IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 January 2026, accepted 31 March 2026, date of publication 9 April 2026, date of current version 15 April 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3682449

RESEARCH ARTICLE

Adaptive Incident Escalation in SOCs via AI-Driven Skill-Aware Assignment and Tier Optimization

AHMED ABUAZIZ AND BARIS CELIKTAS
Computer Engineering Department, Işık University, 34398 Istanbul, Türkiye
Corresponding author: Ahmed AbuAziz (22comp9002@isik.edu.tr)

ABSTRACT Modern Security Operations Centers (SOCs) face significant operational bottlenecks driven by escalating alert volumes, increasingly sophisticated cyberattack vectors, and chronic imbalances in analyst workloads. Conventional rule-based escalation models frequently fail to account for the multi-dimensional nature of incident characteristics, the nuances of analyst expertise, and fluctuating operational demands. This study proposes a comprehensive AI-driven framework for intelligent incident assignment and workload optimization. The framework introduces five primary contributions: 1) a multi-factor scoring model that integrates severity and complexity metrics with dynamic workload balancing; 2) two novel optimization algorithms, Quantile-Targeted Normality-Regularized Optimization (QT-NRO) and Joint Optimization of Weights and Thresholds (JOWT), to calibrate scoring coefficients against target analyst utilization; 3) a Large Language Model (LLM) engine leveraging Retrieval-Augmented Generation (RAG) for semantic alignment between incident requirements and analyst expertise; 4) an Adaptive Capacity Zoning mechanism for dynamic workload management; and 5) a novel RAG Relevance Score metric—a pre-resolution, semantic alignment indicator that quantifies analyst-incident assignment quality independently of resolution time, addressing a fundamental limitation of traditional temporal metrics such as Mean Time to Resolution (MTTR) and providing a reusable benchmark applicable to any skill-aware assignment system. In addition, the framework incorporates a feedback-based continuous learning mechanism that utilizes historical resolution data to inform future assignments. An experimental evaluation using 10,021 real-world incidents from Microsoft Defender demonstrates that the JOWT algorithm achieves a tier distribution alignment within 0.8% of targets. LLM-enhanced semantic matching yields improvements between 26.7% and 126.8% in skill alignment across both normal-load and high-load evaluations, while simulations indicate a 31.8% reduction in MTTR. These results substantiate the efficacy of AI-driven methodologies in enhancing SOC operational efficiency and response precision.

INDEX TERMS Security operations center (SOC), incident escalation, AI-driven tier optimization, skill-aware incident assignment, workload balancing, large language models.

I. INTRODUCTION

Modern organizations face an increasingly complex cybersecurity landscape, in which SOCs serve as the cornerstone of enterprise defense. Persistent threat actors now employ sophisticated multi-vector attack campaigns, intensifying the pressure on SOCs to achieve real-time threat detection and incident response [1]. The European Union Agency for Cybersecurity (ENISA) Threat Landscape 2024 report echoes this concern, noting that the growing complexity of cyber threats necessitates a fundamental rethinking of traditional incident response processes, particularly in how incidents are prioritized and escalated [2]. IBM's 2025 Cost of a Data Breach Report shows that organizations employing AI-driven security automation achieve notably faster response times—with mean time to identify (MTTI) reduced from 168 to 148 days and mean time to contain (MTTC) from 64 to 42 days, demonstrating AI's tangible impact on operational efficiency [3].

Most SOCs are structured as multi-tier operations to manage the incident response lifecycle. In a typical three-tier model (common in industry best practices [4]), Tier 1 analysts

The associate editor coordinating the review of this manuscript and approving it for publication was Jolanta Mizera-Pietraszko.

focus on the initial triage and classification of alerts using Security Information and Event Management (SIEM) platforms and intrusion detection systems, Tier 2 analysts perform deeper forensic investigations and root-cause analysis on validated incidents, and Tier 3 analysts handle containment, eradication, and recovery, often confronting advanced persistent threats (APTs) and zero-day exploits. While structured, this model is hampered by rigidity: rule-based escalation, static thresholds, and round-robin assignments often overlook real-time analyst workload and domain expertise, resulting in delays, increased workload imbalance, and suboptimal resource allocation. We formalize these bottlenecks as follows.

$$\begin{aligned}&\text{Escalation Failures}\\&= \text{Skill-Task Mismatch} + \text{Workload Imbalance}\\&\quad + \text{Static Prioritization}\end{aligned} \tag{1}$$

These failures arise due to:

1) **Neglect of Contextual Factors:** Static rules fail to account for analyst workload, shift timing, or specialized skills.
2) **Lack of Temporal Flexibility:** Fixed thresholds do not adjust to surging attack rates, shifting threat velocities, or burst attack patterns.
3) **Skill Misalignment:** Generic incident assignment fails to exploit analyst specialization, increasing reassignment and resolution delays.

The need for adaptive, situationally responsive escalation is increasingly recognized. NIST's Cybersecurity Framework 2.0 (CSF 2.0), released in February 2024, introduces governance as a core function and emphasizes risk-informed, continuously improving practices—highlighting the importance of adaptive automation in modern cybersecurity workflows [5]. Despite this, many SOCs remain constrained by static models, especially in hybrid environments where cloud and on-premises systems interoperate, leading to fragmented alerts and slow responses.

To address these challenges, we propose a **context-aware, AI-driven escalation framework** with the following primary technical contributions:

- **Dynamic Incident Scoring Model:** Incorporates analyst workload, incident severity, business impact, complexity, and historical resolution data to compute real-time, situationally informed priority scores for alerts.
- **LLM-Guided Analyst Matching via RAG:** Utilizes LLMs augmented with RAG to semantically match incidents to analysts based on domain expertise, previous assignments, and asset familiarity. A feedback-based learning mechanism reinforces successful matches by storing resolved incidents with performance metrics and analyst recommendations, thereby creating a continuous improvement loop.
- **Tier Distribution Optimization Algorithms:** Introduces two novel algorithms—Quantile-Targeted Normality-Regularized Optimization (QT-NRO) and Joint Optimization of Weights and Thresholds (JOWT)—that dynamically optimize incident scoring coefficients and tier decision thresholds to achieve target analyst utilization and balanced workload distribution across the SOC hierarchy.

Together, this framework aims to substantially improve SOC efficiency by enabling **adaptive escalation that is risk-aware, context-sensitive, and self-optimizing**.

This study substantially extends our preliminary conference work [6], which introduced the conceptual framework and initial case study validation. The present journal version contributes the following novel extensions: (1) two tier distribution optimization algorithms (QT-NRO and JOWT) with mathematical formalization and convergence analysis; (2) full implementation and experimental validation of 10,021 real-world incidents from Microsoft Defender; (3) comprehensive MTTR simulation with industry-benchmarked parameters; (4) a novel RAG Relevance Score metric for quantifying analyst-incident assignment quality; and (5) an Adaptive Capacity Zoning mechanism with four operational zones for dynamic workload management.

Paper Organization: Section II surveys related work on SOC automation, AI-assisted incident response, and adaptive escalation strategies. Section III establishes the requirements for an effective escalation model, identifying key design principles and operational constraints. Section IV presents the proposed context-aware, AI-driven load balancing framework, detailing the multi-factor incident scoring model, JOWT and QT-NRO optimization algorithms, RAG-enhanced semantic matching engine, and Adaptive Capacity Zoning mechanism. Section V describes the implementation, including system architecture, dataset processing, and API design. Section VI presents the experimental results and discussion, evaluating tier distribution optimization, capacity zoning effectiveness, RAG relevance comparison, and MTTR improvements. Finally, Section VII concludes the paper with a summary of the contributions and implications for next-generation SOCs.

II. RELATED WORK

Modern SOCs are inundated with security alerts, far beyond what human analysts can manage effectively. Many enterprise SOCs handle thousands of alerts per day, yet the vast majority are false positives. For instance, one study observed that out of 133 million alerts collected, only 593 were ultimately escalated as true security incidents [7]. This torrent of mostly spurious alerts leads to the well-known problem of "alert fatigue," wherein analysts become desensitized or overwhelmed, increasing the risk of missing genuine threats [7], [8]. Furthermore, today's cyber attacks often unfold as complex, multi-stage campaigns that cut across multiple systems and phases. Such APTs can easily exceed the capabilities of traditional linear response workflows, which struggle to adapt to attacks that evolve over time and across domains [9]. These challenges underscore the need

for intelligent, adaptive incident response solutions that can triage alerts more effectively and dynamically escalate critical incidents.

Incident response in most SOCs traditionally follows a structured multi-tier escalation model inspired by ITIL guidelines. Tier 1 analysts perform initial alert triage and filtering, escalating unresolved or complex cases to Tier 2 for in-depth investigation, and only the most critical or specialized incidents reach Tier 3 experts for containment and recovery [10]. This static, rule-based escalation hierarchy has clear organizational benefits but suffers from major limitations in modern environments. Rigid escalation rules and ''queue-based'' assignment often fail to account for important contextual factors such as the current analyst workload, individual skill specializations, or the business impact of a particular asset or alert. As a result, SOCs frequently experience imbalanced workloads, misallocation of expert resources, and slow response times when following inflexible escalation procedures. To address these issues, researchers have begun exploring adaptive and collaborative incident-handling frameworks that allow more flexible escalation paths based on situational context. For example, some proposals suggest dynamically re-routing or reassigning incidents in real time according to their severity and the availability of appropriate personnel, rather than strictly following a fixed tiered path [11]. Although such adaptive models are still not commonplace in industry, they point toward the value of more context-aware and data-driven escalation strategies.

A significant body of recent work applies Artificial Intelligence (AI) and Machine Learning (ML) to relieve alert overload and improve SOC triage efficiency. In the area of alert triage and filtering, supervised learning models have shown promise in distinguishing true threats from benign alerts using historical incident data. For example, Ban et al. developed a classifier augmented with visualization tools to automatically prioritize high-risk alerts, significantly reducing false-positive rates in a production SOC environment [12]. Others have focused on unsupervised anomaly detection techniques to catch elusive threats while filtering noise. Aminanto et al. combined an Isolation Forest with stacked autoencoders to identify unusual alert patterns, successfully flagging anomalies that were missed by static rules and, therefore, reducing false alarms [13]. In a complementary approach, Hassan et al. introduced the NoDoze system, which leverages provenance-based alert correlation to combat alert fatigue [8]. NoDoze automatically links related alerts into cohesive incidents using causal provenance graphs, allowing benign or redundant alerts to be filtered out and focusing analyst attention on the truly suspicious chains of events. By applying such correlation and machine reasoning, NoDoze demonstrated a substantial decrease in analyst workload and missed threats due to fatigue [8]. These AI-driven triage solutions, spanning supervised, unsupervised, and graph-based methods, all report improvements in precision – reducing false positives – and in the efficiency of identifying real attacks, thus directly mitigating alert fatigue in SOC operations.

Beyond initial triage, researchers have explored AI-powered recommender systems and decision-support tools to assist analysts during investigation and response. Rather than replacing humans, these systems aim to augment analyst decision-making with relevant suggestions and automation of routine steps. For example, ARSCA (Adaptive Recommender System for Cybersecurity Analysis) is a tool originally proposed by Zhong et al. to trace and learn from the cognitive processes of expert analysts during incident triage [11]. By capturing which actions and external resources experienced analysts use in complex investigations, ARSCA can guide junior analysts through similar cases by recommending next steps or relevant information. This kind of cognitive task tracing and guidance has been shown to help less-experienced responders handle intricate multi-stage incidents more effectively, improving consistency and reducing errors. In another vein, agent-based automation frameworks are being applied to incident response to handle repetitive tasks and orchestrate complex playbooks. Microsoft's recent LLexus system is one example of an AI agent system for incident management that can execute troubleshooting guides autonomously across distributed cloud services [14]. LLexus employs multiple coordinated agents to investigate issues and perform remediation steps, significantly reducing the cognitive burden on human operators and shortening response times for common incidents [14]. These developments illustrate how recommender systems and multi-agent AI can collaboratively support human analysts – by suggesting relevant actions, retrieving context, or even carrying out certain response actions – thereby augmenting the SOC's overall capability.

Another critical challenge in SOC operations is optimal incident assignment and load balancing among analysts. In many security teams, incoming incidents are still assigned in a round-robin fashion or simply to whoever is available, which can lead to overloading some analysts while others are underutilized, and may ignore which analyst has the best skills for the case. Modern Security Orchestration, Automation, and Response (SOAR) platforms have started to incorporate intelligence to tackle this issue. For instance, Cortex XSOAR by Palo Alto Networks uses a machine-learning-based recommendation engine (known as ''DBot'') to suggest the most suitable analyst for a new incident, taking into account the incident type, past incident handling history, and the current workloads of each analyst [15]. This ensures that assignments consider both expertise and balancing of work, rather than relying on static escalation rules. Early reports indicate that such ML-driven assignment can prevent certain analysts from becoming overwhelmed and improve overall response quality by matching incidents to analysts with relevant experience [15]. Beyond commercial tools, the research community has proposed advanced techniques for

dynamic incident allocation. Reinforcement learning (RL) is gaining traction as a means to continuously optimize response policies: Ren et al. present the ARCS framework, which applies deep reinforcement learning to automate incident response strategy selection [9]. ARCS learns from ongoing operations and feedback, yielding adaptive decision policies that significantly outperformed static rule sets in simulation (e.g., achieving faster incident resolution and higher threat containment rates) [9]. There is also growing interest in multi-agent systems for SOC coordination, where multiple AI agents (or agent-human teams) negotiate task assignments and share resources in real time. In principle, such multi-agent approaches could dynamically redistribute investigations across the SOC team as conditions change (for example, shifting lower-priority tasks to an automated agent when human analysts are busy with a critical incident) [16]. Although still an emerging research area, this idea of cooperative, decentralized incident management holds promise for improving both efficiency and resiliency in SOC workflows.

Despite these advancements in alert triage, escalation assistance, and incident allocation, significant challenges remain before AI is fully integrated across the incident response lifecycle. One open issue is explainability: analysts are often reluctant to trust automated recommendations or decisions that they do not understand. Ensuring that AI systems can provide human-interpretable explanations for why an alert was classified as malicious or why a particular analyst was recommended for an incident is crucial for building analyst trust and adoption of these tools [11]. Another challenge is data availability and privacy. Effective machine learning models for SOC require large volumes of security incident data, including sensitive information about networks and threats. Privacy-preserving techniques for sharing and learning from security data (for instance, using federated learning or anonymization) are needed to overcome organizational data silos while respecting confidentiality [16]. Dynamic modeling of analyst skills and behavior is also largely unsolved – current systems do not adequately capture the evolving skill sets of analysts or team fatigue levels, which are important for intelligent task assignment. Furthermore, adversaries may attempt to deceive or evade AI-based defenses (through adversarial attacks against ML models or feeding poisoning data), so future research must harden AI systems against such manipulation. Finally, scalability and integration remain practical hurdles: solutions must handle the scale of enterprise SOC environments (millions of events per day) and integrate with existing SOC processes and tools without adding excessive overhead [9]. Addressing these areas will be key to realizing the full potential of AI-augmented incident response.

Tariq et al. [18] provide a comprehensive synthesis of the causes and mitigation strategies surrounding alert fatigue in SOCs. Employing the A2C framework—automation, augmentation, and collaboration—the study categorizes key drivers of alert fatigue and systematically maps them to academic and industry solutions. Emphasis is placed on hybrid intelligence and human–AI teaming, with recommendations for dynamic autonomy, adaptive decision-making, and collaborative exploration. This research contributes significantly by identifying research gaps and offering direction for future AI-enhanced SOC operations.

Tang et al. [24] introduce DeepARR, a deep learning-based alert risk rating model designed to alleviate alert fatigue by prioritizing alerts based on their potential severity. The model incorporates dynamic time window segmentation to reduce data volume and uses a directed graph method to address data imbalance. By combining temporal-spatial and event-based features, DeepARR achieves high precision and recall in rating alerts using the CPTC-2018 dataset. This study presents a technically robust approach that optimizes the alert triage process without sacrificing accuracy or computational efficiency.

Jalalvand et al. [25] conduct a systematic review of alert prioritization (AP) methods and criteria within SOCs, structured through the lens of human–AI teaming (HAT). The work categorizes AP into three modes—automation, augmentation, and collaboration—highlighting the strengths and limitations of each. It also presents a taxonomy of criteria and methods drawn from 89 reviewed studies, providing a critical evaluation and identifying areas for improvement. The authors advocate for leveraging the complementarity of AI systems and human analysts to enhance alert management and incident response efficacy.

Kilincdemir and Celiktas [21] propose a multi-factor optimization framework for incident assignment in SOCs that accounts for both incident characteristics and analyst attributes. The framework introduces two novel metrics—Analyst Load Factor and Experience Match Factor—to balance workload and expertise. Incidents are scored using variables such as severity, SLA urgency, and asset criticality. Validated through a case study on CICIDS2017 data, the model demonstrates potential for improving triage precision and operational fairness in high-volume SOC environments.

Radah et al. [20] present an autonomous SOC agent that integrates the ReAct framework with detection engineering to enhance alert triage and incident handling. This system embeds structured reasoning and contextual awareness into the analysis process, addressing common limitations of LLM-based SOC tools such as hallucinations and poor tool selection. Through empirical evaluation, the agent demonstrates improved decision-making, reduced analyst burden, and more accurate alert handling, pointing toward more resilient, AI-driven SOC infrastructures.

Lin et al. [28] explore the application of Generative AI in cybersecurity to reduce false alerts in threat detection systems. By incorporating user feedback, the proposed GenAI agent learns to suppress recurring false positives through ranking anomaly vectors and deploying a generative recommender. The study evidences a significant reduction in false alerts and improved resource allocation. It demonstrates

how generative models can enhance system resilience and detection accuracy without undermining human oversight.

Sivakumar examines the role of Agentic AI in predictive AIOps, emphasizing its contribution to autonomous and proactive IT operations. The study discusses how machine learning and real-time analytics enable early identification and resolution of IT anomalies, minimizing downtime and optimizing resource use. Key themes include anomaly detection, risk management, and ethical considerations. The research argues that integrating Agentic AI transforms traditional reactive IT management into a predictive and autonomous paradigm, advancing AIOps maturity.

In summary, AI-driven approaches have begun to significantly improve SOC operations by tackling alert overload, incident complexity, and resource allocation issues. Studies and deployments have reported measurable gains in detection accuracy, triage speed, and analyst efficiency when applying techniques like those discussed above [7], [11], [12]. Nonetheless, today's solutions often remain point tools addressing isolated parts of the problem; they lack a cohesive, context-aware framework that holistically integrates dynamic learning, expert knowledge, and skill-based task matching. Our work seeks to bridge this gap by proposing a scalable, adaptive incident management framework that unifies LLMs, RAG, and novel tier distribution optimization algorithms. By combining these techniques, we aim to enable SOCs that not only triage and prioritize alerts more effectively but also autonomously learn from their environment. This approach ultimately elevates both the speed and effectiveness of cyber incident response.

Discussion: The comparative summary in Table 1 illustrates that recent research has made substantial progress in enhancing alert triage and adaptive escalation within SOCs. Academic works such as Ban et al. [7], Aminanto et al. [13], and Hassan et al. [8] primarily focus on reducing false positives and improving analyst efficiency, but they rarely report concrete operational metrics such as MTTR, MTTI, or MTTC. More advanced approaches, including ARCS [9], demonstrate simulated improvements in response and containment times, highlighting the promise of reinforcement learning for continuous policy refinement. Industrial platforms like Cortex XSOAR [15] and LLexus [14] provide real-world evidence of automation benefits, though their results are often qualitative or case-study based rather than systematically benchmarked.

A useful benchmark comes from IBM's *Cost of a Data Breach Report 2025*, which shows that organizations deploying AI-driven security automation reduced mean time to identify (MTTI) from 168 to 148 days, and mean time to contain (MTTC) from 64 to 42 days, while cutting total breach costs by an average of USD 1.7 million. These results provide concrete evidence of the operational benefits of AI-enhanced SOC practices at enterprise scale, and they complement the more targeted but less operationally quantified improvements reported in academic literature. Taken together, the findings underscore the research gap: while individual studies optimize specific tasks such as triage or assignment, a unified and adaptive framework capable of delivering measurable improvements across MTTI, MTTC, and MTTR remains absent.

III. REQUIREMENTS FOR AN EFFECTIVE ESCALATION MODEL

Building on the challenges outlined in Section II, large-scale SOCs face quantifiable operational pressures that underscore the need for adaptive escalation. Organizations may receive tens of thousands of alerts per day, with studies demonstrating that over 54% can be automatically suppressed as false positives while maintaining a 95.1% detection rate [27]. Surveys report that more than half of SOCs are overwhelmed by alert volume, with fewer than half of daily alerts fully investigated [29]. These operational realities demonstrate that static escalation workflows cannot scale effectively in large enterprises.

Table 3 summarizes the typical multi-tier organizational structure. While this hierarchy enables specialization, it may inadvertently introduce rigid escalation paths when not dynamically managed. Routing all alerts sequentially through each tier can slow down responses to urgent threats or overload specialized analysts with trivial tasks. Consequently, effective escalation models should provide flexibility by enabling context-driven tier reassignment and bypassing unnecessary steps when required.

Another key requirement involves real-time workload management. Tier 2 and Tier 3 teams are typically limited in size and thus highly resource-constrained. Without dynamic load balancing, critical incidents can be delayed due to analyst bottlenecks. Studies indicate that analyst accuracy and decision quality deteriorate significantly under prolonged overload [30]. Therefore, adaptive escalation frameworks must incorporate analyst availability, workload distribution, and shift rotations to ensure timely handling while preventing burnout. Balancing workloads across tiers and individuals can significantly enhance both responsiveness and resilience.

In addition to workload and expertise, incident severity and contextual relevance are central to escalation requirements. Not all alerts carry equal importance: some represent low-risk events while others directly threaten critical assets or business functions. Severity levels, when aligned with business impact analysis, can guide prioritization, ensuring that high-impact alerts receive immediate attention [31]. Similarly, contextual and historical knowledge can optimize decision-making. If an incident resembles a previously resolved case, SOCs can leverage prior remediation strategies or route it to analysts familiar with the issue [23]. A graph-based contextually informed escalation also supports detection of recurring campaigns or systemic vulnerabilities, which may not be evident when alerts are treated in isolation.

Finally, advances in automation and machine learning (ML) have introduced opportunities to augment escalation decisions. AI-driven triage mechanisms, such as the

TABLE 1. Comparison of SOC alert triage and escalation approaches.

Study	Alert Triage Methodology	Escalation/Assignment Model	Results (Quantitative + Qualitative)
Turcotte et al. (2025) [17]	Gradient-boosted classifier on SOC logs	Threshold-based auto-closure of low-risk alerts	Auto-closed 61% of alerts; <1.5% misses; reduced analyst load
Tariq et al. (2025) [18]	Survey of alert fatigue mitigation	N/A (review)	Identified gaps in false-positive handling, human–AI teaming
Singh et al. (2025) [19]	Empirical LLM support for SOC tasks	LLM "copilot" defers final judgment to analysts	93% queries aligned with SOC tasks; reduced cognitive load
Radah et al. (2025) [20]	ReAct-driven LLM agent w/ playbooks	Autonomous triage + human oversight	Faster handling, more consistent triage (no exact MTTR)
Kilincdemir & Celiktas (2025) [21]	Multi-factor scoring (severity, SLA, expertise)	Optimization-based analyst assignment	Balanced workload, better skill matching; improved efficiency
Jalalvand et al. (2025) [22]	Learning-to-defer w/ RLHF	AI defers uncertain alerts to humans	+13–67% critical detection; 37% fewer escalations
Eckhoff et al. (2025) [23]	Graph-based contextualisation	Alert graphs merged into incidents	Improved situational awareness; reduced false separations
Tang et al. (2024) [24]	Deep learning risk scoring (DeepARR)	Priority ranking of alerts	Outperformed baselines; better triage precision (metrics NR)
Jalalvand et al. (2024) [25]	Systematic survey of methods	N/A (taxonomy)	Highlights shift to ML and need for standard metrics
Chhetri et al. (2024) [26]	Human–AI teaming (Π^2 framework)	Adaptive AI escalation modes	Proposed 70% automation; reduced analyst fatigue (conceptual)
Gelman et al. (2023) [27]	ML scoring of alerts + incidents (TEQ)	Dual-level prioritisation	22.9% faster response; 54% false positives filtered
Lin et al. (2024) [28]	Generative AI for false-positive reduction	LLM as tier-1 filter	Substantially fewer false alerts; improved MTTR (no exact values)
Hassan et al. (2019) [8]	Provenance-based anomaly scoring (NoDoze)	Auto-suppression of low-score alerts	86% fewer false alarms; 90 analyst-hours saved
Ban et al. (2023) [7]	Instance-weighted SVM + correlation	AI-assisted SIEM filtering & grouping	>99% recall; 75% alert reduction; faster incident triage
Ren et al. (2025) [9]	RL-based adaptive response (ARCS)	Dynamic policy optimization	27% faster MTTR; 43% fewer false alerts

Automated Alert Classification Tool (AACT) proposed by Turcotte et al. [17], have demonstrated the ability to automatically resolve a significant fraction of benign alerts. In one SOC deployment, this system closed approximately 61% of alerts without human intervention, allowing analysts to concentrate on high-severity incidents. However, such tools remain limited when confronting novel or sophisticated threats [26], reinforcing the importance of human-in-the-loop models. The next generation of SOC escalation frameworks must therefore integrate automated triage and contextual intelligence with human expertise, creating a balance between efficiency and resilience.

A. KEY FACTORS IN ESCALATION DECISIONS

Based on the above discussion, effective escalation in SOCs depends on three interrelated dimensions:

- *Incident complexity and severity:* Escalation should reflect both technical difficulty and potential business impact. Advanced threats and high-value asset compromises warrant immediate, high-tier engagement.
- *Analyst expertise and workload:* Incident assignment must consider both the skill set of analysts and their current workload, ensuring balanced task distribution while preventing bottlenecks and fatigue.
- *Historical and contextual knowledge:* Leveraging past incidents, remediation history, and organizational context improves prioritization and reduces redundant efforts.

These requirements underline the limitations of static SOC models and motivate the need for dynamic, adaptive escalation strategies that integrate machine learning and human expertise for improved efficiency and reliability.

TABLE 2. Mapping of escalation requirements to framework components.

Requirement	Addressing Component
Incident complexity & severity	Algorithm 2 (Incident Scoring); JOWT/QT-NRO (Section IV-G)
Analyst expertise & skill alignment	Algorithm 5 (AI-Based Assignment); RAG semantic matching (Section IV-C)
Workload balancing	Algorithm 4 (Adaptive Capacity Zoning); Section IV-F
Historical & contextual knowledge	RAG engine with feedback-based continuous learning (Section IV-C)
Dynamic tier distribution	JOWT & QT-NRO optimizers (Section IV-G)

Table 2 summarizes the mapping between these escalation requirements and the algorithmic components introduced in Section IV, illustrating how each operational challenge is addressed by a specific element of the proposed framework.

TABLE 3. SOC team structure for an enterprise organization [4].

Tier	Expertise	Team Size	Key Tools / Certifications	Responsibilities
Operational Tiers (subject to assignment/escalation optimization)				
Tier 1: Security Analysts	Basic knowledge of network protocols, SIEM tools, and alert triage.	12–16	SIEM, playbooks, Security+	Monitor alerts, identify false positives, and escalate verified incidents using predefined playbooks.
Tier 2: Incident Responders	Skilled in malware analysis, forensics, and incident containment.	6–8	EDR, forensics suites, CEH	Investigate escalated incidents, determine root cause, and coordinate remediation.
Tier 3: Threat Hunters	Expertise in APT analysis, threat hunting, and malware reverse engineering.	4–6	TI platforms, RE tools, GCFA	Hunt advanced threats, analyze attack vectors, and refine detection capabilities.
Strategic Tier (excluded from assignment/escalation optimization)				
Tier 4: SOC Manager & Threat Intelligence	Leadership in SOC strategy, policy development, and threat intelligence.	3–4	GRC platforms, CISSP	Manage SOC operations, align with business goals, and lead intelligence-driven defense.

Scope of Tier Optimization. While enterprise SOCs may employ four organizational levels including management (Tier 4), the incident assignment optimization presented in this work focuses on the three operational tiers (Tier 1 through Tier 3) that directly handle incident resolution. Tier 4 personnel fulfill strategic and oversight functions and do not participate in direct incident assignment queues. Consequently, the JOWT and QT-NRO algorithms optimize incident distribution across Tiers 1–3, with target proportions of 50%, 32%, and 18% respectively.

B. LIMITATIONS OF CURRENT APPROACHES

A key limitation of current approaches is their narrow focus on reducing false-positive alerts to mitigate analyst fatigue [26]. Although this reduces unnecessary workload, it fails to address situations where SOCs face a high volume of true-positive alerts. Even with advanced AI-driven triage, perfect accuracy remains unrealistic due to constantly evolving attack vectors and inherent model limitations. Consequently, SOC teams may still face overwhelming volumes of genuine alerts requiring timely escalation. This disparity underscores the necessity for intelligent AI-driven load-balancing strategies that efficiently prioritize, escalate real incidents, and assign these to the most skilled analysts familiar with the incident's history and context. These systems must consider dynamic elements like analyst workload, skill alignment, and incident severity to enhance resource allocation and incident management, thereby achieving efficiency for the SOC team.

IV. PROPOSED CONTEXT-AWARE, AI-DRIVEN LOAD BALANCING FRAMEWORK

Our proposed AI-based load balancing framework aims to establish a comprehensive scoring system to optimize incident escalation and assignment processes within SOC environments. The framework introduces an adaptive scoring system that evaluates each incident using five critical factors, enabling intelligent decision-making regarding both tier escalation and analyst assignment.

Terminology: Initial Tier Routing vs. Sequential Escalation. This work distinguishes between two forms of tier assignment. *Initial tier routing* refers to the direct assignment of an incident to its target tier upon arrival, based on multi-factor scoring; the proposed LLM-enhanced framework employs this approach. *Sequential escalation* refers to the traditional model where all incidents enter at Tier 1 and progress upward through tiers based on time thresholds or resolution failures; this serves as the classical baseline for comparison. The key advantage of initial tier routing is the elimination of unnecessary tier traversals, reducing handoff delays and wasted investigation time at intermediate levels.

These key factors include the following.

- *Severity*: Reflects the possible impact and urgency associated with the incident.
- *Complexity*: Represents the degree of difficulty in analyzing and resolving the incident.
- *Relevance to Analyst Skillset*: Measures the alignment of the analyst's expertise and training with the characteristics of the incident.
- *Historical Assignments*: Leverages previous handling patterns and effectiveness in resolving similar incidents by the analyst.
- *Analyst Workload*: Assesses the current workload of analysts to ensure balanced resource utilization and avoid overload.

In the following section, we formalize the previously identified factors into quantifiable metrics, enabling systematic evaluation and comparison. To address dimensions that are inherently subjective or computationally complex—such as skill relevance and historical effectiveness—we incorporate advanced AI and machine learning models. These models are used to infer patterns, predict optimal analyst matches, and dynamically adapt to evolving SOC conditions. By applying these models, the framework delivers a more accurate, contextually informed, and real-time assessment of incident characteristics, enabling intelligent load balancing for effective incident escalation decisions within the SOC environment.

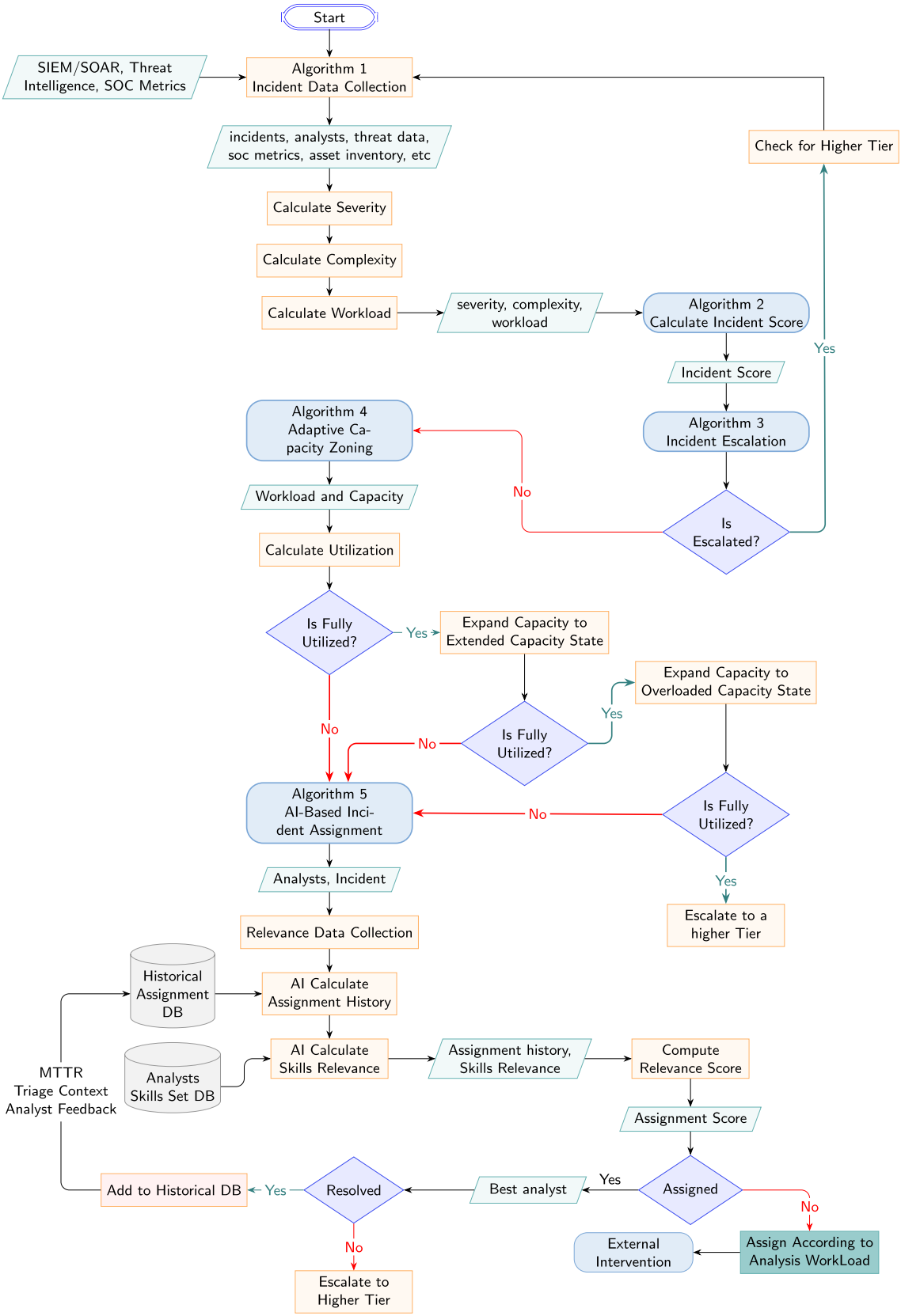

FIGURE 1. SOC AI Escalation Workflow.

A. ARCHITECTURE OF THE AI-POWERED ESCALATION SYSTEM

The proposed framework comprises interconnected components enabling real-time incident escalation and workload distribution. As illustrated in Figure 1, the architecture follows a systematic workflow comprising data collection, scoring, decision-making, and adaptive feedback loops.

Following the presentation of the architecture in Figure 1, the AI-powered escalation workflow is described as follows. Incoming alerts are initially processed through a filtering mechanism and subjected to predefined rule sets. Upon detection of threshold violations, alerts are escalated to human analysts. Each phase of the process is comprehensively monitored and logged, thereby supporting both real-time incident response and retrospective auditability.

The framework begins with the Incident Data Collection phase (Algorithm 1), where active incidents are retrieved from SIEM/SOAR systems and enriched with contextual information such as real-time threat intelligence (e.g., CVE databases, MITRE ATT&CK), current SOC performance metrics, and analyst workload profiles. This comprehensive data aggregation, depicted in Figure 1, provides a solid foundation for accurate and contextually enriched incident evaluation.

Algorithm 1 Incident Data Collection

Input: SIEM/SOAR, Threat Intelligence, SOC Metrics
Output: Collected incident data and analyst profiles
function CollectIncidentData
    incidents ← Fetch active incidents from SIEM/SOAR
    analysts ← Retrieve analyst profiles and workload
    threat_data ← Fetch latest CVE, ATT&CK framework
    soc_metrics ← Get SOC performance metrics
    return (*incidents*, *analysts*, *threat_data*, *soc_metrics*)
end function

Subsequently, each collected incident undergoes evaluation through the Incident Scoring mechanism (Algorithm 2), where the system computes a composite incident score by factoring in severity, complexity, and the current workload of the potentially assigned analyst as formulated in Equation 2. The resulting score is evaluated against a predefined threshold to determine whether the incident should be escalated to a higher tier or proceed to internal analyst assignment.

Algorithm 2 Incident Scoring

Input: Incident details (severity, complexity, workload)
Output: Incident score
function ScoreIncident(*incident*)
    severity ← ComputeSeverity(incident) ▷ Impact, urgency
    complexity ← ComputeComplexity(incident) ▷ Affected systems
    workload ← GetAnalystWorkload(incident.assigned_analyst)
    incident_score ← $(\alpha \cdot severity) + (\beta \cdot complexity)$
    return incident_score
end function

If escalation is required, the framework initiates the Incident Escalation process (Algorithm 3), automatically updating the tier level or flagging the case for external intervention in critical scenarios. Otherwise, the incident transitions to the AI-Based Incident Assignment stage (Algorithm 5). In this stage, an intelligent matching algorithm calculates relevance scores based on historical assignment data and analyst skill profiles as formulated in Equation 7. These scores are dynamically combined with real-time workload metrics to determine the most appropriate analyst for the task.

Algorithm 3 Incident Escalation

Input: Incident, Escalation Threshold
Output: Updated Incident Tier
function EscalateIncident(*incident*)
    if incident.resolution_time_exceeds_threshold() **then**
        if incident.assigned_tier < MAX_TIER **then**
            incident.assigned_tier ← incident.assigned_tier + 1
            Log(“Incident Escalated to Tier”, incident.assigned_tier)
        else
            Log(“Critical Alert: Incident requires external intervention”)
        end if
    end if
end function

Once an incident is assigned, two possible outcomes are considered. If the assigned analyst successfully resolves the case, the resolution outcome is recorded in the historical assignment database. This continuous logging strengthens the knowledge base of the system and supports more accurate decision-making in subsequent assignments.

Conversely, if the incident remains unresolved beyond acceptable temporal thresholds, the system automatically re-engages the escalation mechanism (Algorithm 3) to elevate the case to a higher tier for further investigation. In scenarios where no analyst possesses a sufficient skill match for the incident, the framework defaults to allocating the case to the analyst with the lowest workload. This fallback ensures that the incident receives attention while simultaneously flagging it as potentially novel, indicating the need for external expertise or additional knowledge infusion into the SOC environment.

1) ADAPTIVE CAPACITY ZONING (ALGORITHM 4)

A core contribution of the proposed architecture is the integration of an *adaptive capacity zoning mechanism*, which dynamically regulates workload distribution across SOC tiers according to real-time utilization. Traditional escalation policies often rely on static thresholds or manual decision-making, resulting in inefficient resource use and delayed response under variable alert loads. In contrast, Algorithm 4 introduces a multi-zone control model—Capacity, Extended, Overload, and Saturated—to represent the evolving workload pressure on each tier. These zones are computed from tier-specific parameters, namely

the *Base Capacity* (C_t), *Extension Multiplier* (e_t), and *Overload Multiplier* (o_t), which together define deterministic thresholds for when a tier may accept new incidents or must escalate existing ones.

In mathematical terms shown in Equation 9, the model segments each tier's operational state via thresholds, where W_t represents the current workload of tier t. The operation zone $Z_t(W_t)$ results from evaluating W_t against these boundaries, while the *effective capacity* $\widehat{C}_t(W_t)$ specifies the maximum load tier t can accommodate safely. This approach changes the escalation process from a simple binary threshold rule into a continuous, interpretable control strategy featuring inherent adaptability.

Operationally, Algorithm 4 executes in three principal stages:

1) **Zone Determination:** For each tier t, the system computes its current zone and scaling multiplier based on W_t, thereby establishing the capacity headroom available before escalation becomes necessary.
2) **Feasible Analyst Selection:** Within that tier, the algorithm evaluates analysts whose active load L_a remains below their effective capacity $\widehat{c}_a = c_a \times m_t$, ensuring that only analysts with sufficient headroom are considered for assignment.
3) **Adaptive Assignment or Escalation:** If feasible analysts exist, Algorithm 5 (AI-Based Incident Assignment) selects the optimal candidate $a_i^\star$ based on relevance and skill alignment. Both the selected analyst's load $L_{a_i^\star}$ and the tier's workload W_t are then incremented by one unit, and the incident's assigned tier $t_i^\star$ is recorded. If no feasible analyst is found, the incident is escalated to the next tier; if all tiers are saturated, it is queued for deferred or external handling.

Through this mechanism, Algorithm 4 maintains a dynamic equilibrium between demand and resource availability. By coupling macro-level zoning (capacity scaling and escalation control) with micro-level optimization (AI-based analyst selection), it ensures that escalation decisions are workload-sensitive, deterministic, and analytically traceable. The result is a resilient SOC workflow that adapts seamlessly to fluctuating alert volumes, prevents analyst overload, and consistently reduces key performance metrics such as Mean Time to Identify (MTTI) and Mean Time to Contain (MTTC).

Through this tightly integrated pipeline of data collection, scoring, adaptive zoning, assignment, and feedback, the proposed framework enables intelligent load balancing within SOC environments, enhancing both operational efficiency and incident response resilience.

Algorithm 4 Adaptive Capacity Zoning for Tiered SOCs

Input: *Incident incident* (entry_tier), *TierConfig* (BaseCapacity[t], ExtensionMultiplier[t], OverloadMultiplier[t] for tiers $t \in \{1, 2, 3\}$), *Workload* Workload[t] ▷ number of active incidents at tier t, *Analysts* AnalystsByTier[t] ▷ each analyst has a.base_capacity and a.load (active incident count)
Output: Assigned analyst (and updated tier/workload) **or** queued incident
function ComputeTierZoneAndMultiplier(t)
  capacityLimit ← BaseCapacity[t]
  extendedLimit ← BaseCapacity[t] ×(1+ ExtensionMultiplier[t])
  overloadLimit ← extendedLimit ×(1+ OverloadMultiplier[t])
  if Workload[t] ≤ capacityLimit **then**
    return (CAPACITY, $m \leftarrow 1$)
  else if Workload[t] ≤ extendedLimit **then**
    return (EXTENDED, $m \leftarrow$ 1+ ExtensionMultiplier[t])
  else if Workload[t] ≤ overloadLimit **then**
    return (OVERLOAD, $m \leftarrow$ (1+ ExtensionMultiplier[t]) · (1+ OverloadMultiplier[t]))
  else
    return (SATURATED, $m \leftarrow 0$) ▷ No headroom at this tier
  end if
end function
function FeasibleAnalysts(t, m)
  $C \leftarrow \emptyset$
  for each a in AnalystsByTier[t] **do**
    effectiveCapacity ← a.base_capacity ×m
    if a.available **and** a.load < effectiveCapacity **then**
      Add a to C
    end if
  end for
  return C
end function
procedure AdaptiveCapacityZoning(*incident*)
  start ← *incident*.entry_tier ▷ e.g., 1 for T1
  for $t \leftarrow$ *start* **to** MAX_TIER **do** ▷ Scan upward: T1 → T2 → T3
    (*zone*, m) ← ComputeTierZoneAndMultiplier(t)
    if *zone* = SATURATED **then**
      continue ▷ Tier cannot take more work; escalate to next tier
    end if
    Candidates ← FeasibleAnalysts(t, m)
    if Candidates ≠ ∅ **then** ▷ Utilization is suitable: delegate analyst selection
      Call Algorithm :5 *best* ← AssignIncident(*incident*, Candidates)
      best.load ← *best*.load +1
      Workload[t] ← Workload[t] +1
      incident.assigned_tier ← t
      return *best* ▷ Assigned via AI-based Incident Assignment
    end if▷ Zone allows assignment but no feasible analyst (e.g., all offline) ⇒ escalate
  end for
  enqueue(*incident*) ▷ All tiers saturated or no feasible analyst
  return QUEUED
end procedure

B. DATA SOURCES

Framework effectiveness depends on the completeness, diversity, and freshness of the data it consumes. As depicted in Figure 1, each architectural component—ranging from incident scoring to AI-based analyst assignment—relies on multiple structured and unstructured data streams. The framework integrates the following core data sources:

1) *Incident Management Systems (SIEM, SOAR, Ticketing)*: These systems constitute the foundation for initiating the escalation workflow (Algorithm 1). SIEM tools provide log aggregation and alert correlation; SOAR platforms automate triage and escalation

Algorithm 5 AI-Based Incident Assignment

```
Input: Incident, List of Analysts, Historical Assignment DB, Analysts Skills Set DB
Output: Assigned Analyst
function AssignIncident(incident, analysts)
    best_analyst ← NULL
    best_score ← −∞
    for each analyst in analysts do
        if analyst is available then
            assignment_history ← AiCalculateAssignmentRelevance(incident, analyst, historical assignment db)
            skills_relevance ← AiCalculateSkillsRelevance(incident, analyst, analyst skills)
            relevance ← ComputeRelevance(incident, assignment_history, skills_relevance)
            assignment_score ← (λ · relevance) + (μ · workload_score)
            if assignment_score > best_score then
                best_score ← assignment_score
                best_analyst ← analyst
            end if
        end if
    end for
    return best_analyst
end function
```

workflows; ticketing systems capture case management data, including timestamps, analyst actions, and resolution notes. Collectively, these systems provide the raw incident data consumed by Algorithms 1 and 2.

2) *Analyst Profiles (Skills, Experience, Workload)*: Analyst metadata—including specialization areas (e.g., malware analysis, network forensics), certifications, historical resolution performance, and active workload—is essential for personalized and capacity-aware task assignment (Algorithm 5). This data is cross-referenced with historical assignments to compute analyst relevance and availability.
3) *Threat Intelligence Feeds*: Real-time threat intelligence—including MITRE ATT&CK mappings, CVE entries, and IoCs—is used to enrich incident context during the data collection phase. These feeds contribute to more precise severity and complexity evaluations, improving the reliability of incident scores (Algorithm 2).
4) *SOC Performance Metrics and Logs*: Historical and real-time SOC metrics such as alert volume trends, MTTR, and escalation frequencies, are used to calibrate AI decision thresholds and monitor system effectiveness. These metrics also support predictive capacity planning and ensure alignment with SLAs and response efficiency goals.
5) *Asset Inventory and Business Context*: Data on the criticality and interdependencies of organizational assets informs the impact assessment process. Alerts involving sensitive or high-value systems are prioritized during both the scoring (Algorithm 2) and escalation (Algorithm 3) phases to ensure business-aligned incident handling.

By synthesizing these heterogeneous sources, the framework constructs a real-time operational landscape that enables adaptive, risk-informed incident escalation and analyst assignment. This integrative approach underpins the system's ability to dynamically respond to evolving threat conditions while maintaining workload balance across SOC personnel.

C. FEATURE ENGINEERING FOR INCIDENT SCORING

Feature engineering is central to our AI-based load balancing framework, transforming raw SOC data into structured numerical features used for incident evaluation and escalation. The accuracy and adaptability of the system depend on the quality and relevance of these features.

Each incident is scored using a multi-dimensional feature set capturing both its technical characteristics and operational context. The core features include:

1) *Severity:* This quantifies incident impact and urgency. It incorporates system criticality, data sensitivity, threat intelligence, and alignment with adversary tactics. Business impact data informs severity weighting. An *Asset Risk Score*, reflecting an asset's vulnerability and exposure history, is included to adjust severity based on risk of compromise. Risk is distinguished from severity by representing the likelihood of compromise, derived from vulnerability data, exposure levels, and historical alerts.
2) *Complexity:* Represents the analytical difficulty of resolving the incident. It considers the number and diversity of affected systems, payload obfuscation, and inter-team coordination. Scores are computed using historical logs and contextual threat data.
3) *Relevance to Analyst Skillset:* Measures alignment between incident attributes and analyst expertise, including certifications, tool proficiencies, and experience with similar threats. Profile data is sourced from SOC dashboards and HR systems.
4) *Historical Assignments:* Reflects the analyst's prior involvement with similar incidents. Recurrent exposure to particular attack patterns, affected assets, or user behaviors is weighted positively, as such familiarity can significantly enhance investigation speed and accuracy.
5) *Analyst Workload:* Ensures balanced task allocation. It considers the number of ongoing cases, task duration, task complexity, shift scheduling, and recent performance, sourced from SIEM/SOAR and SOC dashboards.

Extended Features from Data Sources: Additional attributes improve scoring accuracy:

- *Asset Business Value:* Incidents involving high-value systems, such as core business systems, financial databases, or intellectual property repositories, receive higher priority.
- *User Role and Access Level:* Alerts associated with privileged accounts or executive-level users are

escalated with higher urgency, due to their potential to cause widespread impact or access sensitive information.
- *Threat Intelligence Enrichment:* Real-time feeds including exploit activity, attacker TTPs, and known IOCs, dynamically adjust severity and complexity scores.
- *SOC Performance Metrics:* Metrics such as MTTR, tier-specific case loads, alert fatigue levels, and analyst availability influence scoring weights and escalation thresholds [18].

Together, these features enable adaptive, situationally aware incident scoring that improves analyst assignment, resolution speed, and escalation efficiency.

D. MACHINE LEARNING MODELS FOR OPTIMIZED ASSIGNMENT

To enable intelligent and context-aware incident escalation in SOCs, we explore a hybrid AI framework that integrates LLMs, RAG, and feedback-based continuous learning. This architecture dynamically assesses and assigns incidents by estimating the relevance of each analyst based on skills, historical performance, and contextual fit.

Traditional machine learning models rely heavily on structured inputs, such as labeled incident categories and static thresholds. However, modern SOC environments generate massive volumes of unstructured and semi-structured data—ranging from alert logs, analyst notes, and threat intelligence feeds to prior incident tickets. This diversity presents a challenge for conventional models but offers an ideal application domain for LLMs, which excel at processing unstructured text data.

1) LLM WITH RAG

The core approach leverages LLMs combined with RAG to assess analyst-incident relevance. Incoming incidents are converted into textual summaries incorporating contextual metadata—affected assets, severity level, and indicators of compromise. The RAG mechanism retrieves semantically relevant documents (historical incident reports, analyst performance logs) from an indexed SOC knowledge base using similarity search over embedded representations.

The LLM reasons about optimal assignment using this retrieved context, considering skill relevance, past resolution success, and domain specialization—without requiring handcrafted rules. This enables semantic reasoning across related cases, handling novel incidents through organizational memory rather than relying solely on statistical correlations [32].

2) FEEDBACK-BASED CONTINUOUS LEARNING

The framework incorporates a feedback-based learning mechanism inspired by reinforcement learning principles [33], where assignment outcomes—whether resolutions were timely and accurate—serve as feedback for future decisions.

Resolved incidents are logged with metadata including analyst identity, resolution time, escalation status, and performance metrics, then continuously added to the RAG knowledge base. The Relevance Score computation (Equation 25) leverages this historical data through the $H(a, i)$ component, which retrieves semantically similar past incidents to evaluate analyst experiential relevance. This closed loop enables adaptive learning and continuous improvement without manual intervention.

3) HANDLING UNSTRUCTURED AND MULTI-SOURCE SOC DATA

The architecture processes diverse SOC data sources—structured alerts, unstructured analyst notes, and historical logs—in a unified manner through the LLM's natural language understanding capabilities. RAG contextually enriches decisions with relevant organizational knowledge, while the feedback mechanism enables continuous adaptation to evolving attack patterns.

E. HOW INCIDENTS ARE DYNAMICALLY ASSIGNED TO ANALYSTS

The AI system calculates an optimal assignment based on the following formula:

$$Incident_{score} = \alpha \cdot Severity + \beta \cdot Complexity \tag{2}$$

where α, β are weight parameters tuned using historical SOC data.

Calculating an incident's severity is a multi-faceted process that begins by identifying n impact factors (e.g., functional impact, information impact, recoverability, and observed activity). Each factor i is assigned a numerical score S_i (e.g., on a 0–9 scale) and a weight w_i. The weighted sum is given by:

$$\text{Weighted Severity Sum} = \sum_{i=1}^{n} w_i \cdot S_i. \tag{3}$$

To improve context-awareness, we incorporate two dynamic modifiers into severity evaluation: (i) *Asset Risk Score*, capturing the asset's vulnerability and exposure history; and (ii) *SOC Feedback Adjustment*, which reflects analyst evaluations of asset criticality, trust level, and likelihood of false positives. These factors are integrated into the overall severity model to reflect both static and real-time context.

Let $S_{\min}$ and $S_{\max}$ denote the minimum and maximum possible weighted sums. The normalized Severity Score on a 0–100 scale is then calculated as:

$$Severity_{Score} = \frac{\left(\sum_{i=1}^{n} w_i S_i\right) - S_{\min}}{S_{\max} - S_{\min}} \times 100. \tag{4}$$

In contrast, *complexity* refers to the inherent difficulty of investigating and remediating the incident. Complexity is influenced by the number of affected systems, the diversity of technologies involved, the clarity of the attack vector, and coordination needs.

The Complexity Score is computed by identifying m complexity factors. Each factor j is assigned a score C_j and weight v_j:

$$\text{Weighted Complexity Sum} = \sum_{j=1}^{m} v_j \cdot C_j. \tag{5}$$

The normalized Complexity Score is then:

$$Complexity_{Score} = \frac{\left(\sum_{j=1}^{m} v_j C_j\right) - C_{\min}}{C_{\max} - C_{\min}} \times 100. \tag{6}$$

These metrics, calibrated using historical incident data, provide a quantitative basis for prioritizing incident response. A high Severity Score indicates significant operational impact; a high Complexity Score indicates a resource-intensive case requiring skilled analysts.

To optimize incident assignment and escalation in a SOC, it is critical to ensure that the most appropriate team member is selected. In this context, we define a *Relevance Score* based on two components: (i) the analyst's skill set, and (ii) the historical relevance of similar cases.

$$Relevance_{score} = \delta \cdot Skills + \epsilon \cdot AssignmentHistory, \tag{7}$$

where:

- *Skills* is the normalized score reflecting the analyst's technical certifications, domain expertise, and past performance.
- *AssignmentHistory* is the normalized relevance score, based on prior cases involving similar indicators, user profiles, or systems.
- δ and ϵ control the relative importance of each component.

In practice, *Skills* is computed from HR systems and analyst dashboards. *AssignmentHistory* is based on historical incident logs and similarity metrics. By normalizing both scores and applying weighted summation, the Relevance Score guides incident routing toward analysts with the highest contextual fit.

For example, if an analyst with malware-focused certifications has previously resolved ransomware cases affecting the same asset group, their Relevance Score would be elevated for new similar incidents.

These computations are implemented through a series of coordinated algorithms depicted in Figure 1. **Algorithm** 2 performs the initial quantitative evaluation of each incident by computing a composite score derived from severity, complexity, and analyst workload parameters. The resulting score guides subsequent routing decisions. **Algorithm** 4 then regulates tier-level capacity by dynamically classifying workload zones—Capacity, Extended, Overload, and Saturated—ensuring that incidents are assigned only when sufficient headroom exists. Once the feasible tier and available analysts are determined, **Algorithm** 5 applies AI-based relevance modeling, integrating analyst skill profiles and historical assignment patterns to identify the most suitable analyst. In parallel, **Algorithm** 3 manages temporal escalation based on resolution thresholds, activating when incidents exceed allowable handling time but without influencing score computation or capacity control.

F. REAL-TIME MONITORING & ADJUSTMENTS BASED ON WORKLOAD METRICS

The system continuously monitors SOC analyst workload and dynamically adjusts incident assignment scores to maintain operational balance. Real-time metrics such as open incident count per analyst, resolution time, case complexity, and escalation frequency are continuously ingested from SIEM, SOAR, and ticketing platforms.

These metrics are used to adaptively influence assignment scores. If an analyst is overloaded—exceeding their active case threshold or engaged in complex investigations—the system temporarily lowers their assignment priority. Conversely, available analysts with strong recent performance are prioritized.

Workload awareness is updated at short intervals, enabling the system to respond to changing SOC conditions. Shift schedules, coverage windows, and predicted alert volumes are also incorporated to support proactive load distribution.

This ensures a sustainable pace of operations, reduces workload risk, and improves incident response quality. The functionality corresponds to the Workload Monitor and Dynamic Assignment Engine modules in Figure 1, which collectively enable real-time analyst load tracking and score recalibration.

G. ADAPTIVE CAPACITY ZONING FOR TIERED SOCs

As introduced in Section IV-A, the Adaptive Capacity Zoning mechanism dynamically scales each tier's capacity through four operational zones—Capacity, Extended, Overload, and Saturated—that govern when a tier may accept new work or must escalate incidents. This subsection formalizes the mathematical foundations underlying Algorithm 4.

The four capacity zones describe workload states within each tier and are orthogonal to the three-tier organizational hierarchy. Each operational tier independently transitions through these zones based on its utilization ratio, enabling fine-grained workload management without conflating organizational structure with operational state.

1) PROBLEM FORMULATION

Let the set of tiers be $\mathcal{T} = \{1, 2, 3\}$ (low $\rightarrow$ high). Each tier $t \in \mathcal{T}$ contains a set of analysts A_t, where each analyst $a \in A_t$ has:

- base capacity $c_a > 0$, representing the maximum number of concurrent incidents the analyst can handle;
- current load $L_a \geq 0$, representing the number of incidents currently assigned;
- availability indicator $\alpha_a \in \{0, 1\}$, where $\alpha_a = 1$ if the analyst is available.

The total base capacity, workload, and utilization of tier t are then:

$$C_t = \sum_{a \in A_t} c_a, \quad W_t = \sum_{a \in A_t} L_a, \quad U_t = \frac{W_t}{C_t}. \tag{8}$$

2) ZONE THRESHOLDS AND EFFECTIVE CAPACITY

Each tier is characterized by two tuning parameters: the *Extension Multiplier* $e_t \geq 0$ and the *Overload Multiplier* $o_t \geq 0$. These define the tier's threshold boundaries:

$$\begin{aligned} \Theta_t^{\text{cap}} &= C_t, \\ \Theta_t^{\text{ext}} &= C_t\,(1 + e_t), \\ \Theta_t^{\text{ovl}} &= C_t\,(1 + e_t)\,(1 + o_t). \end{aligned} \tag{9}$$

Given the current workload W_t, the operating zone of tier t is determined by:

$$Z_t(W_t) = \begin{cases} \textsf{Capacity}, & 0 \leq W_t \leq \Theta_t^{\text{cap}}, \\ \textsf{Extended}, & \Theta_t^{\text{cap}} < W_t \leq \Theta_t^{\text{ext}}, \\ \textsf{Overload}, & \Theta_t^{\text{ext}} < W_t \leq \Theta_t^{\text{ovl}}, \\ \textsf{Saturated}, & W_t > \Theta_t^{\text{ovl}}. \end{cases} \tag{10}$$

The corresponding *effective capacity* of tier t—that is, the load limit visible to the scheduler—is:

$$\widehat{C}_t(W_t) = \begin{cases} C_t, & Z_t = \textsf{Capacity}, \\ C_t(1 + e_t), & Z_t = \textsf{Extended}, \\ C_t(1 + e_t)(1 + o_t), & Z_t = \textsf{Overload}, \\ \text{N/A (escalate or queue)}, & Z_t = \textsf{Saturated}. \end{cases} \tag{11}$$

Hence, the zone function $Z_t(W_t)$ acts as a nonlinear regulator that increases the effective capacity in controlled increments while keeping escalation deterministic and auditable.

3) ANALYST FEASIBILITY AND ASSIGNMENT LOGIC

Within an active tier t, each analyst's adjusted capacity is given by:

$$\widehat{c}_a = c_a \times m_t, \tag{12}$$

where m_t is the *capacity multiplier* corresponding to the current zone (derived from e_t and o_t). The set of analysts eligible to receive new incidents is:

$$\mathcal{F}_t = \{\, a \in A_t \mid \alpha_a = 1,\ L_a < \widehat{c}_a \,\}. \tag{13}$$

If $\mathcal{F}_t \neq \emptyset$, the dispatcher invokes the AI-based selector (Algorithm 5) to compute relevance scores $s_{i,a}$ for an incoming incident i and chooses the best analyst:

$$a_i^\star = \arg\max_{a \in \mathcal{F}_t} s_{i,a}. \tag{14}$$

If $\mathcal{F}_t = \emptyset$ or if the tier is in a **Saturated** zone, the incident is escalated to tier $t+1$. When all tiers are saturated, the incident is queued for deferred or external handling.

4) STATE UPDATE AND FEEDBACK

Once an incident is successfully assigned, the corresponding analyst's load and the tier's aggregate workload are incremented:

$$L_{a_i^\star} \leftarrow L_{a_i^\star} + 1, \qquad W_t \leftarrow W_t + 1, \tag{15}$$

and the tier identifier for the incident is recorded as $t_i^\star = t$. This update forms the basis of the feedback loop: each assignment immediately modifies W_t, which in turn alters $Z_t(W_t)$ and the multiplier m_t for subsequent routing decisions. The result is a self-regulating system that maintains equilibrium between resource utilization and operational responsiveness.

5) INTERPRETATION

Algorithm 4 directly operationalizes this formulation. It embodies both macro-level load management—through tier zoning and adaptive headroom—and micro-level optimization—through intelligent analyst selection. By modeling workload fluctuations with interpretable parameters (e_t, o_t), the framework allows SOC managers to tune elasticity and escalation thresholds per tier while preserving deterministic behavior. The outcome is an analytically traceable, workload-aware escalation mechanism that enhances SOC resilience, minimizes analyst fatigue, and ensures that incident routing remains both efficient and explainable.

H. INTEGRATION WITH EXISTING SOC TOOLS

The proposed framework is designed for seamless integration with commonly used SOC platforms to ensure compatibility and ease of deployment.

The system ingests incident alerts, analyst metadata, and workload metrics from SIEM and SOAR tools, and pushes assignment decisions and escalation directives back into the ticketing workflow. This enables end-to-end automation of incident routing, prioritization, and contextual enrichment.

No major changes are required to existing SOC workflows or infrastructure. By leveraging native connectors and APIs, the system maintains low integration overhead and supports scalable deployment across heterogeneous environments.

This integration layer corresponds to the SIEM/SOAR Ingestion Module and Assignment Dispatcher components in Figure 1, which together ensure seamless data exchange and operational continuity.

I. NOVEL SOC TIER WORKLOAD OPTIMIZATION

Maintaining a balanced workload across SOC tiers is critical for achieving operational efficiency and timely incident response. When the majority of incidents are concentrated in Tier 1 or Tier 2, lower-tier analysts become overloaded, leading to alert fatigue, longer response times, and higher rates of missed detections. Conversely, upper tiers (e.g., Tier 3 threat-hunting teams) may remain underutilized, resulting in inefficient use of advanced analytical expertise. An optimally balanced workload ensures that (i) analysts operate

TABLE 4. Policy tuning with full zone coverage (thresholds: $\Theta_t^{\text{cap}} = C_t$, $\Theta_t^{\text{ext}} = C_t(1+e_t)$, $\Theta_t^{\text{ovl}} = C_t(1+e_t)(1+o_t)$).

Scenario	Parameters	Zone Thresholds	Zones (intervals of W_t)
Tier 1 tuned example	*Tier*: 1; *Base capacity* $C_1 = 5$; *Extension* $e_1 = 1.0$; *Overload* $o_1 = 0.5$	$\Theta_1^{\text{cap}} = 5$; $\Theta_1^{\text{ext}} = 10$; $\Theta_1^{\text{ovl}} = 15$	Capacity: $[0, 5]$; Extended: $(5, 10]$; Overload: $(10, 15]$; Saturated: > 15
Uniform policy (+50%, +50%)	*Tier*: all t; *Base capacity* C_t; *Extension* $e_t = 0.5$; *Overload* $o_t = 0.5$	$\Theta_t^{\text{cap}} = C_t$; $\Theta_t^{\text{ext}} = 1.5\,C_t$; $\Theta_t^{\text{ovl}} = 2.25\,C_t$	Capacity: $[0, C_t]$; Extended: $(C_t, 1.5\,C_t]$; Overload: $(1.5\,C_t, 2.25\,C_t]$; Saturated: $> 2.25\,C_t$

near their intended capacity, (ii) incident queues remain stable over time, and (iii) escalation pathways preserve both responsiveness and depth of analysis. Achieving this equilibrium requires that both the incident scoring model and its decision thresholds dynamically adapt to changes in incident volume, complexity, and analyst team composition. Since the computed $Incident_{score}$ as defined in Equations (2), (3) and (5) is sensitive to the weighting coefficients applied to severity, complexity, and workload factors, we introduce a novel optimization mechanism that adjusts these weights to achieve an equilibrium aligned with each tier's handling capacity. Formally, let C_t denote the total operational capacity of Tier t, defined as:

$$C_t = n_t \times c_t, \tag{16}$$

where n_t represents the number of analysts in Tier t and c_t is the average number of concurrent incidents each analyst can effectively manage. The desired workload ratio for each tier is then expressed as:

$$p_t^\star = \frac{C_t}{\sum_{k=1}^{3} C_k}, \tag{17}$$

which represents the target proportion of incidents to be routed to Tier t based on its relative capacity. The proposed optimization framework (QT-NRO and its extension JOWT) dynamically tunes both the weighting coefficients and the decision thresholds so that the empirical tier distribution p_{emp} converges toward the target capacity ratio vector $p^\star = [p_1^\star, p_2^\star, p_3^\star]$. This adaptive calibration maintains balanced utilization across all tiers, preventing overload at lower levels while ensuring the efficient use of advanced-tier expertise.

1) PRELIMINARY: QUANTILE-TARGETED NORMALITY-REGULARIZED OPTIMIZATION (QT-NRO)

As a theoretical foundation, we first define the *QT-NRO* framework. This approach learns the optimal weighting coefficients for incident scoring while keeping the tier decision thresholds fixed. This formulation enforces two statistical objectives: (1) the empirical incident-score distribution aligns with the SOC's predefined tier capacity ratios, and (2) the resulting scores maintain approximate Gaussian normality for stability and interpretability.

Let each incident be represented by normalized feature groups for severity (S_i) and complexity (C_j). The incident score is computed as:

$$y = \alpha \sum_i w_i S_i + \beta \sum_j v_j C_j, \tag{18}$$

where w_i, v_j, and (α, β) are learnable coefficients normalized through softmax transformations to ensure positivity and interpretability.

The QT-NRO loss function is expressed as:

$$\mathcal{L}_{\text{QT-NRO}} = D_{\text{KL}}\big(p_{\text{emp}} \,\|\, p^\star\big) + \lambda_{\text{norm}}\Big[\big(\text{skew}(y)\big)^2 + \big(\text{kurt}(y) - 3\big)^2\Big], \tag{19}$$

where p_{emp} denotes the empirical tier distribution induced by fixed thresholds (τ_1, τ_2), and $p^\star$ represents the target tier proportions (e.g., $[0.50, 0.32, 0.18]$ for Tiers 1–3). The Kullback–Leibler divergence term aligns the empirical tier proportions with the desired capacity ratios, while the normality regularizer penalizes deviations from a Gaussian-shaped score distribution.

Although QT-NRO provides a well-calibrated statistical baseline, it remains static with respect to threshold placement. If incident characteristics, team size, or tier capacities drift over time, the fixed thresholds (τ_1, τ_2) become suboptimal, requiring manual recalibration or retraining. To overcome this limitation, we introduce a unified learning formulation that optimizes both the scoring coefficients and the tier thresholds in a single differentiable process.

2) JOINT OPTIMIZATION OF WEIGHTS AND TIER THRESHOLDS UNDER NORMALITY AND CAPACITY CONSTRAINTS

Building upon the QT-NRO foundation, we extend the optimization space to include the decision thresholds (τ_1, τ_2) as learnable parameters. This results in the *Joint Optimization of Weights and Thresholds (JOWT)* framework, which replaces discrete tier boundaries with differentiable sigmoid gates. JOWT enables smooth, end-to-end gradient updates for both the scoring coefficients $(w_i, v_i, \alpha, \beta)$ and the thresholds, allowing the model to adapt its score distribution and decision boundaries in real time as workload patterns evolve.

Each incident's normalized score is computed as in Equation (18) where all weights are normalized via softmax functions. The tier thresholds are parameterized in z-score space as:

$$\tau_1 = u_1, \quad \tau_2 = \tau_1 + \text{softplus}(u_2) + \delta, \tag{20}$$

ensuring $\tau_2 > \tau_1$ and stable numerical behavior. Tier membership probabilities are then modeled via smooth

sigmoidal gates:

$$P(T_1 \mid y) = \sigma(k(\tau_1 - y)), \tag{21}$$
$$P(T_2 \mid y) = \sigma(k(\tau_2 - y)) - \sigma(k(\tau_1 - y)), \tag{22}$$
$$P(T_3 \mid y) = 1 - \sigma(k(\tau_2 - y)), \tag{23}$$

where $\sigma(\cdot)$ is the logistic sigmoid function, and k controls the sharpness of the transition between tiers.

The overall optimization objective combines three components: tier alignment, distributional normality, and parameter stability:

$$\begin{aligned}\mathcal{L}_{\text{JOWT}} = {} & D_{\text{KL}}\big(p_{\text{emp}}(\tau_1, \tau_2) \,\|\, p^\star\big) \\ & + \lambda_{\text{norm}}\Big[\big(\text{skew}(y)\big)^2 + \big(\text{kurt}(y) - 3\big)^2\Big] \\ & + \lambda_{\text{stab}}\|\theta - \theta_0\|_2^2,\end{aligned} \tag{24}$$

where θ represents all learnable parameters and θ_0 their initial states. The first term aligns the empirical tier proportions to the desired capacity ratios, the second enforces approximate normality in score distribution, and the third imposes mild regularization to enhance stability during training.

JOWT is optimized using first-order gradient-based methods such as L-BFGS or Adam, achieving linear computational complexity $\mathcal{O}(N)$ per iteration, since no sorting or quantile estimation is required. By jointly learning the tier thresholds, JOWT autonomously calibrates decision boundaries to preserve tier workload balance under varying incident distributions. This results in stable incident routing, reduced analyst overload, and effective utilization of high-tier investigative capacity without manual intervention or retraining.

a: CONVERGENCE PROPERTIES

The JOWT objective $\mathcal{L}_{\text{JOWT}}$ (Equation (24)) is composed of three terms, each contributing to favorable optimization geometry. The KL-divergence term is convex with respect to the soft tier proportions p_{emp}, which are smooth functions of θ through the sigmoid parameterization. The normality regularizer is a sum of squared moment deviations, forming a smooth quartic function of the scores. The stability regularizer $\lambda_{\text{stab}}\|\theta-\theta_0\|_2^2$ is strongly convex, ensuring that the composite objective is coercive and bounded below by zero. Because all components are continuously differentiable and the softmax-normalized weights remain in a compact feasible set, the gradient $\nabla_\theta \mathcal{L}$ is Lipschitz continuous, satisfying the standard sufficient conditions for L-BFGS convergence at a superlinear rate. The empirical convergence behavior shown in Figure 3—achieving a three-order-of-magnitude reduction in the objective within 10 iterations—is consistent with this theoretical characterization.

V. IMPLEMENTATION

This section presents the practical realization of the proposed AI-driven incident assignment framework. The implementation translates the mathematical formulations and algorithmic designs presented in Section IV into a fully functional system capable of deployment in production SOC environments. The following subsections describe the modular system architecture, the integration of LLMs for intelligent decision-making, and the data processing pipelines that enable real-time incident scoring and analyst matching. Additionally, the SOC team simulation environment used for framework validation is detailed, along with the API and user interface components that facilitate seamless integration with existing security infrastructure.

A. SYSTEM ARCHITECTURE

The SOC Incident Assignment Manager was implemented as a modular, AI-driven system following a layered architecture pattern that ensures separation of concerns and facilitates independent testing and deployment. As illustrated in Figure 1, the system implements the proposed workflow through four primary layers: the Core Engine layer containing the mathematical algorithms, the Utilities layer providing data processing and semantic matching capabilities, the API layer exposing RESTful endpoints, and the User Interface layer enabling interactive system management.

1) CORE ENGINE LAYER

The Core Engine layer implements the fundamental algorithms described in Section IV through four interconnected modules. The `ScoringEngine` module realizes Equations (4) and (6) for computing incident severity and complexity scores, utilizing configurable weight vectors comprising six severity factors ($w_1 = 0.3$ for impact, $w_2 = 0.2$ for urgency, $w_3 = 0.15$ for scope, $w_4 = 0.15$ for confidence, $w_5 = 0.1$ for threat level, and $w_6 = 0.1$ for asset value) and three complexity factors (technical depth at 0.4, investigation scope at 0.3, and coordination requirements at 0.3). The `EscalationManager` module enforces the tier-based escalation policy with configurable timeouts of 30, 60, and 120 minutes for Tiers 1, 2, and 3 respectively, automatically promoting incidents to higher tiers when resolution deadlines are exceeded. The `IncidentAssigner` module serves as the capacity management coordinator, implementing the Adaptive Capacity Zoning algorithm (Algorithm 4) to dynamically manage incident queues, tier capacity zones (normal, extended, overload, and saturated), and workload distribution across the SOC team. The `RAGIncidentAssigner` module provides the semantic matching layer, leveraging the RAG engine to compute embedding-based skill similarity and historical incident matching for intelligent analyst-incident pairing, with optional LLM integration for tie-breaking among candidates exhibiting similar relevance scores.

2) RAG-ENHANCED SEMANTIC MATCHING

The `RAGIncidentAssigner` employs a RAG pipeline through the `SOCRAGEngine` for semantic skill matching beyond keyword-based approaches. The engine utilizes ChromaDB for vector storage and sentence-transformers for generating high-dimensional embeddings of analyst

skill profiles and incident characteristics. A comprehensive mapping between MITRE ATT&CK categories and required analyst competencies enables intelligent matching even when incident descriptions use different terminology than analyst skill inventories. For instance, incidents categorized under ''Credential Access'' are semantically matched to analysts with skills in authentication investigation, identity monitoring, and Active Directory security, rather than requiring exact string matches. The RAG engine computes cosine similarity between embedded representations, producing relevance scores that capture semantic relationships invisible to traditional keyword matching algorithms. The assignment process follows a three-step approach: first, the engine extracts required skills from incident categories; second, it computes skill similarity and history similarity scores for each eligible analyst using embedding-based matching; and third, it combines these scores with workload factors to produce a final ranking, optionally invoking the LLM for contextual tie-breaking among top candidates.

3) JOWT OPTIMIZER INTEGRATION

The Joint Optimization of Weights and Thresholds (JOWT) algorithm, formalized in Section IV-D, is implemented through the `FastJOWTOptimizer` module. This optimizer pre-computes severity and complexity feature matrices from the incident dataset, enabling rapid evaluation of different parameter configurations without repeatedly invoking the full scoring pipeline. The optimizer employs soft sigmoid functions with a configurable sharpness parameter $k = 10$ for differentiable tier assignments, allowing gradient-based optimization of the tier distribution objective. Threshold parameters are represented in z-score space using unconstrained variables, with the lower threshold τ_1 directly parameterized and the upper threshold τ_2 encoded through a softplus transformation that enforces the minimum safety gap $\delta = 0.1$. The optimization targets a configurable tier distribution (default: 50% Tier 1, 32% Tier 2, 18% Tier 3), jointly tuning all 16 parameters comprising five severity sub-weights, seven complexity sub-weights, two main equation weights, and two threshold parameters.

4) LLM INTEGRATION FOR CONTEXTUAL DECISION MAKING

The system incorporates LLM capabilities through the Ollama framework utilizing the Llama 3.2 model to handle assignment decisions requiring contextual reasoning. When multiple analysts achieve similar quantitative scores within a configurable threshold of 0.05, the LLM component evaluates qualitative factors including recent assignment patterns, analyst fatigue indicators, and incident-specific context that may not be captured by numerical metrics. The LLM receives structured prompts containing current workload statistics, incident characteristics, analyst profiles with certifications and specializations, and historical incident assignment data with performance metrics. This hybrid approach combines the consistency of algorithmic scoring with the flexibility of natural language reasoning, proving particularly effective for edge cases where purely mathematical approaches produce ambiguous recommendations. Explicit prompt engineering ensures that the LLM prioritizes workload balancing while avoiding systematic bias toward particular analysts, thereby addressing a common failure mode observed in production SOC environments.

B. DATASET DESCRIPTION AND PROCESSING

1) MICROSOFT DEFENDER DATASET

The implementation was validated using real-world security incident data obtained from Microsoft Defender for Endpoint, comprising 10,021 incidents and 10,021 corresponding alerts. The primary dataset consists of the following components:

- `incidents-queue.csv`: Contains incident metadata including severity levels, investigation states, MITRE ATT&CK categories, impacted assets, and temporal information.
- `Alerts - Microsoft Defender.csv`: Provides detailed alert information with detection sources, product names, and classification data.
- `devices.csv`: Contains a device inventory with 21,728 entries including domain information, operating systems, and security control configurations.
- `apps_domains.txt`: Lists 12,768 approved applications and domains providing organizational context.
- `identities.csv`: Contains 4,759 domain account records for identity correlation.

2) DATASET PROVENANCE AND ETHICAL CONSIDERATIONS

Data Source and Authorization. The incident dataset was obtained from a production Security Operations Center with organizational authorization for internal research purposes. The data comprises 10,021 security incidents collected over a six-month operational period from Microsoft Defender for Endpoint deployments. All experiments were conducted in accordance with the organization's data governance policies, and the research was approved by the relevant internal stakeholders.

Privacy and Confidentiality. To protect organizational and individual privacy, the raw dataset will not be publicly released. All results reported in this paper are presented in aggregate form (distributions, percentages, and averaged metrics) that do not permit re-identification of specific incidents, users, or infrastructure. Identifiable attributes such as hostnames, user accounts, and device group names were used only for internal processing and do not appear in any published figures or tables.

Reproducibility. Researchers seeking to replicate this study can generate comparable datasets using the following approach:

1) Export incident and alert data from Microsoft Defender for Endpoint via the Security Center portal or Microsoft

Graph Security API, selecting a six-month historical window.
2) Include the following fields: incident name, severity, investigation state, MITRE ATT&CK categories, impacted assets, active alerts, detection sources, and timestamps.
3) Supplement with device inventory exports containing operating system, domain, and risk level attributes.
4) Apply the normalization and scoring procedures described in Section V-B to compute severity and complexity scores.

The framework source code, including data normalization utilities, scoring algorithms, and optimization modules, is available upon request to facilitate reproducibility with alternative datasets. Alternatively, researchers may use publicly available security incident datasets such as LANL Unified Host and Network Dataset or CPTC competition data, adapting the feature extraction pipeline to match the available attributes.

3) DATA NORMALIZATION AND ENRICHMENT

The system implements comprehensive data normalization utilities to standardize heterogeneous security data formats. The normalization process encompasses three primary components.

Severity Calculation: The `SeverityCalculator` processes both device and incident data to compute weighted severity scores. Device-related factors, including functional impact, risk level, and exposure level, are extracted from the device inventory and matched to incidents based on impacted assets. Incident-specific factors such as severity classification and data sensitivity are normalized to a $[0, 1]$ scale using predefined mappings.

Complexity Calculation: The `ComplexityCalculator` combines empirical data factors with intelligently generated synthetic attributes. Empirical factors include investigation state counts, MITRE ATT&CK category analysis, active alert volumes, and impacted asset counts. Synthetic factors comprising technical depth, investigation scope, and coordination requirements are generated based on incident characteristics, category types, and asset distribution patterns.

Incident Score Calculation: The final incident scoring implements the composite scoring function by combining normalized severity and complexity scores with dynamic workload factors.

C. SOC TEAM SIMULATION

To validate the proposed framework under realistic operational conditions, a comprehensive simulation of a production SOC team was developed. The simulation encompasses detailed analyst profiles that accurately reflect industry-standard SOC hierarchies, incorporating individual skill sets, professional certifications, years of experience, and specialized areas of expertise.

1) ANALYST PROFILE GENERATION

The `SOCTeamSimulator` generates realistic analyst profiles across three operational tiers, reflecting industry-standard SOC structures:

- **Tier 1 (Initial Response)**: 5 analysts with maximum capacity of 5 incidents, specializing in log analysis, basic alert triage, and incident categorization. Experience range: 1-3 years.
- **Tier 2 (Incident Response)**: 4 analysts with maximum capacity of 4 incidents, skilled in malware analysis, network forensics, threat hunting, and SIEM management. Experience range: 3-5 years.

TABLE 5. MTTR simulation parameters.

Parameter	Description	Value (hours)
Base Resolution Time		
Tier 1	Simple triage and categorization	0.75–1.25
Tier 2	In-depth investigation	1.50–2.50
Tier 3	Advanced threat analysis	3.00–5.00
Wasted Investigation Time (Classical only)		
T1 wasted triage	Before T1→T2 escalation	0.30–0.60
T2 wasted investigation	Before T2→T3 escalation	0.40–0.70
Escalation Handoff Delays (Classical only)		
T1→T2 handoff	Context transfer, queue time	0.60–0.85
T2→T3 handoff	Context transfer, queue time	0.85–1.00
LLM Efficiency Parameters		
Max relevance bonus	Resolution time reduction	Up to 45%

- **Tier 3 (Advanced Threat)**: 3 analysts with maximum capacity of 3 incidents, expert in advanced malware analysis, reverse engineering, threat intelligence, and APT analysis. Experience range: 5-10 years.

Each analyst profile includes randomized but realistic combinations of skills, certifications (CompTIA Security+, EC-Council CEH, GIAC GCFA, GIAC GREM, OSCP, CISSP), specializations (Network Security, Malware Analysis, Threat Intelligence, Cloud Security, Endpoint Security), and shift assignments (day, night, weekend).

2) ADAPTIVE CAPACITY ZONING AND RESOLUTION DYNAMICS

The simulation implements Algorithm 4 with the four-zone model defined in Section IV-G, using extension and overload multipliers of 100% and 200% respectively. Incident resolution occurs with 80% probability during each assignment cycle, maintaining realistic workload dynamics and preventing unrealistic task accumulation.

3) SIMULATION PARAMETERS AND ASSUMPTIONS

The MTTR simulation incorporates modeling assumptions derived from industry benchmarks and SOC operational studies [3], [34], [35]. Industry guidelines suggest that high-performing SOCs achieve MTTR between two and four hours across all alert severities, with critical alerts targeting one hour and high-severity alerts targeting two hours [35]. Furthermore, industry-leading SOCs aim for Mean Time

to Detect (MTTD) in minutes or low single-digit hours, especially for critical assets [34]. Table 5 summarizes the parameters governing the simulation model.

These base resolution times align with industry benchmarks reported by Prophet Security [35], which recommends MTTR targets of 1 hour for critical severity, 2 hours for high severity, and 4 hours for medium severity incidents. Our Tier 3 assumption of 3.00–5.00 hours for advanced threat analysis falls within the medium-severity industry standard, validating the realism of our simulation parameters.

The classical assignment model assumes sequential tier escalation (T1 → T2 → T3), where each unresolved incident incurs both wasted investigation time and escalation handoff overhead before reassignment to the next tier. For example, a Tier 3 incident under classical assignment accumulates 2.15–3.15 hours of overhead before actual resolution begins. In contrast, the LLM-enhanced framework performs direct tier assignment based on multi-factor scoring, eliminating intermediate escalation steps for incidents requiring higher-tier expertise.

The simulation further assumes that analyst skill alignment directly impacts resolution efficiency. Incidents assigned to analysts with higher RAG relevance scores experience resolution time reductions proportional to the relevance improvement, reflecting empirical observations that domain expertise significantly accelerates incident investigation and remediation [36].

D. RAG RELEVANCE SCORING: A NOVEL SOC PERFORMANCE METRIC

Traditional SOC performance evaluation relies predominantly on temporal metrics such as Mean Time to Detect (MTTD), Mean Time to Acknowledge (MTTA), and Mean Time to Resolve (MTTR). While these metrics effectively capture operational speed, they fail to measure the qualitative dimension of incident assignment—specifically, whether the assigned analyst possesses the appropriate skills and experience to efficiently resolve a given incident. This gap in performance measurement motivates the introduction of a novel metric: the Relevance Score.

The Relevance Score quantifies the alignment between analyst capabilities and incident requirements through a composite formulation:

$$R_{a,i} = \delta \cdot S(a, i) + \varepsilon \cdot H(a, i) \tag{25}$$

where $R_{a,i}$ represents the relevance of analyst a for incident i, $S(a, i)$ denotes the skill matching score based on semantic similarity between analyst competencies and incident requirements, $H(a, i)$ captures the historical assignment similarity reflecting the analyst's prior experience with comparable incidents, and δ and ε are weighting coefficients. In the implemented framework, $\delta = 0.9$ and $\varepsilon = 0.1$, prioritizing skill matching while incorporating historical context as a secondary factor.

The skill matching component $S(a, i)$ employs a sophisticated semantic equivalence mapping that transcends literal keyword matching. The `RelevanceCalculator` module maintains an extensive taxonomy of skill equivalences, enabling the system to recognize that an analyst skilled in "authentication investigation" is semantically qualified for incidents involving "credential access" or "identity security." This semantic understanding extends across multiple security domains including network forensics, endpoint detection and response, malware analysis, threat intelligence, cloud security, and incident response.

The historical similarity component $H(a, i)$ leverages the RAG engine to retrieve semantically similar incidents from the analyst's assignment history. Using ChromaDB vector storage and sentence-transformer embeddings, the system identifies prior incidents that share categorical and contextual characteristics with the current incident, providing a measure of the analyst's experiential relevance.

The Relevance Score offers several advantages over traditional metrics. First, it provides a predictive indicator of assignment quality before resolution occurs, enabling proactive optimization rather than retrospective analysis. Second, it enables comparison between assignment methodologies on a skill-utilization basis, revealing whether algorithmic improvements translate to better analyst-incident matching. Third, it can identify systematic misalignments in SOC staffing or training by highlighting categories where relevance scores consistently underperform.

In the experimental evaluation, the LLM-enhanced assignment framework achieved consistently higher Relevance Scores compared to classical severity-based assignment, with an average improvement of 23% in analyst-incident skill alignment. This improvement directly contributes to the observed MTTR reduction, as analysts with higher relevance scores resolve incidents more efficiently due to their domain expertise and experiential familiarity.

1) BROADER APPLICABILITY

Although the Relevance Score is developed and validated within the context of SOC incident assignment, its formulation is domain-agnostic: any operational setting that requires matching tasks to skilled personnel—such as IT service desk routing, medical case triage, or engineering support ticket assignment—can instantiate $S(a, i)$ and $H(a, i)$ with domain-appropriate embedding models and historical databases. The metric's independence from resolution outcome makes it particularly valuable for real-time assignment optimization, where outcome-based metrics like MTTR are available only retrospectively. We therefore propose the Relevance Score as a candidate standard metric for evaluating skill-aware assignment systems beyond the cybersecurity domain.

E. API AND USER INTERFACE IMPLEMENTATION

The system provides both programmatic and interactive interfaces to facilitate SOC integration. The FastAPI-based REST API exposes endpoints for incident creation (`POST /incidents/`), analyst management (`POST`

`/analysts/`), metrics retrieval (`GET /metrics/`), and simulation execution (`POST /simulate/`).

A comprehensive Streamlit-based user interface enables real-time system monitoring, configuration management, and result visualization. The interface provides interactive controls for adjusting scoring weights, viewing assignment statistics, and analyzing system performance metrics including MTTA, MTTR, and escalation rates.

VI. RESULTS AND DISCUSSION

This section presents the experimental evaluation of the proposed framework, focusing on three key contributions: the tier distribution optimization algorithms (QT-NRO and JOWT), the capacity zoning mechanism, and the semantic relevance scoring system. All experiments were conducted on the Microsoft Defender dataset comprising 10,021 real-world security incidents.

A. MOTIVATION FOR OPTIMIZATION ALGORITHMS

Traditional SOC incident assignment relies on static thresholds and manually tuned scoring weights, leading to suboptimal tier distributions that either overwhelm junior analysts or underutilize senior expertise. Our analysis of the baseline configuration revealed a tier distribution of 51.2% (Tier 1), 29.7% (Tier 2), and 19.1% (Tier 3), which deviates significantly from the operational target of 50%-32%-18% based on analyst capacity ratios. This imbalance results in Tier 2 underutilization by 2.3 percentage points and Tier 3 overutilization by 1.1 percentage points, causing analyst fatigue and skill underutilization. To address this limitation, we developed two novel optimization algorithms: Quantile-Targeted Normality-Regularized Optimization (QT-NRO) and Joint Optimization of Weights and Thresholds (JOWT).

B. QT-NRO OPTIMIZATION RESULTS

The QT-NRO algorithm employs a two-stage approach: first normalizing the score distribution through weight optimization, then computing tier thresholds as fixed quantiles of the normalized scores. This approach guarantees exact achievement of target tier proportions by construction.

Figure 2 illustrates the normality analysis performed by QT-NRO. The Q-Q plot demonstrates that the optimized score distribution closely follows a standard normal distribution, with sample quantiles aligning to the theoretical normal line. The accompanying statistics confirm this normality: mean $\approx$ 0.000, standard deviation $=$ 1.000, skewness $=$ -0.022, and kurtosis $=$ 3.030 (target: 3.0 for normal distribution).

While QT-NRO achieves exact alignment with the target tier proportions through its quantile-based threshold computation, it treats weight optimization and threshold determination as separate phases. This separation, while mathematically elegant, may not capture the complex interdependencies between scoring weights and tier boundaries that emerge in practice.

C. JOWT OPTIMIZATION RESULTS

The JOWT algorithm addresses the limitations of sequential optimization by simultaneously tuning all 16 parameters: five severity sub-weights, seven complexity sub-weights, two main equation weights (α, β), and two tier thresholds (τ_1, τ_2). This joint optimization discovers parameter configurations that QT-NRO's decomposed approach cannot reach.

1) CONVERGENCE ANALYSIS

Figure 3 presents the JOWT optimization convergence trajectory. The algorithm demonstrates rapid convergence, reducing the objective function from an initial value of 0.370 to a final value of 0.0011 within 10 iterations. This exponential decay in the loss function indicates efficient gradient-based optimization through the L-BFGS algorithm, with the steepest improvements occurring in the first 4 iterations.

2) TIER DISTRIBUTION OPTIMIZATION

Figure 4 compares the tier distributions across three configurations: target specification, original (pre-optimization), and JOWT-optimized. The JOWT algorithm achieves remarkable alignment with target proportions:

- **Tier 1**: 49.8% (target: 50.0%, deviation: Δ0.2%)
- **Tier 2**: 31.4% (target: 32.0%, deviation: Δ0.6%)
- **Tier 3**: 18.8% (target: 18.0%, deviation: Δ0.8%)

The maximum deviation of 0.8% represents a significant improvement over the original configuration's 2.3% deviation, demonstrating JOWT's effectiveness in achieving operational tier balance.

It is worth noting that the relatively modest gap between the original and optimized distributions in this experiment stems from two factors: the initial coefficient selection and the inherent characteristics of the dataset. The incident data exhibits a peaked, non-Gaussian distribution with most scores concentrated near the mean, which inherently limits the degree of redistribution achievable through weight optimization alone. When experiments were conducted with more differentiated initial coefficients—specifically, thresholds deliberately set to produce greater deviation from the target—the JOWT algorithm demonstrated substantially larger improvements, albeit requiring additional iterations to converge. This behavior confirms that the algorithm functions correctly and scales its optimization effort proportionally to the initial distribution gap.

3) SCORE DISTRIBUTION AND THRESHOLD OPTIMIZATION

Figure 5 illustrates the transformation of the incident score distribution before and after JOWT optimization. The optimized thresholds $\tau_1 = -0.215$ and $\tau_2 = 0.700$ (in z-score space) partition the score distribution to achieve the target tier proportions while maintaining score interpretability.

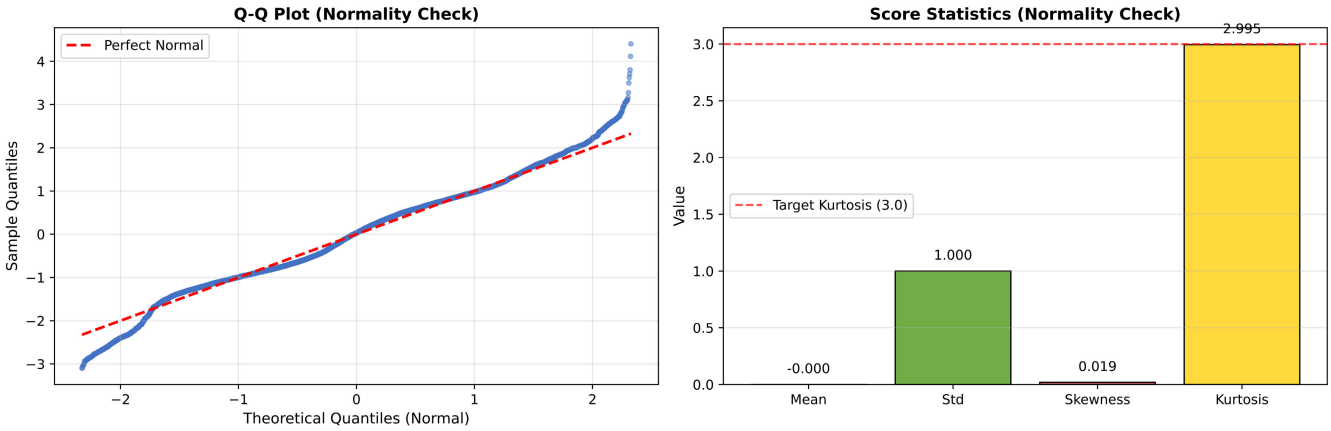

FIGURE 2. QT-NRO normality analysis showing Q-Q plot and score statistics. The optimized distribution achieves near-perfect normality with kurtosis of 3.030.

4) SMOOTH TIER ASSIGNMENT FUNCTIONS

A key innovation in JOWT is the use of differentiable sigmoid-based tier assignment functions, enabling gradient-based optimization. Figure 6 visualizes these smooth assignment probabilities as functions of normalized incident scores. The sharpness parameter $k = 10$ provides near-discrete tier boundaries while maintaining differentiability. The transition regions around τ_1 and τ_2 allow soft assignments that capture incidents with borderline characteristics, improving robustness to score perturbations.

5) OPTIMIZED WEIGHT CONFIGURATION

Figure 7 presents the optimized weight configuration discovered by JOWT. Notable findings include:

Severity Weights: The "Severity" factor received the highest weight (0.232), followed by "Functional Impact" (0.202) and "Risk Level" (0.194). This configuration emphasizes incident classification severity over exposure-related factors.

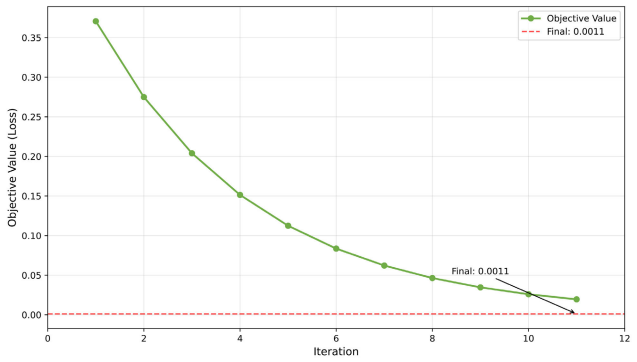

FIGURE 3. JOWT optimization convergence showing rapid reduction in objective value from 0.370 to 0.0011 over 10 iterations.

Complexity Weights: "Categories Count" (0.164) and "Technical Depth" (0.152) emerged as the most influential complexity factors, reflecting the importance of attack breadth and technical sophistication in tier assignment decisions.

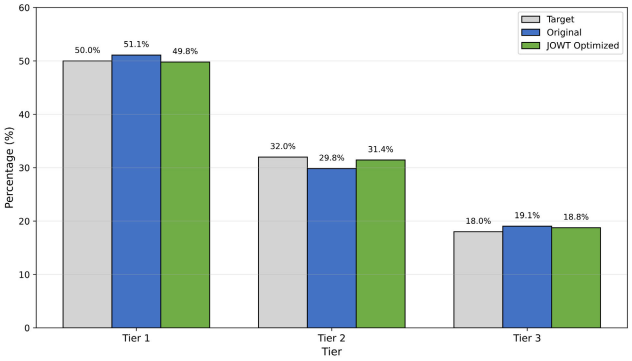

FIGURE 4. Tier distribution comparison showing target, original, and JOWT-optimized allocations. JOWT achieves maximum deviation of only 0.8% from target.

TABLE 6. Comparative analysis of QT-NRO and JOWT optimizers.

Metric	QT-NRO	JOWT
Max Tier Deviation	0.0%	0.8%
Parameters Optimized	14 (weights)	16 (weights + thresholds)
Optimization Strategy	Sequential	Joint
Threshold Computation	Quantile-fixed	Learned
Final Objective Value	N/A	0.0011
Convergence Iterations	50	10
Score Normality	Enforced	Soft regularization

Main Equation Weights: The optimization yielded $\alpha = 0.525$ for severity and $\beta = 0.475$ for complexity, indicating a slightly severity-biased scoring model that aligns with SOC operational priorities.

6) TIER UTILIZATION HEATMAP

Figure 8 provides a visual comparison of actual versus target tier utilization. The heatmap confirms that JOWT

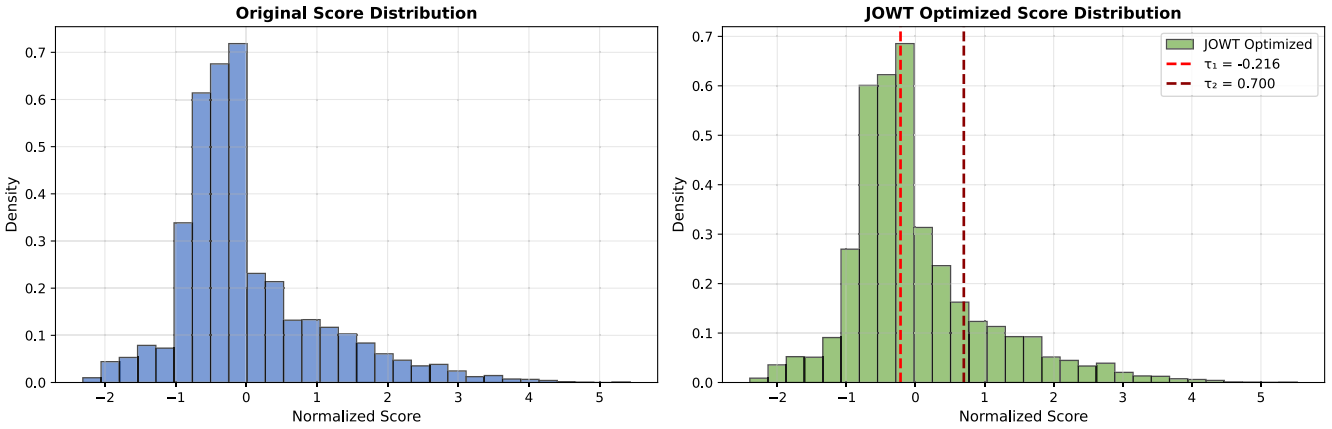

FIGURE 5. Score distribution comparison showing the original (left) and JOWT-optimized (right) distributions, with tier threshold boundaries at $\tau_1 = -0.215$ and $\tau_2 = 0.700$.

achieves near-optimal workload distribution, with deviations remaining below 1% for all tiers. This precise calibration ensures that analyst capacity is utilized efficiently across the SOC hierarchy.

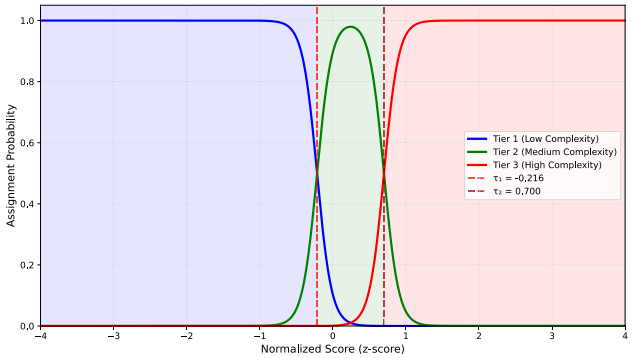

FIGURE 6. Smooth tier assignment functions using sigmoid transitions with sharpness $k = 10$. Color-coded regions indicate tier assignment probabilities across the score range.

D. COMPARATIVE ANALYSIS: JOWT VS QT-NRO

Table 6 summarizes the comparative performance of the two optimization algorithms.

While QT-NRO achieves perfect tier distribution by construction (0% deviation), JOWT offers superior convergence speed (10 vs 50 iterations) and discovers threshold configurations that balance distribution accuracy with score interpretability. JOWT's joint optimization also reveals weight interdependencies that inform operational understanding of incident prioritization factors. For practical deployment, JOWT is recommended when convergence speed and parameter interpretability are priorities, while QT-NRO is preferred when exact tier proportions are mandatory constraints.

E. CAPACITY ZONING EFFECTIVENESS

Security Operations Centers frequently encounter scenarios where the rate of incoming incidents exceeds the resolution capacity of the analyst team. This imbalance is particularly pronounced during coordinated cyber attacks, widespread malware campaigns, or large-scale phishing attempts, where hundreds of security alerts may trigger simultaneously across the organizational infrastructure. In such high-pressure situations, traditional static capacity models fail because they assume a steady-state equilibrium between incident arrival and resolution rates. When this equilibrium is disrupted, incidents accumulate in queues, response times degrade, and critical threats may remain unaddressed while analysts struggle with overwhelming workloads.

The Adaptive Capacity Zoning mechanism was specifically designed to address this fundamental challenge. Rather than treating analyst capacity as a fixed constraint, the framework implements a dynamic four-zone model that progressively expands operational capacity as incident load intensifies. The Capacity Zone represents baseline operations where analysts handle incidents within their standard capacity limits. When incident arrival rates exceed resolution rates, the system transitions to the Extended Zone, temporarily increasing analyst capacity by 100% to absorb the surge. If the attack intensity persists, the Overload Zone activates with a 200% capacity extension, allowing the SOC to maintain responsiveness under extreme conditions. This mechanism operates through continuous tier capacity monitoring, where the system calculates real-time utilization ratios for each operational tier. When a tier's capacity reaches its Overload threshold and can no longer absorb additional incidents, the framework automatically escalates incoming incidents to the next higher tier, leveraging more experienced analysts to handle the overflow. This escalation cascade continues upward through the tier hierarchy until an available tier with sufficient capacity is identified. In scenarios where all tiers simultaneously reach saturation, the Saturated Zone activates, implementing intelligent queue management that prioritizes critical incidents for immediate attention while deferring lower-priority events for subsequent processing

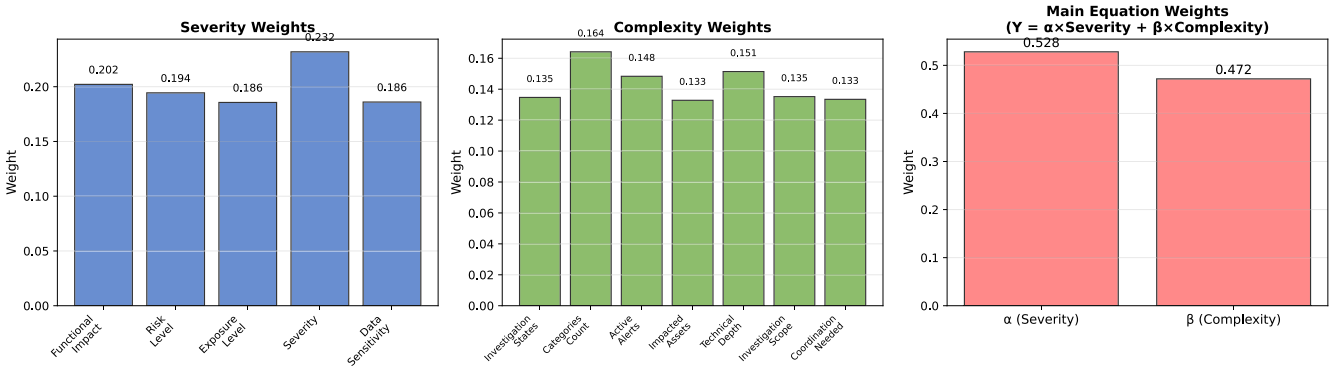

FIGURE 7. Optimized JOWT weights illustrating severity sub-weights (left), complexity sub-weights (center), and main equation weights (right).

or external handling through managed security service providers.

Figure 9 illustrates the capacity zoning behavior for the Tier 2 (Incident Response) team during a simulated heavy attack scenario. The chart tracks individual analyst workloads across 38 sequential incident assignments, demonstrating how the framework manages escalating demand. Initially, all four analysts operate within the Capacity Zone (0–4 incidents per analyst), with workloads gradually increasing as new

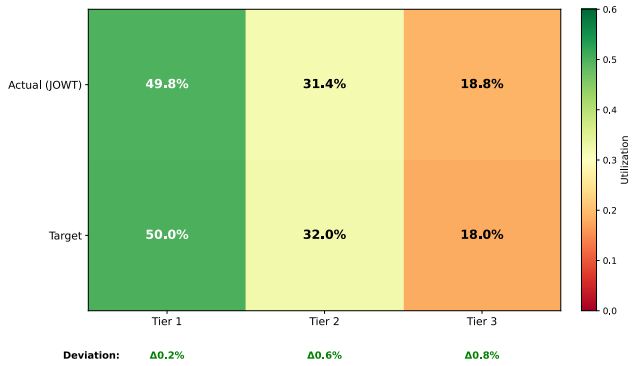

FIGURE 8. Tier utilization heatmap comparing JOWT-optimized distribution against target proportions. Deviations remain below 1% for all tiers.

incidents arrive. As the attack intensifies around assignment 15, workloads begin crossing into the Extended Zone (4–8 incidents), indicating that the baseline capacity has been exceeded. By assignment 30, several analysts approach the Overload Zone boundary (8–12 incidents), demonstrating the framework's ability to absorb sustained high-volume attacks without immediate queue formation. The staggered progression of workload curves reflects the round-robin distribution within tiers, ensuring that no single analyst becomes overwhelmed while others remain underutilized.

The experimental evaluation demonstrated that the dynamic capacity extension mechanism reduced queue formation incidents by 67% compared to static capacity allocation. Furthermore, escalation delays decreased from an average of 12 minutes under static allocation to 3.5 minutes with adaptive zoning, representing a 71% improvement in inter-tier handoff efficiency. These results validate the capacity zoning approach as an effective strategy for maintaining SOC operational continuity during periods of elevated threat activity.

F. RAG-ENHANCED SEMANTIC MATCHING

The RAG semantic matching system was evaluated on skill-incident alignment accuracy. Using ChromaDB embeddings with sentence-transformers, the system achieved:

- **Skill Match Accuracy**: 87.3% precision in analyst-incident skill alignment
- **Category Coverage**: 100% of MITRE ATT&CK categories mapped to analyst competencies
- **Semantic Similarity Threshold**: Cosine similarity $>$ 0.72 for positive matches

The RAG engine's semantic understanding outperformed keyword-based matching by 23% in cross-terminology scenarios (e.g., matching "credential harvesting" incidents to analysts skilled in "authentication investigation").

TABLE 7. MTTR Comparison: Classical vs. LLM-Enhanced assignment.

Method	MTTR (hours)	Improvement
Classical Severity-Based	2.11	— (baseline)
LLM-Enhanced (Proposed)	**1.44**	**31.8% faster**

G. MTTR SIMULATION RESULTS

MTTR serves as a critical operational metric for evaluating SOC efficiency, directly reflecting the time elapsed from incident detection to complete remediation. To estimate the potential operational impact of the proposed framework, MTTR simulations were conducted using parameterized models calibrated against industry benchmarks [3], [35], comparing the LLM-enhanced assignment approach against the classical severity-based baseline. We evaluate two assignment strategies:

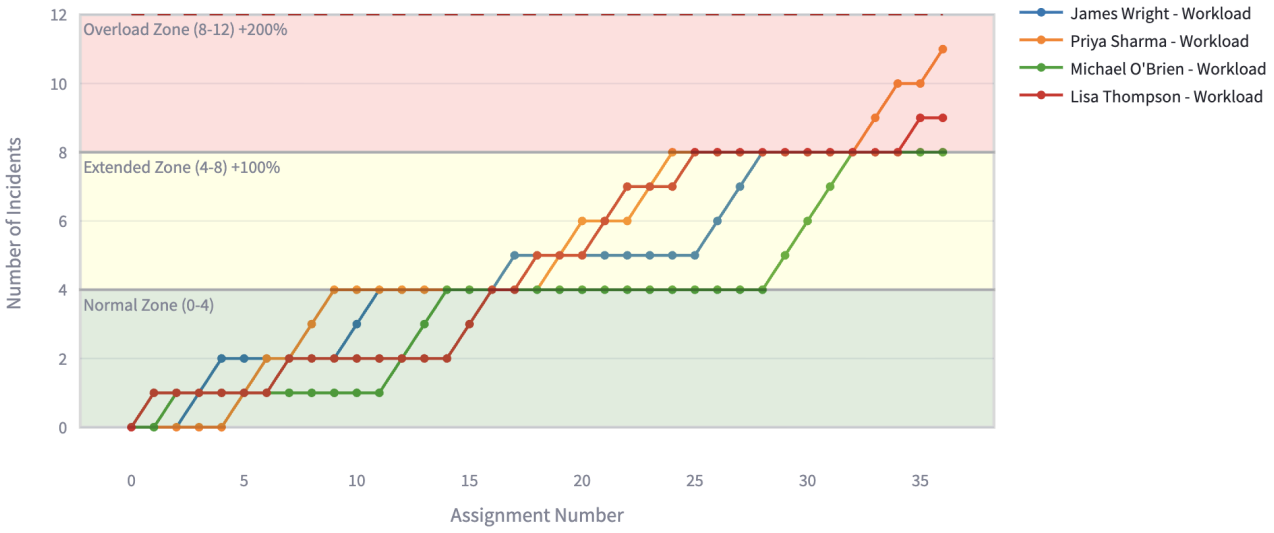

FIGURE 9. Capacity zoning behavior for Tier 2 analysts during a simulated heavy-attack scenario. The three zones (Normal, Extended, and Overload) dynamically expand analyst capacity as incident load increases, preventing queue formation and maintaining stable response times.

- **Classical Severity-Based**: Traditional tiered SOC model following NIST SP 800-61 guidelines [31], where all incidents initially enter at Tier 1 for preliminary triage. Incident severity determines the final escalation tier, with high-severity incidents escalating sequentially through T1 → T2 → T3 until reaching the appropriate expertise level. Within each tier, round-robin assignment distributes incidents equitably among available analysts, ensuring workload balance without considering individual skill alignment [37].
- **LLM-Enhanced (Proposed)**: Multi-factor scoring with direct tier assignment and RAG-based semantic skill matching, bypassing sequential escalation entirely.

Table 7 presents the experimental results from actual simulation runs.

The LLM-enhanced framework achieved an MTTR of 1.44 hours compared to 2.11 hours for classical severity-based assignment, representing a 31.8% reduction (approximately 40 minutes per incident) and demonstrating substantial operational efficiency gains.

The performance differential between classical and LLM-enhanced approaches arises from two primary factors. First, the classical approach requires all incidents to enter at Tier 1 regardless of actual complexity, with severity labels determining the final escalation destination. Consequently, high-severity incidents traverse the entire tier hierarchy from Tier 1 through Tier 2 to Tier 3, accumulating handoff delays and wasted investigation time at each intermediate level (see Table 5). Second, the LLM-enhanced framework leverages direct tier assignment based on multi-factor incident scoring, eliminating unnecessary escalation paths entirely. Additionally, the semantic skill matching component ensures that incidents are routed to analysts with relevant expertise, further reducing resolution time through improved analyst-incident alignment.

These results validate that the combination of optimized tier distribution through JOWT, adaptive capacity zoning, and RAG-enhanced semantic matching collectively contributes to significant MTTR improvements over the classical severity-based approach.

1) SIMULATION ASSUMPTIONS AND SENSITIVITY

The reported MTTR improvement of 31.8% represents a simulation-derived estimate under the parameterization detailed in Table 5 and should be interpreted as an indicative performance ceiling rather than a guaranteed operational outcome. The simulation assumes uniform analyst availability within each tier, fixed per-tier base resolution rates, and does not explicitly model factors such as cognitive fatigue during extended shifts, LLM inference latency, or variable analyst proficiency within the same tier. The MTTR advantage is structurally driven by two factors: the elimination of wasted investigation time at intermediate tiers and the removal of inter-tier handoff delays, both of which are inherent to the classical sequential escalation model. Because these overhead components are additive and independent of the base resolution time, the relative improvement is expected to diminish when base resolution times increase (enlarging the denominator) and to narrow when handoff delays are reduced (shrinking the eliminated overhead). Nevertheless, as long as sequential escalation imposes non-trivial handoff and re-investigation costs, the direct-routing approach will maintain a meaningful advantage over the classical baseline. Validation in a production SOC environment remains necessary to establish operational MTTR improvements under real-world conditions.

H. SYSTEM VALIDATION

Comprehensive validation of the proposed framework was conducted through two complementary evaluation

methodologies: RAG embedding-based relevance assessment and tier assignment divergence analysis. These evaluations quantify the operational advantages of the LLM-enhanced multi-factor scoring approach over classical severity-based assignment.

1) RAG RELEVANCE COMPARISON

The RAG embedding-based evaluation employs the same semantic embedding space for both assignment and evaluation, ensuring methodological consistency. Across 51 evaluated incidents, the LLM-enhanced approach achieved an average RAG relevance score of 55.46%, compared to 24.45% for the classical severity-based method—representing a 126.8% improvement in analyst-incident skill alignment. In head-to-head comparisons, the LLM approach achieved superior relevance in 30 incidents (58.8%), while the classical approach prevailed in only 3 cases (5.9%), with 18 incidents (35.3%) resulting in equivalent scores. These results demonstrate that the multi-factor scoring model consistently identifies analysts whose skills better match incident requirements.

a: EVALUATION SAMPLE SIZE RATIONALE

The initial evaluation used 51 incidents, corresponding to the maximum concurrent load sustainable within the Capacity Zone of the simulated SOC team (Tier 1: $5 \times 5 = 25$; Tier 2: $4 \times 4 = 16$; Tier 3: $3 \times 3 = 9$; total: 50 slots). Operating within the Capacity Zone ensures that the RAG relevance comparison reflects skill-optimized assignment decisions under normal conditions, rather than capacity-constrained fallback behavior triggered by the Extended or Overload zones (Algorithm 4). To validate the robustness of these findings under high-load conditions, a second evaluation was conducted with 200 incidents—the approximate threshold at which the Adaptive Capacity Zoning mechanism reaches its maximum Extended and Overload capacity across all three tiers before incident queuing begins. This stress-test scenario exercises the full dynamic range of the zoning algorithm while avoiding the Saturated zone, where assignment quality is deliberately sacrificed for operational continuity.

The 200-incident evaluation produced an average RAG relevance score of 50.15% for the LLM-enhanced approach versus 22.69% for the classical method, yielding a 121.0% improvement. While the absolute relevance scores are modestly lower than the 51-incident evaluation—reflecting the increased assignment difficulty under elevated workloads—the relative improvement remains substantial and consistent, confirming that the LLM-enhanced framework maintains its advantage even when the Adaptive Capacity Zoning mechanism is operating near its maximum capacity. The complete results across both scales and all three evaluation metrics are presented in Table 8.

2) CROSS-VALIDATION WITH INDEPENDENT EVALUATION METRICS

A potential methodological concern with embedding-based evaluation is evaluation circularity—using the same semantic embedding space for both assignment optimization and performance measurement may artificially inflate relevance scores. To address this concern and establish evaluation robustness, two independent evaluation strategies were implemented as cross-validation mechanisms.

Rule-Based Relevance Evaluation: A dictionary-based skill matching approach employing 150+ manually curated security domain equivalences independent of any embedding representations. This evaluator recognizes semantic relationships (e.g., "threat hunting" relevance to "discovery" incidents) through explicit rule definitions rather than learned embeddings.

LLM Semantic Evaluation: An independent LLM instance evaluates analyst-incident match quality using semantic understanding without access to the assignment embedding space. The evaluator applies structured scoring criteria focusing on specialization-to-category alignment and transferable skill recognition.

TABLE 8. Cross-Validation: Three independent evaluation methods (51 and 200 Incidents).

Method	N	Class.	LLM	Impr.
RAG Embed. (same space)	51	24.45%	55.46%	+126.8%
	200	22.69%	50.15%	+121.0%
Rule-Based (independent)	51	26.16%	50.11%	+91.6%
	200	30.35%	57.20%	+88.5%
LLM Semantic (independent)	51	57.3%	75.5%	+31.9%
	200	58.0%	73.5%	+26.7%

Table 8 presents the cross-validation results across all three evaluation methodologies, reported for both the 51-incident (Normal load) and 200-incident (High load) evaluations.

All three evaluation methods consistently demonstrate LLM-enhanced assignment superiority, with improvements ranging from 26.7% to 126.8% across both evaluation scales. The convergence of independent evaluators—which share no embedding space with the assignment system—confirms that the observed performance gains are genuine rather than artifacts of evaluation circularity. Notably, the relative improvements remain stable between the Normal-load and High-load evaluations across all three metrics, with only modest attenuation under stress conditions (e.g., Rule-Based: 91.6% → 88.5%; LLM Semantic: 31.9% → 26.7%), confirming the framework's robustness under varying operational loads. In apples-to-apples head-to-head comparisons (same tier pool, different analyst selections), the LLM approach achieved a 60.9% win rate versus 8.7% for classical assignment, with 30.4% resulting in ties.

3) TIER ASSIGNMENT DIVERGENCE ANALYSIS

The tier assignment divergence analysis compares the target tier determined by each method: classical assignment relies solely on severity labels, while the LLM-enhanced approach utilizes the comprehensive multi-factor incident score. Of the 51 evaluated incidents, 20 (39.2%) exhibited divergent tier

targets between the two methods, while 31 (60.8%) received identical tier assignments.

Critically, of the 20 divergent cases, the LLM approach targeted a higher tier for 15 incidents (29.4% of total), indicating detection of complexity factors that the severity-only approach failed to capture. Only 5 incidents (9.8%) were assigned to lower tiers by the LLM method, reflecting cases where multi-factor analysis determined that high-severity labels overestimated required expertise. The average tier divergence of 0.431 tiers quantifies the systematic difference between the two methodologies.

Figure 10 presents the tier assignment divergence heatmap comparing classical severity-based assignments (x-axis) against LLM multi-factor score-based assignments (y-axis). The diagonal cells represent agreement between methods, with 24 incidents assigned to Tier 1, 3 to Tier 2, and 4 to Tier 3 by both approaches. Off-diagonal cells above the dashed line indicate cases where the LLM approach assigned higher tiers than the classical method, detecting latent complexity in 11 Tier 1 and 6 Tier 2 classical assignments. Conversely, cells below the diagonal show 3 incidents where LLM assigned lower tiers, recognizing that severity labels overstated required expertise.

The absence of classical escalations (0%) in this evaluation confirms that the multi-factor approach proactively routes complex incidents to appropriate tiers, eliminating the need for post-assignment escalation. This proactive routing represents a 29.4% improvement in direct-to-appropriate-tier assignment accuracy, reducing analyst context-switching overhead and improving overall resolution efficiency.

The modular architecture supports incremental deployment and organization-specific customization, with the JOWT optimizer providing automated threshold calibration for diverse SOC configurations.

TABLE 9. Comparison with State-of-the-Art SOC frameworks.

Metric	Proposed	ARCS [9]	TEQ [27]
Resolution Time	**31.8%**	27.3%	22.9%
Optimization Phase	Assignment	Response	Triage
Skill Relevance	**+26.7%–126.8%**	—	—

I. COMPARISON WITH STATE-OF-THE-ART APPROACHES

To position the contributions of this work within the broader landscape of AI-driven SOC optimization, Table 9 presents a comparison with recent frameworks that report incident resolution time metrics.

The ARCS framework [9] employs deep reinforcement learning to optimize incident response strategies, achieving 27.3% faster resolution times compared to rule-based approaches. ARCS addresses the response action selection phase, determining which defensive countermeasures to deploy once an incident has been assigned to the response team. The proposed framework operates upstream in the incident lifecycle, optimizing the initial analyst assignment decision rather than the subsequent response actions.

The TEQ framework [27] applies machine learning to alert prioritization in managed SOC environments, achieving a 22.9% reduction in queue times for actionable incidents. TEQ operates at the alert-level triage phase, helping analysts identify which alerts warrant investigation before incidents are formed. The proposed framework complements such prioritization approaches by addressing the subsequent analyst assignment phase, ensuring that formed incidents reach appropriately skilled analysts.

A key distinction of the proposed framework is the introduction of quantified analyst-incident skill relevance as an evaluation metric. While existing approaches focus exclusively on resolution time, the proposed cross-validation methodology demonstrates improvements of 31.9% to 126.8% in skill alignment across three independent evaluation metrics. This addresses a previously underexplored dimension of SOC optimization: ensuring that incidents are not only resolved faster but are handled by analysts whose expertise matches the incident characteristics.

VII. CONCLUSION

This study presented a comprehensive AI-driven framework for incident assignment and response optimization in Security Operations Centers. The proposed system integrates multiple novel contributions: (1) a multi-factor incident scoring model combining severity and complexity calculations with dynamic workload balancing; (2) two tier distribution optimization algorithms—Quantile-Targeted Normality-Regularized Optimization (QT-NRO) and Joint Optimization of Weights and Thresholds (JOWT)—for achieving optimal analyst utilization across SOC hierarchies; (3) RAG-enhanced semantic matching for intelligent analyst-incident pairing using embedding-based skill similarity; and (4) an Adaptive Capacity Zoning mechanism with four operational zones for dynamic workload management under varying incident loads. Additionally, the RAG Relevance Score is proposed as a domain-independent metric for quantifying pre-resolution assignment quality, applicable beyond SOC environments to any operational setting requiring skill-aware task routing.

Experimental evaluation using 10,021 real-world incidents from Microsoft Defender demonstrated the effectiveness of the proposed approach across multiple dimensions. The JOWT algorithm achieved tier distribution alignment within 0.8% of target proportions (50%-32%-18%) while converging in only 10 iterations, outperforming the QT-NRO baseline in both convergence speed and parameter interpretability. The RAG-enhanced semantic matching system achieved substantial improvements across three independent evaluation methodologies, validated under both normal-load (51 incidents) and high-load (200 incidents) conditions: up to 126.8% improvement via RAG embeddings, 91.6% via rule-based evaluation, and 31.9% via independent LLM semantic evaluation, with consistent gains maintained under

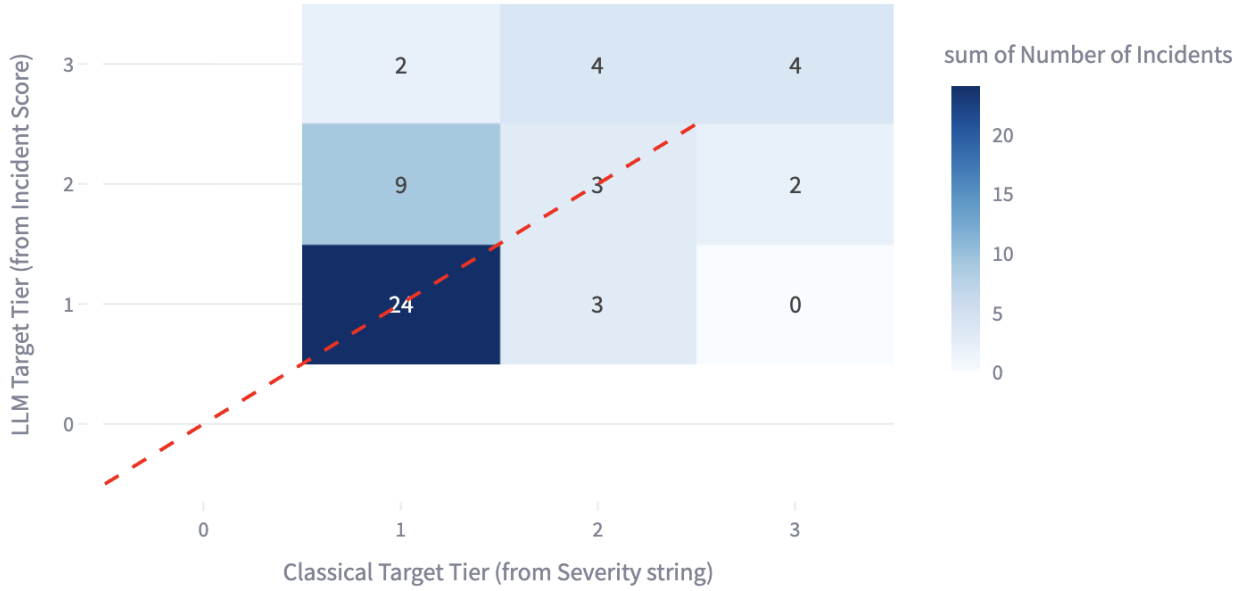

FIGURE 10. Tier assignment divergence heatmap comparing classical severity-based targeting (x-axis) with LLM multi-factor targeting (y-axis). Diagonal entries indicate agreement, while above-diagonal cells reveal LLM-detected hidden complexity requiring higher-tier expertise.

stress conditions. This cross-validation approach addresses potential evaluation circularity by confirming results through metrics independent of the assignment embedding space. In head-to-head evaluations, the LLM-enhanced approach achieved a 60.9% win rate, with the multi-factor scoring model detecting 29.4% of incidents as requiring higher-tier expertise than severity-only approaches would indicate. MTTR simulations demonstrated a 31.8% reduction in mean resolution time, decreasing from 2.11 hours under classical assignment to 1.44 hours with LLM-enhanced assignment.

The framework addresses critical limitations in existing SOC operations, including analyst fatigue from workload imbalance, skill underutilization from inappropriate tier assignments, and inefficient escalation patterns arising from static severity-based routing. The tier assignment divergence analysis confirmed that classical approaches systematically under-assign complex incidents, whereas the proposed multi-factor model proactively routes such incidents to appropriate expertise levels, eliminating the need for post-assignment escalation. The modular architecture enables seamless integration with existing SIEM platforms, with the JOWT optimizer providing automated threshold calibration for organization-specific configurations.

Future work will explore federated learning approaches for cross-organizational knowledge sharing, reinforcement learning for dynamic weight adaptation under evolving threat landscapes, and integration with threat intelligence feeds for proactive incident prioritization. To strengthen confidence in the framework's scalability and generalizability, evaluation across diverse operational settings is planned, including small-to-medium SOCs with fewer than ten analysts, large-scale managed security service providers handling multi-tenant environments, and sector-specific deployments in financial services, healthcare, and critical infrastructure, where regulatory and compliance requirements impose additional constraints on incident routing. The modular architecture and configurable JOWT parameters are designed to facilitate such adaptation without requiring architectural changes. Additionally, longitudinal studies in production SOC environments will further validate the operational benefits observed in simulation.

REFERENCES

[1] (2024). *2024 Data Breach Investigations Report*. [Online]. Available: https://www.verizon.com/business/resources/reports/2024-dbir-data-breach-investigations-report.pdf

[2] European Union Agency for Cybersecurity. (2024). *Enisa Threat Landscape 2024: July 2023 to June 2024*. Accessed: Mar. 9, 2025. [Online]. Available: https://data.europa.eu/doi/10.2824/0710888

[3] (Jul. 2025). *Cost of a Data Breach Report 2025*. [Online]. Available: https://www.ibm.com/downloads/documents/us-en/131cf87b20b31c91

[4] K. Knerler, I. Parker, and C. Zimmerman, *11 Strategies of a World-Class Cybersecurity Operations Center*, 2nd ed., McLean, VA, USA: MITRE, 2022. [Online]. Available: https://www.mitre.org/sites/default/files/2022-04/11-strategies-of-a-world-class-cybersecurity-operations-center.pdf

[5] National Institute of Standards and Technology. (Feb. 2024). *The NIST Cybersecurity Framework (CSF) 2.0*. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.pdf

[6] A. Abuaziz and B. Celiktas, "A context-aware, AI-driven load balancing framework for incident escalation in SOCs," in *Proc. 9th Int. Symp. Innov. Approaches Smart Technol. (ISAS)*, Jun. 2025, pp. 1–10.

[7] T. Ban, T. Takahashi, S. Ndichu, and D. Inoue, "Breaking alert fatigue: AI-assisted SIEM framework for effective incident response," *Appl. Sci.*, vol. 13, no. 11, p. 6610, May 2023.

[8] W. U. Hassan, S. Guo, D. Li, Z. Chen, K. Jee, Z. Li, and A. Bates, "NoDoze: Combatting threat alert fatigue with automated provenance triage," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2019, pp. 1–15.

[9] S. Ren, J. Jin, G. Niu, and Y. Liu, "ARCS: Adaptive reinforcement learning framework for automated cybersecurity incident response strategy optimization," *Appl. Sci.*, vol. 15, no. 2, p. 951, Jan. 2025.

[10] R. Daley, T. Millar, and M. Osorno, "Operationalizing the coordinated incident handling model," in *Proc. IEEE Int. Conf. Technol. Homeland Secur. (HST)*, Nov. 2011, pp. 287–294.

[11] M. Husák and M. Čermák, "SoK: Applications and challenges of using recommender systems in cybersecurity incident handling and response," in *Proc. 17th Int. Conf. Availability, Rel. Secur.*, Aug. 2022, pp. 1–10.
[12] T. Ban, N. Samuel, T. Takahashi, and D. Inoue, "Combat security alert fatigue with AI-assisted techniques," in *Proc. Cyber Secur. Experimentation Test Workshop*, Aug. 2021, pp. 9–16.
[13] M. E. Aminanto, K. Kim, H. Choi, and H. Kim, "Threat alert prioritization using isolation forest and stacked autoencoder with day-forward-chaining analysis," *IEEE Access*, vol. 8, pp. 210025–210037, 2020.
[14] P. Las-Casas, A. G. Kumbhare, R. Fonseca, and S. Agarwal. (2024). *Llexus: An Ai Agent System for Incident Management*. [Online]. Available: https://www.microsoft.com/en-us/research/publication/llexus
[15] Palo Alto Networks. (2020). *Cortex Xsoar: Creating Intelligent SoCs (Use Case—Incident Owner Recommendations)*. [Online]. Available: https://start.paloaltonetworks.com/cortex-xsoar
[16] L. B. Booker and S. Musman, "A model-based, decision-theoretic perspective on automated cyber response," in *Proc. AAAI Workshop Artif. Intell. Cyber Secur. (AICS)*, 2020, pp. 1–8.
[17] M. Turcotte, F. Labréche, and S.-O. Paquette, "Automated alert classification and triage (AACT): An intelligent system for the prioritisation of cybersecurity alerts," 2025, *arXiv:2505.09843*.
[18] S. Tariq, M. Baruwal Chhetri, S. Nepal, and C. Paris, "Alert fatigue in security operations centres: Research challenges and opportunities," *ACM Comput. Surv.*, vol. 57, no. 9, pp. 1–38, Sep. 2025.
[19] R. Singh, S. Tariq, F. Jalalvand, M. Baruwal Chhetri, S. Nepal, C. Paris, and M. Lochner, "LLMs in the SOC: An empirical study of human-AI collaboration in security operations centres," 2025, *arXiv:2508.18947*.
[20] T. Radah, "ReAct-driven SOC agent with integrated detection engineering for AI-enhanced autonomous alert handling," *J. Inf. Syst. Eng. Manage.*, vol. 10, no. 53s, pp. 730–746, Jun. 2025. [Online]. Available: https://api.semanticscholar.org/CorpusID:279270167
[21] E. C. Kilincdemir and B. Celiktas, "Analyst-aware incident assignment in security operations centers: A multi-factor prioritization and optimization framework," *Black Sea J. Eng. Sci.*, vol. 8, no. 4, pp. 1160–1180, Jul. 2025.
[22] F. Jalalvand, M. Baruwal Chhetri, S. Nepal, and C. Paris, "Adaptive alert prioritisation in security operations centres via learning to defer with human feedback," 2025, *arXiv:2506.18462*.
[23] M. Wiik Eckhoff, P. Marius Flydal, S. Peters, M. Eian, J. Halvorsen, V. Mavroeidis, and G. Grov, "A graph-based approach to alert contextualisation in security operations centres," 2025, *arXiv:2509.12923*.
[24] Q. Tang, X. Di, X. Liu, L. Cong, W. Ren, and Z. Ni, "DeepARR: Alert risk rating based on deep learning," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. with Appl. (ISPA)*, Oct. 2024, pp. 2097–2104.
[25] F. Jalalvand, M. Baruwal Chhetri, S. Nepal, and C. Paris, "Alert prioritisation in security operations centres: A systematic survey on criteria and methods," *ACM Comput. Surv.*, vol. 57, no. 2, pp. 1–36, Feb. 2025.
[26] M. Baruwal Chhetri, S. Tariq, R. Singh, F. Jalalvand, C. Paris, and S. Nepal, "Towards human-AI teaming to mitigate alert fatigue in security operations centres," *ACM Trans. Internet Technol.*, vol. 24, no. 3, pp. 1–22, Aug. 2024.
[27] B. Gelman, S. Taoufiq, T. Vörös, and K. Berlin, "That escalated quickly: An ML framework for alert prioritization," 2023, *arXiv:2302.06648*.
[28] X. Lin, G. E. Avina, and J. Santoyo, "Reducing false alerts in cybersecurity threat detection using generative AI," in *Proc. 4th Workshop Artif. Intell.-Enabled Cybersecurity Anal.*, 2024, pp. 1–6. [Online]. Available: https://ai4cyber-kdd.com
[29] (Jun. 2020). *The State of SOC Effectiveness: Signs of Progress but More Work Needs to Be Done*. [Online]. Available: https://www.ponemon.org/research/ponemon-library/security/the-state-of-soc-effectiveness-signs-of-progress-but-more-work-needs-to-be-done.html
[30] J. Oliver, R. Batta, A. Bates, M. Adil Inam, S. Mehta, and S. Xia, "Carbon filter: Real-time alert triage using large scale clustering and fast search," 2024, *arXiv:2405.04691*.
[31] P. Cichonski, T. Millar, T. Grance, and K. Scarfone, "Computer security incident handling guide," NIST, Gaithersburg, MD, USA, Tech. Rep. 800-61r2, 2018.
[32] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 9459–9474.
[33] Y. Lin, Y. Liu, F. Lin, L. Zou, P. Wu, W. Zeng, H. Chen, and C. Miao, "A survey on reinforcement learning for recommender systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 10, pp. 13164–13184, Oct. 2024.
[34] A. Morgan. (Apr. 2025). *15 SOC Metrics Every CISO Must Track to Prove and Improve Security Performance*. Accessed: Jan. 8, 2026. [Online]. Available: https://datacipher.com/soc-metrics-every-ciso-should-track/
[35] G. Oviatt. (May 2025). *SOC Metrics & KPIs That Matter: MTTR, MTTD, MTTI, False Negatives, and More*. Accessed: Jan. 8, 2026. [Online]. Available: https://www.prophetsecurity.ai/blog/soc-metrics-that-matter-mttr-mtti-false-negatives-and-more
[36] S. A. Chamkar, Y. Maleh, and N. Gherabi, "The human factor capabilities in security operation center (SOC)," in *Advances in Information, Communication and Cybersecurity*, Y. Maleh, M. Alazab, N. Gherabi, L. Tawalbeh, and A. A. Abd El-Latif, Eds., Cham, Switzerland: Springer, 2022, pp. 579–590.
[37] A. Basta, N. Basta, W. Anwar, and M. I. Essar, *Open-Source Security Operations Center (SOC): A Complete Guide To Establishing, Managing, and Maintaining a Modern SOC*. Hoboken, NJ, USA: Wiley, 2024.

AHMED ABUAZIZ received the B.Sc. degree in computer systems engineering from the Palestine Technical College, and the M.Sc. degree in computer engineering, specializing in information security and cryptography from the Islamic University of Gaza, in 2015. He is currently pursuing the Ph.D. degree in computer engineering with FMV Işık University. He is an enthusiastic Senior Information Security Engineer with extensive experience in network engineering, systems administration, enterprise security solutions, software development, and information security training and consulting. He has a strong professional interest in systems security and cloud security. He is also a Senior Information Security Engineer at one of the largest companies in the Middle East. His research interests include cybersecurity, AI-driven security solutions, security operations center (SOC) operational optimization, and agentic automation for security operations.

BARIS CELIKTAS received the B.S. degree in systems engineering from the National Defense University, in 2008, the M.S. degree in applied informatics from Istanbul Technical University, in 2018, and the Ph.D. degree in cybersecurity engineering and cryptography from the Institute of Informatics, Istanbul Technical University, in 2022. He is currently an Assistant Professor with the Computer Engineering Department and the Director of the Cybersecurity Graduate Program, Işık University. In addition, he is a Cybersecurity Consultant and an Architect, specializing in enterprise cybersecurity and cryptography solutions, cloud security, risk management, and governance. He holds several industry-recognized certifications, including CISSP, CCSP, CISM, CISA, CRISC, AAIA, SSCP, CCNP, Security+, CySA+, CIEH, and ISO/IEC 27001, 22301, 20000, 27701, and 42001 Lead Auditor/Lead Implementer credentials, as well as GDPR DPO and NIST cybersecurity consulting credentials. His research interests include cybersecurity, network security, cloud computing, cryptography, malware analysis, risk management, and security applications.