estimated from the semantic and numerical branches. The final window-level anomaly score $r_{u,w}$ combines the supervised discrimination output $\hat{y}_{u,w}$ with normalized dual-branch deviation responses: $r_{u,w} = \beta_1 \hat{y}_{u,w} + \beta_2 \tilde{s}_{u,w}^{sem} + \beta_3 \tilde{s}_{u,w}^{num}$. The non-negative coefficients are empirically calibrated to $\beta_1 = 0.58$, $\beta_2 = 0.27$, and $\beta_3 = 0.15$ through a grid search on the validation set, which balances the primary supervised classification signal with the unsupervised semantic and statistical deviations. This score is utilized for triage prioritization rather than for stand-alone attribution.

The implementation includes a self-supervised pre-training stage on unlabeled log windows before supervised fine-tuning, which matches the ablation setting reported later. Specifically, the pre-training employs a Masked Event Modeling objective to learn normal temporal regularities and a feature reconstruction objective for relational regularities from benign-dominated logs. To mitigate the extreme class imbalance inherent in the CERT r4.2 dataset, this stage utilizes a negative window down-sampling ratio of 1:4.5. The model is pre-trained for 45 epochs using an AdamW optimizer with a learning rate of 2×10^{-4} and an early stopping patience of 10 epochs prior to fine-tuning.

Beyond anomaly scoring, the detection module supplies structured targets for the investigations conducted by the agent by encapsulating the model outputs in the alert clue object $\mathcal{C}_{u,w}$ defined above. The anomaly source summary $\gamma_{u,w}$ indicates whether the risk is driven predominantly by semantic conflict or statistical deviation, which helps initialize the retrieval intent. The key behavioral basis set $\mathcal{B}_{u,w}$ isolates operation chains that contribute to the risk score and maintains the focus of the LLM on relevant log evidence. The associated object set $\mathcal{O}_{u,w}$ and the tracking field set $\mathcal{T}_{u,w}$ constrain the boundaries of fact checking and provide indices for log backtracking. This mechanism converts raw observations into bounded inputs for the LLM-assisted investigation.

3.4. Star Schema Data Mart and Rule Tree Index

The LLM-assisted investigation requires an external evidence infrastructure that provides stable, retrievable inputs for analysis. Evidence of insider threats generally includes structured facts from audit logs for the quantitative verification of behavior and unstructured rules for the interpretation of compliance. To reduce ambiguity across these heterogeneous sources, we construct a dual-track evidence infrastructure.

For the structured fact space, raw security logs are reconstructed into an analysis-oriented star schema data mart $\mathcal{D}_{sql} = (\mathcal{T}_f, \mathcal{T}_d, \mathcal{R})$, where $\mathcal{T}_f$, $\mathcal{T}_d$, and $\mathcal{R}$ denote the fact tables, the dimension tables, and the associative relations, respectively. The granularity of the fact tables is defined at the level of temporal operations between users and assets, with metrics such as data transmission volume, session duration, and access counts being utilized for the quantitative verification of behavior. To manage the changing attributes of subjects, the dimension table for personnel incorporates a Slowly Changing Dimension Type 2 mechanism. The recording of historical versions and effective intervals of attributes, such as resignation handovers or the status of performance improvement, supports queries over temporal slices in complex Text-to-SQL tasks [52].

For the unstructured rule space $\mathcal{D}_{txt}$, the fixed-length chunking via sliding windows is replaced with structured chunking based on rule trees to ensure that the context of clauses is preserved. The system parses compliance documents into a rule tree $\mathcal{J}_i = (V_{tree}, E_{tree})$ according to the document hierarchy and concatenates each leaf clause with the context from the root to the parent to form the rule chunk $c_{i,j}$:

$$c_{i,j} = Node_{root} \oplus \dots \oplus Node_{parent} \oplus Clause_{leaf} \quad (3)$$

A pre-trained encoder maps these chunks into dense vectors $v_{i,j}$. The Hierarchical Navigable Small World algorithm is utilized for efficient approximate nearest-neighbor retrieval in the vector space [53]. During the reasoning process, the cosine similarity between the query vector q_{vec} and the chunk vectors identifies relevant rule evidence:

$$sim(q_{vec}, v_{i,j}) = \frac{q_{vec} \cdot v_{i,j}}{\|q_{vec}\|_2 \|v_{i,j}\|_2} \quad (4)$$

Together, the star schema data mart and the rule tree index provide the dual-track evidence infrastructure for the multi-step investigation conducted by the agent.

3.5. Intent-Aware Hybrid Retrieval-Augmented Mechanism

During the LLM-assisted investigation, sub-questions may necessitate different types of evidence, such as structured facts regarding the scale of access or unstructured rules regarding the responsibilities of roles. To reduce contextual noise and unnecessary retrieval in general RAG-style pipelines [45], an intent-aware hybrid retrieval mechanism routes queries between structured and unstructured sources according to the semantics of the sub-questions, the priors of the clues, and the gaps in the evidence.

For a sub-question q_t , encoding the alert clue object $\mathcal{C}_{u,w}$ yields $c_{u,w} = \mathcal{E}_{clue}(\mathcal{C}_{u,w})$. The concatenation of the representation of the sub-question $e(q_t)$, the clue encoding $c_{u,w}$, and the descriptor of the unmet evidence δ_t forms the routing representation $z_t = [e(q_t) \oplus c_{u,w} \oplus \delta_t]$. A routing classifier outputs a preference distribution over the two evidence spaces: $[\alpha_t^{sql}, \alpha_t^{txt}] = \text{Softmax}(W_r z_t + b_r)$. Given the routing margin threshold τ_{margin} (empirically optimized to 0.165 to balance single-path precision and dual-path recall), the system executes single-path retrieval when one path dominates and concurrent dual-path retrieval when the two probabilities are similar.

Following the dual-path retrieval, the cross-modal alignment of evidence identifies the correspondences between facts and rules. Because structured facts cannot be directly compared with text passages, predefined templates serialize the fact items into natural language descriptions $\tilde{f}_i = \mathcal{T}_{ser}(f_i)$. Serialized facts $\tilde{f}_i$ and rule segments k_j are mapped into a shared semantic space, and the score of their correspondence is defined as $s_{ij} = \cos(h_f(i), h_k(j))$. A correspondence is retained when s_{ij} exceeds the similarity threshold τ_{sim} , which is set to 0.724 in our experiments. The system subsequently prioritizes the aligned pairs of facts and rules to construct the augmented context G_t , thereby providing evidence units for the subsequent reasoning. Algorithm 1 formalizes the procedure for routing and alignment.

Algorithm 1 Intent-Aware Hybrid Retrieval with Fact–Rule Alignment

Require: Sub-query q_t , clue encoding $c_{u,w}$, unmet-evidence descriptor δ_t ; thresholds

$\tau_{sim}, \tau_{margin}$

Ensure: Evidence context G_t for the current reasoning step

```

1:  $\mathcal{F}_t \leftarrow \emptyset, \mathcal{K}_t \leftarrow \emptyset, G_t \leftarrow \emptyset$  ▷ Initialize fact, rule, and context buffers
2:  $z_t \leftarrow [e(q_t) \oplus c_{u,w} \oplus \delta_t]$  ▷ Encode query, clue prior, and evidence gap
3:  $[\alpha_t^{sql}, \alpha_t^{txt}] \leftarrow \text{Softmax}(W_r z_t + b_r)$  ▷ Estimate source-routing probabilities
4: if  $\alpha_t^{sql} - \alpha_t^{txt} > \tau_{margin}$  then
5:    $\mathcal{F}_t \leftarrow \text{Text-to-SQL}(q_t)$  ▷ Retrieve structured audit facts
6: else if  $\alpha_t^{txt} - \alpha_t^{sql} > \tau_{margin}$  then
7:    $\mathcal{K}_t \leftarrow \text{HNSW}(q_t)$  ▷ Retrieve policy or rule evidence
8: else
9:    $\mathcal{F}_t \leftarrow \text{Text-to-SQL}(q_t)$  ▷ Use both sources for ambiguous or hybrid intent
10:   $\mathcal{K}_t \leftarrow \text{HNSW}(q_t)$ 
11: end if
12: if  $\mathcal{F}_t \neq \emptyset \wedge \mathcal{K}_t \neq \emptyset$  then
13:    $H_F \leftarrow \text{Encode}(\mathcal{T}_{ser}(\mathcal{F}_t))$  ▷ Serialize facts and obtain embeddings  $h_f(i)$ 
14:    $H_K \leftarrow \text{Encode}(\mathcal{K}_t)$  ▷ Embed rule segments as  $h_k(j)$ 
15:   for  $i = 1$  to  $|\mathcal{F}_t|$  do
16:     for  $j = 1$  to  $|\mathcal{K}_t|$  do
17:       if  $\cos(h_f(i), h_k(j)) > \tau_{sim}$  then
18:          $G_t \leftarrow G_t \cup \{(f_i, k_j)\}$  ▷ Retain aligned fact–rule evidence
19:       end if
20:     end for
21:   end for
22: else
23:    $G_t \leftarrow \mathcal{F}_t \cup \mathcal{K}_t$  ▷ Return available single-source evidence
24: end if
25: return  $G_t$ 

```

3.6. Multi-Step Reasoning Based on a Constrained ReAct Paradigm

In scenarios involving insider threats, the retrieval of evidence alone does not complete the investigation. The investigation process decomposes the problem, verifies the facts, matches the rules, integrates the evidence, and generates triage decisions around the original anomaly clue. To reduce reasoning drift and unsupported inference during

multi-step generation, we construct a constrained investigation method based on the ReAct paradigm [14], utilizing explicit state representations, a restricted action space, and feedback controls.

The modeling of the LLM-assisted investigation as a controlled sequential decision-making process facilitates maintaining the alignment of long-sequence generation with the investigation target and mitigates known failure modes in current agent architectures [15]. The investigation state at step k is defined as follows:

$$\mathcal{S}^{(k)} = (\mathcal{C}_{u,w}, \mathcal{Q}^{(k)}, \mathcal{F}^{(k)}, \mathcal{K}^{(k)}, \mathcal{H}^{(k)}, \mathcal{J}^{(k)}) \quad (5)$$

These variables represent the fixed clue anchor from the detection layer, the pending sub-question, the collected structured facts, the retrieved rule evidence, the history of reasoning and tool invocations, and the intermediate decisions. The state update follows $\mathcal{S}^{(k+1)} = \Phi(\mathcal{S}^{(k)}, a^{(k)}, O^{(k)})$, where $O^{(k)}$ denotes the observation of the environment following the action $a^{(k)}$.

The method employs three feedback control mechanisms: First, a mask for the action space prunes invalid actions based on the metadata of the state. Given the score vector of the model $z^{(k)}$ over the discrete action space and the mask $\mu^{(k)} \in \{0, 1\}^{|\mathcal{A}|}$, the controlled distribution of actions is

$$\tilde{\pi}^{(k)}(a_i | \mathcal{S}^{(k)}, t^{(k)}) = [\text{Softmax}(z^{(k)} + \log \mu^{(k)})]_i \quad (6)$$

where the masked actions utilize $\mu_i^{(k)} = 0$ and are assigned a probability of zero during implementation, which is achieved equivalently by replacing $\log 0$ with $-\infty$ prior to normalization. Second, a gating mechanism for evidence completeness permits the generation of triage decisions only when the required conditions for evidence are met. The completeness score $\rho^{(k)}$ is defined as follows:

$$\begin{aligned} \rho^{(k)} = & \lambda_1 \cdot \text{Cover}_{fact}(\mathcal{F}^{(k)}) \\ & + \lambda_2 \cdot \text{Cover}_{rule}(\mathcal{K}^{(k)}) \\ & + \lambda_3 \cdot \text{Consis}(\mathcal{J}^{(k)}) \end{aligned} \quad (7)$$

where the empirical weighting coefficients are calibrated to $\lambda_1 = 0.42$, $\lambda_2 = 0.42$, and $\lambda_3 = 0.16$ to emphasize the necessity of the coverage of objective factual and normative evidence over the internal reasoning consistency of the LLM. Third, a state recovery mechanism triggers a fallback strategy $\mathcal{S}^{(k)} \leftarrow \text{Rollback}(\mathcal{S}^{(k^*)}, \Delta_{corr}^{(k)})$ to restore the most recently valid state when the analysis path deviates from the original clue. Algorithm 2 summarizes the constrained investigation process, which operates under a strictly bounded maximum step limit $K_{max} = 12$ and a stringent threshold for completeness $\tau_{finish} = 0.825$ to prevent premature conclusions.

Algorithm 2 Constrained ReAct Workflow for Evidence-Grounded Triage

Require: Clue $\mathcal{C}_{u,w}$, initial sub-query $q^{(0)}$; step limit K_{max} , completeness threshold τ_{finish}
Ensure: Final investigation result $\mathcal{R}_{u,w}$ for analyst review

```

1:  $\mathcal{S}^{(0)} \leftarrow \text{Init}(\mathcal{C}_{u,w}, q^{(0)})$  ▷ Anchor reasoning to the detection clue
2: for  $k = 0$  to  $K_{max} - 1$  do
3:    $t^{(k)} \leftarrow \Pi_{think}(\mathcal{S}^{(k)})$  ▷ Generate task-focused reasoning state
4:    $\mu^{(k)} \leftarrow \Psi(\mathcal{S}^{(k)})$  ▷ Mask actions invalid under current evidence state
5:    $a^{(k)} \leftarrow \arg \max_{a_i \in \mathcal{A}} \tilde{\pi}^{(k)}(a_i | \mathcal{S}^{(k)}, t^{(k)})$ 
6:   if  $a^{(k)} = a_{probe}$  then
7:      $O^{(k)} \leftarrow \text{Hybrid\_Retrieve}(\mathcal{S}^{(k)})$  ▷ Acquire missing fact or rule evidence
8:   else if  $a^{(k)} = a_{fuse}$  then
9:      $O^{(k)} \leftarrow \text{Fuse}(\mathcal{S}^{(k)})$  ▷ Align retrieved facts with applicable rules
10:  else if  $a^{(k)} = a_{conclude}$  then
11:    Compute completeness  $\rho^{(k)}$  ▷ Check fact coverage, rule coverage,
    and consistency
12:    if  $\rho^{(k)} \geq \tau_{finish} \wedge \neg \text{Conflict}$  then
13:       $\mathcal{R}_{u,w} \leftarrow \text{Summarize}(\mathcal{S}^{(k)})$ 
14:      return  $\mathcal{R}_{u,w}$ 
15:    else
16:       $a^{(k)} \leftarrow a_{probe}$  ▷ Defer decision until missing evidence is queried
17:       $O^{(k)} \leftarrow \text{Hybrid\_Retrieve}(\mathcal{S}^{(k)})$ 
18:    end if
19:  else if  $a^{(k)} = a_{fallback}$  then
20:     $\mathcal{S}^{(k)} \leftarrow \text{Rollback}(\mathcal{S}^{(k^*)}, \Delta_{corr}^{(k)})$  ▷ Recover from drift or inconsistent evidence
21:  end if
22:  if  $a^{(k)} \neq a_{fallback}$  then
23:     $\mathcal{S}^{(k+1)} \leftarrow \Phi(\mathcal{S}^{(k)}, a^{(k)}, O^{(k)})$ 
24:  else
25:     $\mathcal{S}^{(k+1)} \leftarrow \mathcal{S}^{(k)}$ 
26:  end if
27: end for
28: return  $\mathcal{R}_{u,w}$  generated from the valid state with the largest completeness score

```

4. Experiments

4.1. Experimental Setup and Evaluation Metrics

The evaluation encompasses the pipeline for detection and investigation, including the experimental environment, the procedure for dataset processing, and the metrics for the two-stage task. The empirical results establish a comprehensive baseline evaluation for the proposed system, demonstrating its core capabilities within the defined operational scope.

4.1.1. Experimental Environment Configuration

The experiments are conducted on a workstation equipped with an NVIDIA GeForce RTX 4090 Laptop GPU (NVIDIA Corporation, Santa Clara, CA, USA), an Intel Core i9-13900HX CPU @ 2.20 GHz (Intel Corporation, Santa Clara, CA, USA), and 64 GB of memory running the Windows 11 operating system (Microsoft Corporation, Redmond, WA, USA), with PyTorch 2.0.1 (Meta Platforms, Inc., Menlo Park, CA, USA) and CUDA 11.8 (NVIDIA Corporation, Santa Clara, CA, USA) as the deep learning framework. Within the dual-track evidence infrastructure, the star schema data mart utilizes ClickHouse 24.11 (ClickHouse, Inc., San Francisco, CA, USA), while the rule tree index employs ChromaDB (Chroma, San Francisco, CA, USA) for vector indexing.

The investigation layer accesses the LLM via a cloud API and employs LangChain 0.1.0 (LangChain, Inc., San Francisco, CA, USA) to orchestrate multi-step tool invocations.

To ensure precise reproducibility and mitigate the randomness of generation, the operational parameters are explicitly defined: the system utilizes the DeepSeek-V2-0628 model snapshot (DeepSeek AI, Hangzhou, China) with a low temperature setting of $T = 0.1$ and a Top- p value of 0.9. The constrained agent operates under a maximum reasoning limit of $K_{max} = 12$ steps and enforces a strict retry policy that permits a maximum of 3 retries per tool invocation upon execution failure. The structured prompt templates for routing and reasoning are strictly defined within the LangChain orchestration pipeline to guarantee standardized JSON outputs.

4.1.2. Dataset Construction and Processing

The proposed method is evaluated on the CERT insider threat benchmark dataset (CERT r4.2), which serves as a widely adopted benchmark for research on insider threats [24]. The dataset simulates an enterprise environment and records 18 months of logs regarding system operations from over one thousand employees. These logs include logins, file accesses, network transmissions, and device insertions. Because the CERT r4.2 dataset is synthetic, it facilitates controlled benchmarking but may not capture the full operational variability present in real enterprise environments.

To address the imbalance of samples and the heterogeneity of event formats, raw logs from multiple sensors are aggregated into subject time windows. The length of the window is set to $T = 24$ h with a sliding stride of 12 h. To prevent data leakage across temporal boundaries, a strict chronological split is enforced: Months 1–12 are allocated for training, Months 13–14 for validation, and Months 15–18 for testing. Any sliding window that overlaps the chronological boundaries is strictly assigned to the subsequent split.

During the training phase, to mitigate the extreme class imbalance, negative windows are down-sampled to maintain a positive-to-negative ratio of 1:4.5, which yields 2450 positive windows and 11,025 negative windows. During the validation and testing phases, the natural distribution of the data is preserved to reflect realistic operational conditions. This approach results in 320 positive and 42,520 negative windows in the validation set and 685 positive and 85,210 negative windows in the test set.

4.1.3. Multi-Stage Evaluation Metrics

To reflect the coupling between the detection of anomalies and the LLM-assisted investigation, we employ a hierarchical evaluation design:

1. **Anomaly Detection Layer Metrics:** We evaluate precision, recall, and the F1-score. Because this layer functions as the initial filter for the agent, recall is emphasized to minimize the risk of missing suspicious windows during the first stage. Candidate false positives are subsequently examined by the investigation layer.
2. **LLM-Assisted Investigation Layer Metrics:** To quantify the multi-step reasoning of the constrained agent and the utilization of external evidence, three operational metrics are employed:
 - **SQL Execution and Logical Validity:** Evaluates whether the Text-to-SQL statements generated by the agent execute correctly and whether the returned results match the manually annotated expected results against the star schema data mart.
 - **Required Path Coverage:** Assesses whether the routing mechanism selects all the evidence paths required by the sub-question; fixed dual-path retrieval provides complete coverage but exhibits lower selectivity.
 - **End-to-End Triage Accuracy:** Measures the agreement between the final triage decision of the agent (true positive, false positive, or insufficient evidence) and the evaluation label.

The automated investigation benchmark comprises 200 Text-to-SQL tasks, 300 routing questions, and end-to-end triage labels. The ground-truth construction followed a rigorous double-blind annotation process conducted by two independent security analysts. Disagreements regarding the expected ground-truth answers or the required evidence paths were adjudicated by a senior reviewer. The inter-annotator agreement achieved a Cohen Kappa score of $\kappa = 0.86$, confirming high label consistency across the evaluated tasks.

4.2. Performance Evaluation of the Heterogeneous Dual-Branch Detection Model

This subsection evaluates the heterogeneous dual-branch network (HDBN) as the initial filter for anomalies. We compare the HDBN with six representative baselines: models based on sequences and reconstruction errors (S-LSTM [41], LSTM-AutoEncoder [34]), the architecture of the pre-trained language model ITDBERT [35], and recent systems for the detection of insider threats (SPYRAPTOR [49], SeqA-ITD [36], RAP-Net [37]). Given the inherent variations in preprocessing procedures, chronological splits, and protocols for threshold selection across the original studies, the reported baseline metrics serve as contextual reference points rather than strictly controlled state-of-the-art comparisons.

4.2.1. Comparative Analysis of Baseline Models

Table 2 presents the quantitative results. To account for the randomness of model initialization and to ensure statistical robustness, the results of the HDBN are reported as the mean $\pm$ standard deviation across five repeated runs. The HDBN achieves the highest recall in this reference comparison, reaching $96.8\% \pm 0.4\%$, higher than S-LSTM [41] (96.0%) and SeqA-ITD [36] (95.6%). This result aligns closely with the objective of our two-stage pipeline: maintaining high recall at the detection filter stage to ensure that weak but potentially malicious anomalies are preserved for downstream evidence-based triage.

The precision ($88.4\% \pm 0.8\%$) and F1-score ($92.4\% \pm 0.5\%$) of the HDBN are lower than those of certain high-precision baselines; for instance, RAP-Net [37] reports an F1-score of 93.8%. Within our unified architecture, this represents a deliberate design trade-off. To substantiate the operational feasibility of this recall-first strategy, we translate the precision of 88.4% into absolute false-positive volume. Based on the 685 positive and 85,210 negative windows within the 4-month test set (detailed in Section 4.1.2), a recall of 96.8% identifies approximately 663 true positives. Consequently, a precision of 88.4% yields roughly 87 absolute false-positive windows over the entire 4-month testing period (fewer than one false alarm per day). Given the 10.1-second end-to-end processing latency of the LLM investigation agent (evaluated in Section 4.5), absorbing and resolving this daily volume of false positives necessitates minimal computational overhead. This calculation demonstrates that the automated investigation layer can seamlessly process the candidate alerts generated by the high-recall filter.

Table 2. Performance comparison of different detection models.

Model	Precision	Recall	F1-Score
S-LSTM	94.9%	96.0%	95.5%
ITDBERT	93.0%	91.9%	92.4%
LSTM-AutoEncoder	97.0%	92.0%	94.0%
SPYRAPTOR	91.0%	89.0%	90.0%
SeqA-ITD	95.8%	95.6%	95.6%
RAP-Net	95.7%	92.0%	93.8%
HDBN (Ours, 5 runs)	$88.4\% \pm 0.8\%$	$96.8\% \pm 0.4\%$	$92.4\% \pm 0.5\%$

4.2.2. Ablation Study of Core Components

To assess the contribution of each module within the HDBN, we conduct an ablation study while maintaining an approximately comparable parameter scale across all variants.

Four variants are tested: removing the statistical branch (without Stat), removing the semantic branch (without Sem), replacing the dynamic gating mechanism with static equal-weight concatenation (without Gating), and removing the self-supervised learning module (without SSL). Figure 2 compares these variants with the complete HDBN.

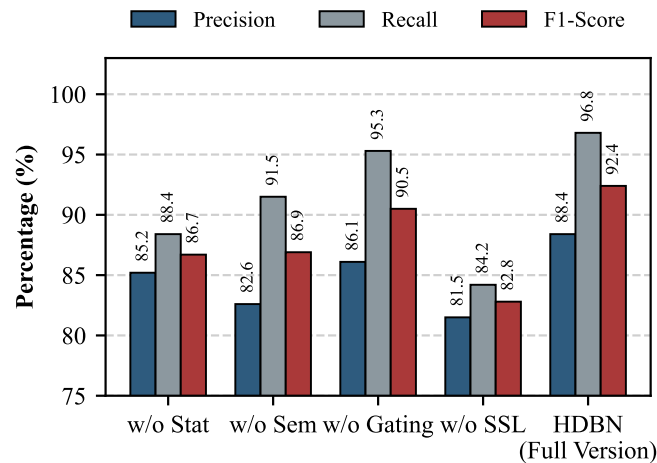


Figure 2. Ablation performance of components within the HDBN.

The ablation results demonstrate the estimated contribution of each module. The removal of the self-supervised learning module (without SSL) causes the most significant decrease, reducing the F1-score to 82.8% (a drop of 9.6 percentage points). Because benign behaviors dominate enterprise logs and malicious samples are scarce, pre-training on unlabeled logs facilitates the modeling of normal workflow patterns and provides a stronger initialization for the detection of anomalies.

Single-view variants correspondingly exhibit inferior performance, achieving F1-scores of 86.7% for the variant without Stat and 86.9% for the variant without Sem. Behaviors associated with insider threats can manifest in various forms: accessing sensitive repositories outside normal working hours may indicate a temporal semantic inconsistency, whereas unusually frequent file downloads within nominal permissions may indicate a statistical deviation. These ablation results indicate that the semantic and statistical branches provide complementary signals.

The removal of the dynamic gating mechanism (without Gating) decreases the F1-score by 1.9 percentage points relative to the complete HDBN. This result implies that static concatenation is less effective for heterogeneous fusion within this context. Dynamic gating adapts the weights of the branches to the dominant anomaly pattern in each window and contributes to more stable initial detection within the evaluated configuration.

4.3. Performance Gain of the Star Schema Data Mart in Fact Extraction

During the LLM-assisted investigation, the agent frequently extracts structured factual evidence from the ClickHouse database via Text-to-SQL. Conventional third normal form (3NF) security log tables may contain complex snowflake-style foreign key associations, which can exacerbate schema linking errors and join ambiguity for SQL statements generated by the LLM. Consequently, we evaluate whether the reconstructed star schema data mart enhances fact extraction within the constructed query set.

4.3.1. Construction of the Test Set and Evaluation Metrics

To simulate scenarios for the investigation of insider threats, we construct a Text-to-SQL test set comprising 200 natural language queries based on the log structure of CERT r4.2 and operational investigation requirements. The test cases are categorized into three

task types according to the complexity of the intent: Single-dimensional basic aggregation evaluates single-point queries over fact tables, such as the calculation of removable device insertions by a specific user within a given time window. Multi-dimensional cross-filtering necessitates conditions across multiple dimensions, such as time and organizational units; an example involves querying research and development employees who accessed a code repository outside of normal working hours. Implicit logical comparison represents the most challenging type within this test set because it requires temporal reasoning over personnel status and Slowly Changing Dimension (SCD Type 2) records. An example of this involves the comparison of the average daily sizes of outgoing email attachments before and after a resignation handover.

The inclusion of the “implicit logical comparison” category reflects a rigid operational requirement in enterprise security investigations. Specifically, the tracking of historical changes in personnel status—such as the identification of anomalous data transfers during the window of a resignation handover—is a core threat vector explicitly modeled within the CERT r4.2 dataset (i.e., Scenario 2). Traditional 3NF architectures frequently struggle with these queries because they typically overwrite historical role attributes, whereas the proposed star schema natively preserves these temporal slices. Evaluating this task therefore measures the capability of the system to handle genuine operational complexities.

We utilize two evaluation metrics: The execution success rate represents the proportion of generated SQL statements that execute in ClickHouse without syntax errors. The logical validity rate represents the proportion of returned result sets that perfectly match the manually annotated expected results, which were rigorously validated through the double-blind adjudication procedure detailed in Section 4.1.3.

4.3.2. Comparative Results and Analysis of Error Distribution

Utilizing the identical LLM interface, we compare the query performance on the traditional 3NF log database and the star schema data mart. Table 3 presents the results.

Table 3. Text-to-SQL performance comparison under different database architectures.

Architecture	Query Type	Samples	Execution Success (%)	Logical Validity (%)
Traditional 3NF	Single-dimensional basic aggregation	80	87.5	85.0
	Multi-dimensional cross-filtering	60	71.6	61.7
	Implicit logical comparison	60	36.6	23.3
	Weighted Average	200	67.5	59.5
Star Schema Data Mart	Single-dimensional basic aggregation	80	98.7	97.5
	Multi-dimensional cross-filtering	60	96.7	93.3
	Implicit logical comparison	60	91.7	88.3
	Weighted Average	200	96.0	93.5

The two architectures differ moderately on tasks involving single-dimensional aggregation (a difference of 12.5 percentage points in logical validity). However, as the dimensionality of the query increases, the logical validity rate of the 3NF architecture declines sharply. For implicit logical comparisons, the 3NF architecture achieves 23.3%, whereas the star schema data mart reaches 88.3%. Overall, the star schema data mart increases the weighted average logical validity from 59.5% to 93.5% within this test set.

It should be explicitly noted that this comparison contains a confounding factor regarding historical data retention. The “implicit logical comparison” query set is inherently designed to require temporal slicing. Because the evaluated 3NF baseline overwrites historical role attributes while the star schema data mart preserves them via SCD Type 2 tracking, the substantial performance gap (23.3% to 88.3%) in this category largely

reflects the presence versus absence of history tracking rather than an intrinsic structural advantage of the star schema itself. Since SCD Type 2 tracking is not strictly exclusive to star schemas, the headline improvement from 59.5% to 93.5% should be interpreted with this explicit caveat.

To investigate the causes of these performance disparities, we manually classify the erroneous SQL statements generated by the LLM. Figure 3 illustrates the distribution of error types for SQL generation under the two architectures.

Multi-table Association Error Aggregation Granularity Confusion
Field Identification Error

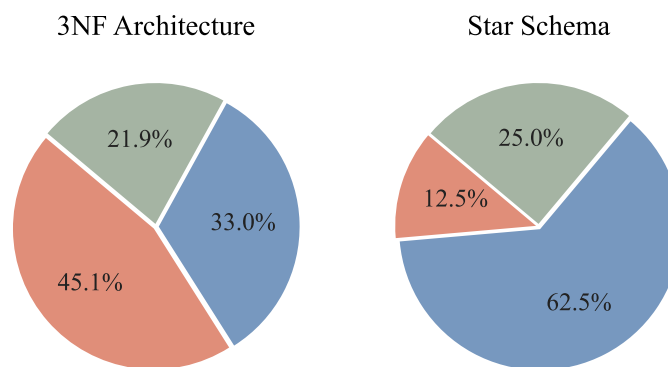


Figure 3. Distribution of error types for SQL generation under different database architectures.

As illustrated in Figure 3 (left), the primary bottleneck under the 3NF architecture lies in the resolution of complex entity relationships. Errors associated with multi-table association account for 45.1% of the failures, followed by errors in field identification (33.0%) and confusion regarding aggregation granularity (21.9%). The snowflake-style foreign key topology exacerbates the likelihood of multi-table JOIN errors when constraints specific to the schema are absent.

Conversely, the star schema data mart mitigates the depth of the data hierarchy. As demonstrated in Figure 3 (right), errors related to multi-table association decrease to 12.5%. Although the relative proportion of errors in field identification increases to 62.5%, the absolute error base is significantly diminished, which is consistent with the enhanced success rates presented in Table 3. Consequently, residual errors in field names constitute the largest proportion of the remaining errors. Furthermore, the SCD Type 2 design within the personnel dimension table provides explicit temporal constraints, thereby enabling complex logical comparisons through a single JOIN or a single-table WHERE filter. Taken together, the distribution of errors suggests that the observed improvement arises from the combination of dimensional flattening, which reduces join depth and mitigates schema linking errors, and SCD Type 2 temporal slicing, which preserves the historical attributes required by temporal queries. These two contributions are not separately ablated in this study, and we therefore refrain from attributing the gain to the star schema reorganization alone. In addition, the Text-to-SQL evaluation uses deterministic decoding (temperature 0) over the 200 fixed queries, so the values reported in Table 3 are single-point measurements rather than mean $\pm$ SD.

4.4. Evaluation of the Collaborative Effectiveness of Hybrid Routing and the Constrained ReAct Agent

4.4.1. Accuracy Analysis of Intent-Aware Hybrid Routing

To reduce the inclusion of irrelevant context and unnecessary token consumption resulting from redundant retrieval, the investigation module employs intent-aware hybrid routing. For validation purposes, we construct a routing set comprising 300 sub-questions: 120 for the confirmation of facts, 90 for the verification of compliance rules, and 90 requiring hybrid verification.

The intent-aware routing strategy is compared against two baselines: The first baseline is unconstrained LLM routing, which relies on zero-shot tool selection. The second baseline is fixed dual-path parallel retrieval, which simultaneously triggers Text-to-SQL and vector retrieval.

The evaluation metrics include the required path coverage, the irrelevant retrieval rate (defined as the proportion of retrieved evidence that is not relevant to the sub-question), and the average number of retrieval invocations. Table 4 summarizes the results.

Table 4. Retrieval and routing performance comparison of different routing strategies.

Routing Strategy	Question Type	Test Samples	Required Path Coverage (%)	Irrelevant Rate (%)	Average Invocations
LLM Routing	Fact Confirmation	120	78.3	21.7	1.18
	Compliance Rule	90	65.6	28.9	1.27
	Hybrid Verification	90	41.1	3.8	1.71
	Weighted Total	300	63.3	18.5	1.37
Fixed Dual-Path Parallel	Fact Confirmation	120	100	50.0	2.00
	Compliance Rule	90	100	50.0	2.00
	Hybrid Verification	90	100	0.0	2.00
	Weighted Total	300	100	35.0	2.00
Intent-Aware Routing	Fact Confirmation	120	97.5	2.5	1.04
	Compliance Rule	90	97.8	2.2	1.05
	Hybrid Verification	90	96.7	1.1	1.96
	Weighted Total	300	97.3	2.0	1.32

The selection of paths via zero-shot LLMs is less reliable in this setting. This approach achieves an overall required path coverage of 63.3% and experiences a decrease to 41.1% on questions involving hybrid verification. For the fixed dual-path retrieval, the coverage of 100% presented in Table 4 denotes complete route coverage rather than the correct discrimination of intent. The strategy utilizing fixed dual paths queries both evidence paths but introduces additional overhead. Because this strategy consistently utilizes two retrieval invocations per query, the irrelevant retrieval rate reaches 35.0%, thereby increasing the volume of irrelevant context transmitted to the agent.

In contrast, the intent-aware routing achieves an overall required path coverage of 97.3% with lower retrieval overhead compared with the fixed dual-path retrieval. For requirements involving a single path, such as retrieval exclusively for facts or exclusively for rules, the average number of invocations approaches 1 (1.04 and 1.05, respectively). The weighted irrelevant retrieval rate subsequently decreases to 2.0%. The combination of the semantics of the sub-questions with the states of unmet evidence improves the routing between structured and unstructured evidence spaces within this evaluation. The robust construct validity of these routing labels is strictly ensured by the double-blind annotation protocol and the high inter-annotator agreement ($\kappa = 0.86$) established in Section 4.1.3.

4.4.2. Ablation Study and Evaluation of the Effectiveness of the Constrained ReAct Agent

The stability and computational efficiency of the investigation module critically affect the operational feasibility within an enterprise security operations center. To evaluate the feedback control mechanisms defined in Section 3.6, we conduct ablation experiments.

To account for the inherent stochasticity of the generation by the LLM and to assure statistical significance, all evaluations of the agent are reported as the mean $\pm$ standard deviation across five repeated independent runs.

The baseline comprises an unconstrained LLM agent that directly processes alert clues and invokes tools without additional process constraints under an identical setting for tool access. The ablation variants remove the state recovery (agent without state recovery) or the action mask (agent without action mask). Table 5 presents the robust statistical results.

Table 5. End-to-end investigation effectiveness and cost analysis of different agent architectures (Mean $\pm$ SD over 5 runs).

Model	End-to-End Triage Acc. (%)	Average Token Consumption
Unconstrained LLM	68.7 $\pm$ 4.5	16,081 $\pm$ 2345
Agent without state recovery	82.3 $\pm$ 2.1	7291 $\pm$ 876
Agent without action mask	84.9 $\pm$ 1.8	6847 $\pm$ 612
Full constrained ReAct agent	91.8 $\pm$ 0.7	4258 $\pm$ 145

The baseline utilizing the unconstrained LLM agent exhibits clear limitations in the evaluated investigation tasks. During the processing of multi-step reasoning chains, the baseline agent frequently selects unsupported actions for tools or repeats reasoning loops, which results in severe instability. The triage accuracy of this baseline fluctuates significantly (68.7% $\pm$ 4.5%), and the average token consumption is excessively high and unpredictable (16,081 $\pm$ 2345).

Among the evaluated variants, the complete constrained ReAct agent achieves the highest absolute triage accuracy (91.8%) while simultaneously exhibiting the lowest cross-run variance ($\pm$ 0.7%). The corresponding token consumption decreases to 4258 $\pm$ 145, confirming that structural constraints effectively suppress runaway generation costs. The ablation variants elucidate the statistical contribution of each mechanism.

The action mask supports bounded execution by preventing invalid actions under the current state. The removal of this mask decreases the mean accuracy to 84.9% and inflates the token consumption to 6847 $\pm$ 612. In the absence of the mask, the agent more frequently attempts premature decisions or irrelevant actions under conditions of incomplete evidence. The restriction of the effective exploration space predictably reduces unnecessary reasoning overhead.

The state recovery mechanism facilitates recovery from invalid or incomplete evidence states. The removal of this module reduces the mean accuracy to 82.3% and increases the variance ($\pm$ 2.1%). The retrieval process may return empty or inconsistent observations; consequently, agents lacking recovery mechanisms may produce unsupported decisions. The rollback of states and the utilization of gap prompts return the agent to the most recent valid checkpoint, thereby reliably limiting the accumulation of irrelevant context.

Within the evaluated tasks, the constrained ReAct design not only improves the aggregate accuracy but also strictly bounds the observed instability and reasoning drift relative to the baseline of the unconstrained LLM agent, thereby ensuring predictable performance for operations facing analysts.

To further elucidate the operational boundaries of the proposed framework, an explicit error analysis is conducted on the residual failed investigation cases of the complete constrained ReAct agent (which constitute an 8.2% failure rate). The failure taxonomy encompasses four primary categories: First, *incorrect SQL and fact retrieval* accounts for approximately 35% of the errors. This occurs when the LLM generates syntactically valid but logically flawed queries that bypass the execution checks. Second, *wrong rule retrieval*

constitutes roughly 25% of the failures, typically triggered by ambiguous sub-questions that mislead the intent-aware routing or the vector search. Third, *hallucinated reasoning* represents 20% of the cases, wherein the LLM misinterprets the retrieved quantitative facts or contradicts the explicit compliance rules despite possessing the correct context. Finally, *premature conclusions under insufficient evidence* account for the remaining 20%. These failures occur when the agent circumvents the gating mechanism for evidence completeness by falsely asserting that the acquired partial evidence satisfies the investigation requirements. This taxonomy demonstrates that, although structural constraints significantly mitigate reasoning drift, the inherent semantic vulnerabilities of the underlying LLM necessitate ongoing refinement.

4.5. Evaluation of System End-to-End Latency and Scalability

We additionally evaluate the end-to-end processing latency within the implementation environment. Although the investigation of insider threats typically does not necessitate real-time blocking decisions, unnecessary bottlenecks during periods of high-volume log processing remain undesirable. Consequently, we decompose the processing time for high-risk alert clues and report the average and P90 latencies across 100 complete investigation workflows. Table 6 presents the results.

Table 6. Evaluation of processing latency distribution across system stages.

System Processing Stage	Average Latency (ms)	P90 Latency (ms)
Anomaly detection stage	15.4	22.1
Evidence routing and retrieval stage	215.8	340.5
Investigation inference and summarization stage	9850.2	12,450.6
End-to-end total pipeline	10,081.4	12,813.2

Table 6 reports the distribution of latency. The stage for anomaly detection requires an average of 15.4 ms per time window, which reflects the lightweight structure and matrix-parallel computation of the heterogeneous dual-branch network. The retrieval of evidence remains below 1 second within this implementation, exhibiting a P90 latency of 340.5 ms; this reflects the utilization of the ClickHouse star schema data mart for multi-dimensional queries and the vector retrieval based on HNSW. The inference for investigation dominates the total latency, accounting for 97.7% of the average end-to-end time owing to the network input/output of the LLM API and autoregressive decoding.

Overall, the system exhibits an average end-to-end latency of approximately 10.1 s (10,081.4 ms) and a P90 latency of 12.8 s. To contextualize these metrics of latency within enterprise-level scalability, we consider the operational alert volume dynamically gated by the detection layer. Based on the performance metrics presented in Section 4.2 (88.4% precision and 96.8% recall on the 4-month test set comprising approximately 1000 users), the HDBN filter yields approximately 750 candidate anomaly windows (combining true and false positives) over 120 days, which averages to 6.25 alerts per day. Given an average processing time of 10.1 s per alert, a single sequential investigation thread utilizing the LLM requires slightly more than one minute of daily computation time.

Extrapolating this workload to a large-scale enterprise comprising 100,000 employees (a 100× scale-up), the expected daily volume of approximately 625 alerts would necessitate roughly 1.75 h of total processing time by the LLM. This volume can be readily absorbed by a single API connection or processed in minutes via low-concurrency asynchronous batching (e.g., five concurrent threads). This mathematical projection demonstrates that, although the autoregressive decoding of the LLM constitutes the computational bottleneck

of the pipeline, the highly selective filtering executed by the HDBN layer ensures that the overall architecture scales feasibly to high-volume enterprise environments without the requirement for massive parallel infrastructure.

5. Discussion

Within the constructed benchmark setting, the collaborative architecture for detection and investigation improves the evaluated metrics for detection, retrieval, and triage. Although the empirical results demonstrate the feasibility of the workflow, several deployment considerations, boundary conditions, and structural limitations necessitate further investigation prior to real-world application.

The primary deployment consideration pertains to data privacy and secure on-premise architectures. The monitoring of enterprise employees inherently demands strict ethical boundaries. The proposed system is designed exclusively as an evidence-grounded tool to support analysts, explicitly excluding automated punitive enforcement or legal attribution. Furthermore, the transmission of confidential alert logs and personnel attributes to external cloud-hosted LLM APIs introduces compliance risks for regulated organizations. Consequently, secure on-premise deployment architectures are imperative. The local operation of advanced open-weight foundation models (e.g., Qwen-family models [54,55]) within the enterprise boundary, combined with parameter-efficient domain adaptation methods such as Low-Rank Adaptation (LoRA) [56], provides a highly scalable alternative to cloud APIs. Furthermore, operational deployments must enforce the redaction of prompts and strict access controls concerning the retrieved evidence.

A secondary consideration involves adversarial robustness and the security of the agent. As established in our threat model statement (Section 3.2), evaluating adversarial evasion and prompt injection is explicitly out of the scope of this study; however, we hypothesize that the proposed dual-track architecture may influence specific vectors of agent manipulation. Unlike generic unstructured RAG pipelines, which are highly susceptible to indirect prompt injection from malicious log payloads, we posit that the separation of structured audit facts from normative text rules could potentially provide a structural defense. Under this assumption, attempts to manipulate the decisions of the agent via poisoned log entries (e.g., the embedding of adversarial instructions within a file name) might be constrained by the gating mechanism for evidence completeness. Because the generated SQL facts must semantically align with the strictly read-only compliance rule tree index, this cross-modal validation is hypothesized to elevate the threshold for adversarial evasion. Future research is required to systematically benchmark these unverified hypotheses under advanced enterprise threat models. Finally, behavior-based baseline modeling encounters inherent challenges regarding cold-start periods and low-frequency attacks. For newly onboarded employees, preliminary analysis indicates that the anomaly scores generated by the HDBN require approximately 14 to 21 days of historical log accumulation to achieve stability. To manage this warm-up phase, the system employs role-based group baseline inheritance, which substitutes missing personalized priors with departmental behavior distributions to control false positives during early deployment. Regarding low-and-slow attacks, the current sliding window stride ($T = 24$ h) may permit exfiltration behaviors spanning several months to blend into normal statistical baselines. Although the semantic branch partially captures extended anomalous sequences, the complete mitigation of low-frequency attacks necessitates a transition from fixed-window aggregation to multi-resolution temporal graph neural networks in future architectural iterations.

6. Conclusions

This paper presented an architecture for the triage of insider threats, spanning from detection to investigation. The proposed method converts suspicious subject time windows into structured clues, organizes audit facts and policy rules via a star schema data mart and a rule tree index, and guides LLM-assisted reasoning through intent-aware retrieval, action masks, gating for evidence completeness, and state recovery. This design ensures that the triage process remains anchored to verifiable evidence, rather than relying exclusively on anomaly scores or unconstrained generation.

During the evaluation on the CERT r4.2 dataset, the HDBN achieved a recall of 96.8% at the selected operating point. The implementation of the star schema data mart increased the logical validity of Text-to-SQL to 93.5%, the intent-aware routing achieved a required path coverage of 97.3%, and the constrained ReAct agent reached an end-to-end triage accuracy of 91.8% while reducing token consumption by 73.5% relative to the baseline of the unconstrained LLM agent. The average end-to-end latency was measured at approximately 10.1 s. These empirical results demonstrate that traceable workflows from detection to investigation are feasible within the constructed benchmark setting. Furthermore, they establish a foundation for future validation involving real enterprise data, private deployment architectures, and the adversarially robust handling of evidence.

Author Contributions: Conceptualization, K.C. and W.Q.; methodology, K.C., D.L. and W.T.; investigation, K.C. and Y.S.; writing—original draft preparation, K.C. and D.L.; writing—review and editing, W.T., Y.S. and W.Q.; supervision, W.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science and Technology Project of the State Grid Corporation of China Central China Branch, “Research on Big Data-Driven Power Grid Network Security Monitoring and Analysis Technology” (Project No.: 52140025000T).

Data Availability Statement: The CERT r4.2 dataset used in this study is publicly available from the CERT Insider Threat Center at Carnegie Mellon University (<https://insights.sei.cmu.edu/library/insider-threat-test-dataset/>, accessed on 30 June 2026), subject to its distribution and access conditions. Derived task construction details and prompt templates can be released when permitted by institutional policy.

Conflicts of Interest: Authors Kai Cheng, Dongkun Li and Weidong Tang were employed by State Grid Corporation of China Central China Branch, Wuhan, China. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Cappelli, D.M.; Moore, A.P.; Trzeciak, R.F. *The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes (Theft, Sabotage, Fraud)*; Addison-Wesley: Boston, MA, USA, 2012.
2. Costa, D.; Albrethsen, M.; Collins, M.; Perl, S.; Silowash, G.; Spooner, D. *An Insider Threat Indicator Ontology*; Technical Report CMU/SEI-2016-TR-007; Software Engineering Institute, Carnegie Mellon University: Pittsburgh, PA, USA, 2016.
3. Verizon Business. *2024 Data Breach Investigations Report*; Technical Report; Verizon Business: New York, NY, USA, 2024. Available online: <https://www.verizon.com/business/resources/reports/2024/2024-dbir-data-breach-investigations-report.pdf> (accessed on 25 May 2026).
4. Verizon Business. *2026 Data Breach Investigations Report*; Technical Report; Verizon Business: New York, NY, USA, 2026. Available online: <https://www.verizon.com/business/resources/T1e0/reports/2026-dbir-data-breach-investigations-report.pdf> (accessed on 25 May 2026).

5. Homoliak, I.; Toffalini, F.; Guarnizo, J.; Elovici, Y.; Ochoa, M. Insight into Insiders and IT: A Survey of Insider Threat Taxonomy, Analysis, Modeling, and Countermeasures. *ACM Comput. Surv.* **2019**, *52*, 30. [\[CrossRef\]](#)
6. Yuan, S.; Wu, X. Deep learning for insider threat detection: Review, challenges and opportunities. *Comput. Secur.* **2021**, *104*, 102221. [\[CrossRef\]](#)
7. Alzaabi, F.R.; Mehmood, A. A review of recent advances, challenges, and opportunities in malicious insider threat detection using machine learning methods. *IEEE Access* **2024**, *12*, 30907–30927. [\[CrossRef\]](#)
8. Villarreal-Vasquez, M.; Modelo-Howard, G.; Dube, S.; Bhargava, B. Hunting for insider threats using LSTM-based anomaly detection. *IEEE Trans. Dependable Secur. Comput.* **2023**, *20*, 451–462. [\[CrossRef\]](#)
9. He, D.; Lv, X.; Xu, X.; Chan, S.; Choo, K.K.R. Double-layer detection of internal threat in enterprise systems based on deep learning. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 4741–4751. [\[CrossRef\]](#)
10. Fei, K.; Zhou, J.; Zhou, Y.; Gu, X.; Fan, H.; Li, B.; Wang, W.; Chen, Y. LaAeb: A comprehensive log-text analysis based approach for insider threat detection. *Comput. Secur.* **2025**, *148*, 104126. [\[CrossRef\]](#)
11. OpenAI. GPT-4 Technical Report. *arXiv* **2023**, arXiv:2303.08774. [\[CrossRef\]](#)
12. Zhang, J.; Bu, H.; Wen, H.; Liu, Y.; Fei, H.; Xi, R.; Li, L.; Yang, Y.; Zhu, H.; Meng, D. When LLMs Meet Cybersecurity: A Systematic Literature Review. *Cybersecurity* **2025**, *8*, 55. [\[CrossRef\]](#)
13. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24824–24837. [\[CrossRef\]](#)
14. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. ReAct: Synergizing Reasoning and Acting in Language Models. In Proceedings of the International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
15. Masterman, T.; Besen, S.; Sawtell, M.; Chao, A. The Landscape of Emerging AI Agent Architectures for Reasoning, Planning, and Tool Calling: A Survey. *arXiv* **2024**, arXiv:2404.11584. [\[CrossRef\]](#)
16. Gong, Y.; Cui, S.; Liu, S.; Jiang, B.; Dong, C.; Lu, Z. Graph-Based Insider Threat Detection: A Survey. *Comput. Netw.* **2024**, *254*, 110757. [\[CrossRef\]](#)
17. Xiao, H.; Zhu, Y.; Zhang, B.; Lu, Z.; Du, D.; Liu, Y. Unveiling shadows: A comprehensive framework for insider threat detection based on statistical and sequential analysis. *Comput. Secur.* **2024**, *138*, 103665. [\[CrossRef\]](#)
18. Duan, S.M.; Yuan, J.T.; Wang, B. Contextual feature representation for image-based insider threat classification. *Comput. Secur.* **2024**, *140*, 103779. [\[CrossRef\]](#)
19. Tian, T.; Zhu, Y.; An, N.; Jiang, B.; Feng, H.; Lu, Z. LSTM-Diff: A Data Generation Method for Imbalanced Insider Threat Detection. In *2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*; IEEE: Piscataway, NJ, USA, 2024; pp. 68–76. [\[CrossRef\]](#)
20. Liu, H.; Liu, M.; Han, L.; Sun, H.; Fu, C. Ripple2Detect: A semantic similarity learning based framework for insider threat multi-step evidence detection. *Comput. Secur.* **2025**, *154*, 104387. [\[CrossRef\]](#)
21. Pang, M.; Ou, W.; Meng, W.; Shen, M.; Yue, Q.; Han, W. MSGSL: A multi-scale group learning model for insider threat detection. *Expert Syst. Appl.* **2026**, *323*, 132379. [\[CrossRef\]](#)
22. Zhang, H.; Huang, J.; Mei, K.; Yao, Y.; Wang, Z.; Zhan, C.; Wang, H.; Zhang, Y. Agent Security Bench (ASB): Formalizing and Benchmarking Attacks and Defenses in LLM-based Agents. In Proceedings of the International Conference on Learning Representations, Singapore, 24–28 April 2025. [\[CrossRef\]](#)
23. Cybersecurity and Infrastructure Security Agency. Insider Threat Analysis [NICE Framework Work Role]. Available online: <https://nics.cisa.gov/tools/nice-framework/work-role/insider-threat-analysis> (accessed on 25 May 2026).
24. Glasser, J.; Lindauer, B. Bridging the gap: A pragmatic approach to generating insider threat data. In *2013 IEEE Security and Privacy Workshops*; IEEE: Piscataway, NJ, USA, 2013; pp. 98–104. [\[CrossRef\]](#)
25. Rashid, T.; Agrafiotis, I.; Nurse, J.R.C. A new take on detecting insider threats: Exploring the use of hidden markov models. In Proceedings of the 8th ACM CCS International Workshop on Managing Insider Security Threats, Vienna, Austria, 28 October 2016; pp. 47–56. [\[CrossRef\]](#)
26. Elmabit, N.; Yang, S.H.; Yang, L.; Zhou, H. Insider threat risk prediction based on Bayesian network. *Comput. Secur.* **2020**, *96*, 101908. [\[CrossRef\]](#)
27. Le, D.C.; Zincir-Heywood, A.N.; Heywood, M.I. Analyzing data granularity levels for insider threat detection using machine learning. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 30–44. [\[CrossRef\]](#)
28. Liu, F.; Wen, Y.; Zhang, D.; Jiang, X.; Xing, X.; Meng, D. Log2vec: A Heterogeneous Graph Embedding Based Approach for Detecting Cyber Threats within Enterprise. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, London, UK, 11–15 November 2019; pp. 1777–1794. [\[CrossRef\]](#)
29. Hong, W.; Li, Z.; Chen, S.; Li, Q.; Niu, W. A Graph Empowered Insider Threat Detection Framework Based on Daily Activities. *ISA Trans.* **2023**, *141*, 84–92. [\[CrossRef\]](#) [\[PubMed\]](#)
30. Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph Attention Networks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.

31. Qi, Y.; Yan, C.; Wang, Z.; Zhang, C.; Liu, S.; Lu, Z.; Jiang, B. ATHITD: Attention-Based Temporal Heterogeneous Graph Neural Network for Insider Threat Detection. *Comput. Secur.* **2025**, *157*, 104587. [\[CrossRef\]](#)
32. Du, M.; Li, F.; Zheng, G.; Srikumar, V. DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*; ACM: New York, NY, USA, 2017; pp. 1285–1298. [\[CrossRef\]](#)
33. Guo, H.; Yuan, S.; Wu, X. LogBERT: Log Anomaly Detection via BERT. In *2021 International Joint Conference on Neural Networks (IJCNN)*; IEEE: Piscataway, NJ, USA, 2021; pp. 1–8. [\[CrossRef\]](#)
34. Tuor, A.; Kaplan, S.; Hutchinson, B.; Nichols, N.; Robinson, S. Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. In *Proceedings of the AAAI Workshop on Artificial Intelligence for Cyber Security*, San Francisco, CA, USA, 4–5 February 2017; pp. 224–231.
35. Huang, W.; Zhu, H.; Li, C.; Lv, Q.; Wang, Y.; Yang, H. ITDBERT: Temporal-semantic representation for insider threat detection. In *2021 IEEE Symposium on Computers and Communications (ISCC)*; IEEE: Piscataway, NJ, USA, 2021; pp. 1–7. [\[CrossRef\]](#)
36. Zhang, F.; Ma, X.; Huang, W. SeqA-ITD: User Behavior Sequence Augmentation for Insider Threat Detection at Multiple Time Granularities. In *2022 International Joint Conference on Neural Networks (IJCNN)*; IEEE: Piscataway, NJ, USA, 2022; pp. 1–7. [\[CrossRef\]](#)
37. Zhu, D.; Huang, X.; Li, N.; Sun, H.; Liu, M.; Liu, J. RAP-Net: A resource access pattern network for insider threat detection. In *2022 International Joint Conference on Neural Networks (IJCNN)*; IEEE: Piscataway, NJ, USA, 2022; pp. 1–8. [\[CrossRef\]](#)
38. Baghalizadeh-Moghadam, N.; Neal, C.; Cuppens, F.; Boulahia-Cuppens, N. NLP and Neural Networks for Insider Threat Detection. In *2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*; IEEE: Piscataway, NJ, USA, 2024; pp. 2010–2018. [\[CrossRef\]](#)
39. Mladenovic, D.; Antonijevic, M.; Jovanovic, L.; Simic, V.; Zivkovic, M.; Bacanin, N.; Zivkovic, T.; Perisic, J. Sentiment classification for insider threat identification using metaheuristic optimized machine learning classifiers. *Sci. Rep.* **2024**, *14*, 25731. [\[CrossRef\]](#) [\[PubMed\]](#)
40. Song, C.; Ma, L.; Zheng, J.; Liao, J.; Kuang, H.; Yang, L. Audit-LLM: Multi-Agent Collaboration for Log-Based Insider Threat Detection. *arXiv* **2024**, arXiv:2408.08902. [\[CrossRef\]](#)
41. Besnaci, S.; Hafidi, M.; Lamia, M. Dealing with extremely Unbalanced Data and Detecting Insider Threats with Deep Neural Networks. In *2023 International Conference on Advances in Electronics, Control and Communication Systems (ICAEECS)*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6. [\[CrossRef\]](#)
42. Lu, N.; Wang, B.; Wang, X.; Xiao, X.; Wang, N.; He, Q.; Wang, W. Black-Box Attacks against Log Anomaly Detection with Adversarial Examples. *Inf. Sci.* **2023**, *639*, 119009. [\[CrossRef\]](#)
43. Liu, Y.; Tao, S.; Meng, W.; Wang, J.; Ma, W.; Chen, Y.; Zhao, Y.; Yang, H.; Jiang, Y. Interpretable Online Log Analysis Using Large Language Models with Prompt Strategies. In *Proceedings of the 32nd IEEE/ACM International Conference on Program Comprehension*, Lisbon, Portugal, 15–16 April 2024; pp. 35–46. [\[CrossRef\]](#)
44. Ferraro, A.; Orlando, G.M.; Russo, D. Generative Agent-Based Modeling with Large Language Models for insider threat detection. *Eng. Appl. Artif. Intell.* **2025**, *157*, 111343. [\[CrossRef\]](#)
45. Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, M.; Wang, H. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv* **2023**, arXiv:2312.10997. [\[CrossRef\]](#)
46. Li, Z.; Zeng, J.; Chen, Y.; Liang, Z. AttacKG: Constructing technique knowledge graph from cyber threat intelligence reports. In *Computer Security—ESORICS 2022*; Springer: Cham, Switzerland, 2022; pp. 589–609. [\[CrossRef\]](#)
47. Zhang, Y.; Du, T.; Ma, Y.; Wang, X.; Xie, Y.; Yang, G.; Lu, Y.; Chang, E.C. AttacKG+: Boosting Attack Graph Construction with Large Language Models. *Comput. Secur.* **2025**, *150*, 104220. [\[CrossRef\]](#)
48. Sun, J.; Xu, C.; Tang, L.; Wang, S.; Lin, C.; Gong, Y.; Ni, L.M.; Shum, H.Y.; Guo, J. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. In *Proceedings of the International Conference on Learning Representations*, Vienna, Austria, 7–11 May 2024.
49. Zhu, D.; Sun, H.; Li, N.; Mi, B.; Huang, X. SPYRAPTOR: A Stream-based Smart Query System for Real-Time Threat Hunting within Enterprise. In *26th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*; IEEE: Piscataway, NJ, USA, 2023; pp. 1055–1062. [\[CrossRef\]](#)
50. Yu, T.; Zhang, R.; Yang, K.; Yasunaga, M.; Wang, D.; Li, Z.; Ma, J.; Li, I.; Yao, Q.; Roman, S.; et al. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 31 October–4 November 2018; pp. 3911–3921. [\[CrossRef\]](#)
51. Pourreza, M.; Rafiei, D. DIN-SQL: Decomposed In-Context Learning of Text-to-SQL with Self-Correction. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 36339–36348. [\[CrossRef\]](#)
52. Cen, J.; Liu, J.; Li, Z.; Wang, J. SQLFixAgent: Towards Semantic-Accurate Text-to-SQL Parsing via Consistency-Enhanced Multi-Agent Collaboration. *Proc. AAAI Conf. Artif. Intell.* **2025**, *39*, 49–57. [\[CrossRef\]](#)

53. Malkov, Y.A.; Yashunin, D.A. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 824–836. [[CrossRef](#)] [[PubMed](#)]
54. Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. Qwen Technical Report. *arXiv* **2023**, arXiv:2309.16609. [[CrossRef](#)]
55. Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. Qwen3 Technical Report. *arXiv* **2025**, arXiv:2505.09388. [[CrossRef](#)]
56. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. In Proceedings of the International Conference on Learning Representations, Virtual Event, 25–29 April 2022.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.