SCIENTIFIC CULTURE
www.sci-cult.com

SCIENTIFIC CULTURE, Vol. 12, No. 4, (2026), pp. 1853-1865
Open Access. Online & Print

DOI: 10.5281/zenodo.12426265

INCIDENT COPILOTS: USING LLMS TO ACCELERATE TRIAGE AND HANDOFFS

Sathwik Rao Sirikonda[1]*, Ashish Garg[2], Kunal Arya[3], Shantanu Barde[4]

[1]Sunnyvale, California USA. ORCID: 0009-0005-9334-9870, sathwik.sirikonda47@gmail.com
[2]Bentonville, Arkansas USA. ORCID: 0009-0003-4402-9593, ashishcoer@gmail.com
[3]Livermore, California USA. ORCID: 0009-0008-8766-8500, kunalarya9@gmail.com
[4]Frisco, Texas USA. ORCID: 0009-0003-5990-0441, shantanu_vnit@yahoo.com

Received: 25/06/2025
Accepted: 08/02/2026

Corresponding Author: Sathwik Rao Sirikonda (sathwik.sirikonda47@gmail.com)

ABSTRACT

Incident response in a large organization depends on the speed of triage and handoffs between the SRE and SecOps units. In contrast, tool fragmentation and the high volume of notifications can impair decision-making and lead to poor documentation. A Sev-1 can generate 50 or more monitoring and SIEM alerts. Initial traffic in the war room will generate 60-100 alerts, and 500 or more chat messages will lead to redundant context acquisition and inconsistent summation. The current study explains incidents in which LLM-powered assistants, applied in chats, aid copilots, the ticket section, and SIEM/SOAR processes, and on call, typically guided by retrieval-enhanced generation (RAG) using runbooks, postmortem, and CMDB datasets. The suggested evaluation will be based on either a practical before/after design or an A/B test of the actual incident data (N ≥ 200 and above) to estimate the measures of MTTA, TTFC, and handoff latency, MTTR (median and p90), document completeness (0-10 rubric), and time saved by an analyst (target <5%). The implementation map is based on access-sensitive retrieval (top-k 5-20), recall@k ≥ 0.80, citation coverage ≥ 0.90, and human verification of 50-100 Copilot documents per month to manage hallucination, privacy, and other operational constraints, with the least privilege principle applied. The expected outcomes include faster classification, shorter time-to-context, and higher-quality production of handoff artifacts by rendering citations mandatory, redacting information, and managing the human-in-the-loop. Research in the future needs to progress to semi-autonomous, constrained remediation plans verifiable against standardized benchmarks.

KEYWORDS: Incident copilots, Large Language Models (LLMs), Retrieval-Augmented Generation (RAG), Triage automation, Handoffs / handoff latency, MTTR (time-to-restore-service).

1. INTRODUCTION

Modern incident response in large organizations is a growing field that includes Site Reliability Engineering (SRE) and Security Operations (SecOps), as outages and cyber incidents often share the same monitoring and infrastructure and follow the same escalation paths [1]. The operational teams usually plan in real time using chat systems, virtual war rooms, pages, and ticketing systems while simultaneously interpreting logs, traces, alerts, and security detections. Mature programs manifest this coordination through incident roles, apparent communication rhythms, and the taking in of information during learning. For example, an incident response process based on the principles of the Incident Command System depicted by Google in its Incident Management Guide will emphasize the three Cs of the Incident Command: coordinate, communicate, and control, such that an incident manager will be capable of ensuring the quality of decision-making in the time of decision uncertainty.

Despite the process maturity, triage has become a bottleneck, where responders have to search through large, heterogeneous volumes of signals to determine the scope, severity, and direction of the next investigation. In most companies, a single Sev-1 can generate 50 or more alerts across monitoring and SIEM pipelines within a short period. Repetitive context gathering is a common failure mode, in which different engineers rebuild the same timeline: what changed, which services have been falling, which entities are involved, who has mitigated the channeling, and what mitigations have been attempted. Handoffs among shifts or between teams often exacerbate the issue when incident notes lack the operationally critical fields of business impact, affected systems, current hypotheses, mitigation status, and explicit next actions. Large Language Models (LLMs) can be effectively deployed for summarization and information synthesis tasks, although they also pose new threats [2]. False information hallucinated may lead the investigation astray and leak confidential information unless access control mechanisms are applied to the retrieval and production routes of the copilot.

This study introduces the term incident copilots, which describe the assistants provided by the LLM integrated into the incident workflow to improve triage speed and the quality of the handoff. It also introduces an effective reference architecture that includes retrieval-augmented generation (RAG), tool integrations (e.g., querying observability platforms and ticketing systems), and policy controls (e.g., least privilege, redaction, and audit logging). A set of operational metrics (mean/median time to acknowledge (MTTA), time to first context in the ticket, handoff latency, and time to restore service (which is frequently defined in terms of MTTR or time-to-recovery) and an explicit qualitative scoring of incident summaries are also outlined in the study as a suitable plan of measurement and statistical evaluation that should be used in the real environments. It also benchmarks these objectives against industry-measurable product trends, including Microsoft Security Copilot features, incident summarization and guided response within Defender processes, PagerDuty AI agent placement, and Atlassian Intelligence, which they can use to generate post-incident analyses.

Areas of the study include speeding up triage with an LLM, producing incident summaries and situational updates, developing handoff objects for change or within the team, and writing the content for a post-incident review. The work assumes that humans are the owners of incidence choices, and it concentrates on copilot behavior, indicating suggestion, summary, and retrieval of decisions, and not their independent implementation. Wholly automated remedial measures, not subject to express human approval, especially measures that modulate the position of production, are regulated as research in the future due to governance, risk, and liability constraints.

This study is structured into several chapters. The Literature Review provides an overview of existing practice in incident management and a new basis for LLM copilots, RAG, and trust/safety controls. Methods and Techniques describes the data collection, structural aspects, and a statistical evaluation plan that may be applied to incident data from popular operational tools. Discussion provides an interpretation of expected benefits and limitations, including the measurable consequences of handoffs and triage, and points to the actual problems encountered during deployment. Future Research Recommendations involve improvement of benchmarking, causal analysis, and non-expansive partial self-remediation. The research paper ends with a summary of practical evidence on SRE and SecOps leaders using incident copilots.

2. LITERATURE REVIEW

2.1 Incident Management Foundations and Metrics

The lifecycle was a straightforward process of detection and analysis (triage), mitigation, recovery, and post-incident learning, as indicated in the literature on incident management. The Site Reliability Engineering (SRE) idea, integrated into the enterprise architecture, shifts towards preemptive resilience, fulfilling telos and service-level objectives (SLOS), and feedback-oriented measures to mitigate the risk of service degradation [3]. In practice, service recovery time (MTTR) is directly

proportional to the latency to detection and the efficiency of triage. Large companies with operational data are highly skewed in incident time, with median incident resolution times (e.g., 45-120 minutes, Sev-2) and thick tails (e.g., 8-24 hours). Such skewness should be reported with robust statistics, including the median, interquartile range (IQR), and p90/p95 percentiles, rather than simple arithmetic means.

Measured reliability is progressively associated with software delivery performance measures. Modern DevOps models relate incident recovery to operational metrics, such as deployment and change failure rates. Agile processes based on trends and predictive metrics. Analytics pipelines with data, such as real-time Jira analytics with JQL, are shown to support useful ticket presentation and support agile processes. The integration is implemented in incident management and provides near-real-time dashboards on the frequency and recurrence of escalations, and MTTA and MTTR. However, the metric evolution must take into account statistical validity, seasonality, and biases, such as infrastructure development or the sensitivity of the monitoring [4; 5]. A feedback loop that transforms post-incident results into architecture and better control is needed so that the whole enterprise becomes seamless, as the predictive resilience models indicate.

Table 1: Effective statistical reporting patterns and incident lifecycle measures of SRE-congruent measures of reliability under skewed incident duration distributions.

Core Aspect	Practical Meaning in Operations	Metrics / Statistical Treatment	Implementation Notes / Examples
Incident lifecycle progression	Incidents typically move from detection → analysis/triage → mitigation → recovery → post-incident learning, requiring disciplined coordination across stages.	Track stage timestamps to derive MTTA, MTTR, escalation frequency, and recurrence.	Use ticketing + paging + comms logs to timestamp each lifecycle transition for consistent measurement.
SRE-aligned predictive resilience	Enterprise architecture aligned with telemetry, SLOs, and feedback loops aims to reduce disruption duration by enabling earlier detection and faster triage.	Monitor SLO breaches and recovery times; relate detection latency and triage efficiency to service restoration time (MTTR).	Maintain service catalogs and SLO dashboards so responders can quickly map alerts to business impact and owners.
Skewed incident duration distributions	Real-world incident durations are often positively skewed, with typical Sev-2 medians ~45–120 minutes and rare heavy-tailed events lasting 8–24 hours.	Prefer median, IQR, and p90/p95 over means to avoid distortion from outliers.	Use percentile-based reporting in dashboards and reviews; segment by severity to keep comparisons valid.
Reliability metrics linked to delivery performance	Recovery outcomes correlate with delivery practices (e.g., frequent changes can affect failure rate), so reliability is measured alongside delivery performance.	Combine incident metrics (MTTA/MTTR) with delivery indicators like deployment frequency and change failure rate.	Real-time analytics pipelines (e.g., Jira + query integrations) can power near real-time dashboards for MTTA/MTTR and recurrence, while accounting for seasonality and confounders.

Table 1 summarizes the incident management and Metrics talk in a tabular format, naming each underlying concept, operational definitions, quantifiable indicators, and the implementation direction. It involves an incident lifecycle of detection and triage, mitigation, recovery, and post-incident learning. Nonetheless, it focuses on telemetry-based predictive resilience supported with SLOs and feedback. The table also indicates that reliability measurement relies on MTTR centrality and explains that incident duration data are generally skewed, thereby necessitating the use of nonparametric statistics, such as median, IQR, and p90/p95 percentiles. It is also easy to monitor reliability because it implements DevOps delivery indices and near-real-time analytics dashboards.

2.2 LLM Copilots for Incident Triage and Investigation

According to recent research on machine learning in operational pipelines, AI can significantly accelerate large-volume operations by automating pattern identification and anomaly detection. Summaries of multi-source telemetry, extraction of entities from logs and tickets, inferences about event-related alert signal relationships, structured playbook actions, and observability queries are all processes for which LLM copilots can be highly helpful during incident response (IR). Concisely, when firms integrate the Microsoft security platform, they have been able to produce artificial intelligence-enabled incident summaries that compress numerous alerts into a single count, saving time during analyst reviews [6]. Another software solution created by Atlassian is post-incident review (PIR), an AI-generated tool that analyses a report based on a Jira Service Management ticket. The other uses of AI in PagerDuty involve reducing incident lifecycles by conducting triage and resolving incidents.

The copilots will reduce cognitive load by transforming unstructured step-by-step inputs into a structured summary, which may include impact statements, affected services, and recommended next actions. When the typical Sev-1 produces both 60-100 alert events and 500+ chat events in the first hour,

automated summarization will achieve an over 30% reduction in the cost of manual reading time, since manual reading is linear at 200-250 words per minute. Productivity has also been found in ETL pipeline systems and machine learning add-ons, and the automation produced has minimized human verification work and improved error-detection performance in fraud-detection systems to 0.90 accuracy [7]. These similarities indicate that parallel augmentation is feasible to IR, though verification and control mechanisms are not to be neglected.

2.3 Retrieval-Augmented Generation (RAG) and "tool-using" copilots in ops

Retrieval-Augmented Generation (RA) architectures extend LLMs' capabilities to include formal sources of knowledge, such as runbooks, configuration management databases (CMDBs), service catalogs, and past incident reports. A hybrid implementation using all retrieval tools, called Arch historical incident similarity search, filters the query by keyword and vector similarity (kNN) and can reveal the similar root causes of a selected event in only a few seconds within the operational environment, thereby expediting hypothesis formulation [8]. Real-time analytics integration schemes also suggest that query execution can be satisfied across thousands of artifacts with sub-second latency by integrating query languages with data warehouses.

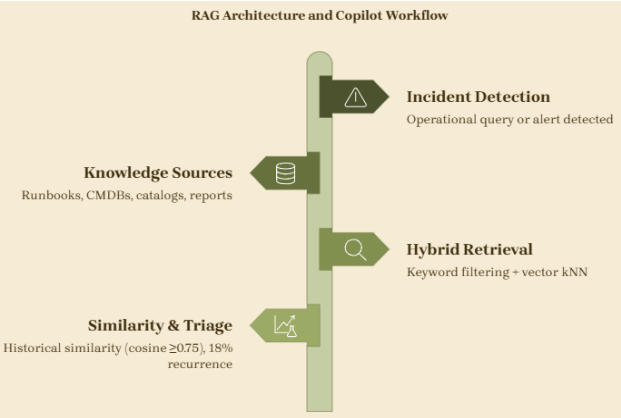

Figure 1: RAG copilot workflow: Incident detection triggers hybrid keyword + vector kNN use of runbooks/CMDBs, permits similarity-based triage (cosine>=0.75), and recurrence-based hypothesis.

Figure 2 shows a Retrieval-Augmented Generation (RAG) architecture and workflow in which event alerts associated with incident detection trigger structured knowledge retrieval, including runbooks, CMDBs, service catalogs, and previous incident alerts. The hybrid retrieval is depicted in the figure, where keyword search is combined with a vector kNN similarity search to retrieve relevant artifacts within seconds [9]. The triage process produces the hypotheses more rapidly. It involves sorting historical events by their similarity to service identifiers and error signatures, using a cosine threshold of 0.75. It also displays the pattern of the recurring signal's capability (0.18 or 18%) to give precedence to probable causes and reinforce recognizable feed laws.

In practice, a copilot boosts the k-top historical events that correspond to service identifiers and error signatures, has a cosine similarity of 0.75, and forms a hypothesis that describes such artifacts. Such a retrieval would highly assist in the triage prioritization, in case a failure mode returns at a rate of 0.18 (18%). Introducing the predictive resilience principle also entails placing these copilots within the enterprise architecture to improve observability, alignment, and architectural feedback loops.

2.4 Risks, Governance, and Trustworthiness in High-Stakes Copilots

Although the efficiency of deploying LLM copilots has been guaranteed, the literature identifies the lack of critical thinking about governance issues in the deployment of automated systems in high-stakes situations as a systemic risk. Unsupported claims in the form of hallucinations are unreliable and can lead to misinformed decision-making by response teams, especially when the team exhibits a wide range of beliefs about the outputs. Limited generation, use of citations, and comparison with authoritative data are among the mitigation measures. The ML-enhanced ETL configurations reduced the propagation of errors by multiple validation and anomaly-detection layers, creating safer settings; similar protection must be applied to incident copilots [10].

Security/Privacy are also of great concern. The datasets for operations are sensitive in both configuration and personal information. Limited use of privileges, role-based access control, data minimization, and access controls, including retention controls and audit logs, are thus compulsory. Research in the domain of cross-domain governance shows that the risks of spreading sensitive information are greater when flows involve non-monolithic organizational types, unless regulatory schemes are defined unambiguously. Such rigidity in administration, particularly in incident response, may help establish accountability and traceability for AI-aided actions.

2.5 Evaluation Approaches Used in Practice

Measurements of productivity reported by vendors are initial indications of value that have to be

verified. The self-reported time savings of at least 40% or efficiency gains (at least a 60% reduction) should be put in terms of the underlying work balance, incident combination, and measurement design. Such cases would require a statistically significant assessment, including a reasonable sample size (e.g., N ≥ 200 incidents), a pre-/post analysis, and a confidence interval (95% CI) for the effect sizes of observed MTTR reductions [11]. Matched-cohort/difference-in-differences analysis is a strong experimental design that addresses bias arising from processes that accompany concomitant changes.

Table 2: An overview of applied verification of vendor productivity claims by adequate sample sizes (N≥200 plus), pre/ post analysis of MTTR by 95% CI, and bias-minimizing experimental design.

Evaluation Point	Operational Interpretation	Quantified Requirements / Examples	Recommended Study Methods
Vendor productivity claims need verification	Reported gains are indicative, but not definitive without independent measurement and controls.	Self-reported time savings ≥40%; efficiency improvements ≥60% (reported reductions).	Treat as hypotheses; validate locally against baseline performance.
Contextualize results to avoid misleading conclusions	Improvements must be interpreted relative to baseline workload, incident mix, and measurement design.	Compare outcomes by severity/service; ensure comparable incident classes before claiming impact.	Use stratification and consistent definitions of metrics across periods.
Ensure statistically meaningful evaluation	Reliable inference requires sufficient sample size and uncertainty reporting around outcomes.	N ≥ 200 incidents; report 95% CI around observed MTTR changes.	Pre-/post analysis with confidence intervals and clear baselines.
Reduce bias from concurrent process changes	Simultaneous tooling/process shifts can confound results if not controlled.	Confounding risk increases during overlapping initiatives (e.g., monitoring retuning, staffing changes).	Matched cohort analysis or difference-in-differences to isolate copilot effects.

Table 2 outlines the approach to interpreting and measuring the influence of copilot incidence assessment in practice. It also suggests the need to receive vendor productivity deliverables as a reference point, which will need extraneous validation rather than definitive evidence. The major quantitative arguments most of them put forward (the time saved is ≥ 40, and efficiency increased ≥ 60) are reported, and it is necessary to keep in mind that such numbers have to be prepared in the context of the workload at that moment, the mix of incidents, and how the measurements are designed. It also provides insight into the statistical requirements for a plausible approach, including adequate sample sizes (e.g., N ≥ 200 events), a pre- and post-comparison, and 95% confidence intervals for the alterations in MTTR. It concurs with desirable experimental procedures, such as matched cohorts and difference-in-differences, to reduce bias.

2.6 Research Gaps and Limitations

Although the augmentation has been successful, gaps still exist. They also do not have standardized rules regarding factual consistency, the accuracy of Copilot incident information, and the outcomes of resolutions. There was little longitudinal research showing deterioration in MTTR over 12 months. The socio-technical implications of automation bias and role redefinition within the SRE team have been empirically studied. The non-heterogeneous integration of enterprise architecture has not received sufficient analysis, especially given the cross of 10,000+ services tracked. These loopholes are instrumental in the execution of systems that succeed in statistical examination and that perform in operationally vital governance settings.

3. METHODS AND TECHNIQUES

3.1 Data Collection Methods

The theoretical basis of incident copilot assessment presupposes that the source-based production settings are planned in the multi-source production record operational records. The minimum viable data includes incident tickets in services such as ServiceNow or Jira Service Management, severity levels (e.g., Sev1-Sev4), dates (opened, acknowledged, mitigated, and resolved), service identifiers, and service owners. The average and median time-to-acknowledge (MTTA) and time-to-restore-service (MTTR) are among the variables that can be computed in such structured fields. The use of collaboration software logs (chat or war-room) is also documented in tools such as Slack or Microsoft Teams to examine context reconstruction and handoff completeness. Such logs must be stored under a consent policy and within storage limitations (e.g., 180 days), and the identifiers involved must be anonymized to comply with data governance requirements.

The third category of databases is the observability signals. Such monitoring announcements (e.g.,

Datadog, Prometheus), distributed tracing announcement traces, PagerDuty or Opsgenie5 events, runbook execution logs, etc., are scraped to gauge signal quantity and escalation patterns [12]. Hypothetically, a Sev1 event in the cloud-native system can generate 40-120 alerts during the first 30 minutes, but a Sev2 event can produce 10-20 alerts. The knowledge repositories are indexed to facilitate retrieval-augmented generation (RAG) in the form of runbooks, postmortems, known error databases, and configuration management databases (CMDBs).

The sampling plan must be statistically valid. It is recommended that a target population of N ≥ 200 - 1000 incidents in a period of three to twelve months will guarantee that the variability and seasonality impact are captured. Taking an average duration of 25 Sev1/Sev2 cases in 1 month, a 6-month time frame is sufficient to do a nonparametric examination of the hypothesis with an alpha of 0.05 with a size of effect of moderate and significant (d 0.4). The severity stratification is essential; integrating Sev1 and Sev3 incidents invalidates the comparison results, since they differ significantly in their expectations for response. This strategy is consistent with predictive risk modelling that uses stratified sampling, leading to greater improvements in variance and corrective inference in the high-stakes operating system.

3.2 Copilot Implementation Blueprint (Reference Architecture)

The architecture of the copilot is on a Retrieval-augmented generation (RAG) pipeline with operational tools. Organized and unstructured artifacts, runbooks, postmortem reports, and incident histories are placed in a vector database that provides access-sensitive retrieval controls. Retrieval parameters are then empirically optimized over a top-k range of 5-20 documents. The criteria for determining the quality of the retrieval will be recall@k ≥ 0.80 and citation coverage ≥ 0.90, indicating that ≥ 90% of the factual statements were mentioned in the sources from which the information was retrieved. They are also achieved through access control, so that only artifacts permitted by the role-based policies can be accessed.

Tool integrations are of two levels. The former provides only read-only access to deployment history, service health, and the last configuration updates. The second tier is human approval before implementation and entails drafting changes to criminal charges, proposed communication templates, or descriptions of playbook activities. CI/automation pipeline patterns also include layers of automation, and the validation stages of its financial data should be gated for approval to avoid legitimate deployments. The introduction of this paradigm to incident copilots would mean that AI cannot autonomously create advisory actions.

Guardrails are mandatory. Generated summaries should reference retrieved artifacts to make claims about both timelines and systems affected [13]. Redaction filters filter the model's input to remove Secrets or personal information. Policy validation engines define pre-response controls to eliminate data leakage. These controls are concurrent with the distributed system governance mechanisms, which guarantee multiparty consent and fault-tolerance leading to the election of leaders before state change and thus system stability.

3.3 Evaluation Design

Three suggested complementary study designs are provided. The former is a quasi-experimental comparative study of 8-12 weeks before rollout and 8-12 weeks after rollout. Rolling averages are used to counteract seasonal changes and time-series changes in load. The A/B design will be applied to on-call rotations, with half of the shifts using Copilot and the other half using regular workflows. The use of randomization reduces the contamination bias. The third method is the matched incident analysis, in which incidents are compared by service, severity, and time of day to remove confounding factors.

To achieve statistical power, assuming a baseline MTTR median of 120 minutes, and a hypothesized 15% reduction (Δ = 80 minutes), a minimum of 90-120 similar incidents per group is required to achieve 80% power under the log-normal assumption. This experimental method is similar to the one that will be used to evaluate the operational efficiency of AI-based service institutions, in which artificial comparisons are used to assess the advantages of automation in real-world settings.

3.4 Data Analysis

Primary customary activities are the MTTA, time to first meaningful context (TTFC), handoff latency, MTTR (presented as median and p90), and documentation completeness score (0-10 rubric). The period between when the ticket is formulated and the title-structured presentation involving the influence and probable cause is called TTFC. The period between the ownership transfer and the subsequent activity is called handoff latency, the time that elapses between the handoff and the investigators' activity. When distributions are right-skewed, the median and interquartile range (IQR) are reported, along with p90 and p95.Hypothesis testing uses the Mann-Whitney U

test for nonparametric comparisons, or the log-normal regression as a multivariate adjustment.

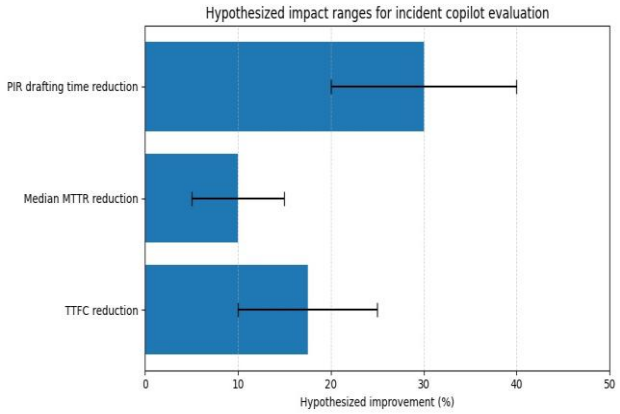

Figure 2: Hypothetical incident: copilot impact. Ranges: TTFC loss will be 10-25%; median MTTR loss will be 5-15%; and PIR drafting time loss will be 20-40% across similar incidents.

Figure 2 depicts the scope of hypothetically predicted performance effects applied in testing incident copilots using the data analysis framework. The graph summarizes the expected percentage changes in three main results: time to first meaningful context (TTFC), median time to restore service (MTTR), and time to draft post-incident reviews (PIR). The study hypothesizes that the copilots will be able to reduce TTFC by 10-25% faster context synthesis, 5-15% median MTTR, and 20-40% PIR drafting effort through automated summarization. These ranges are used to accomplish statistical testing using strong measures and direct comparisons.

When a control group is present, copilot effects of differences in process improvements in the background are decomposed using difference-in-differences estimation. Their effect sizes (Cohen's d) and 95% confidence intervals are reported. Target effects may be expressed in the hypotheses: 10-25% decline in TTFC, 5%-15% decline in median MTTR between groups of the same severity, and 20%-40% decline in post-incident review preparation time. These boundaries represent the efficiency advantages of extensive automation, where AI-enhanced processes increase throughput and reduce overload on human operators by the same orders of magnitude [14].

3.5 Reliability, Safety, and Quality Controls

In reliability testing, known-answer incident datasets containing ground-truth timelines and root causes are recorded. The rate of factual errors is calculated (errors 100) and aims to be less than 5%. The monthly human-in-the-loop quality control includes scoring 50 to 100 sampled outputs on a 5-point scale for accuracy and relevance. The percentage of AI suggestions rejected by responders and the share of incidents that need to be reworked due to misleading AI output are the operational risk key performance indicators. Both metrics must move towards 0% over the course of the quarters. Predictive risk models rely on continuous monitoring and feedback mechanisms to re-adjust the models once their performance falls below an acceptable threshold [15]. All these controls, combined, assure incident copilots, since they are accountable, statistically triangulated decision-support systems, not opaque automation layers.

4. DISCUSSION

4.1 Where Incident Copilots Accelerate Triage

Incident copilots have been shown to hasten triage by increasing cognitive load during the initial analysis stage. In high-volume applications, a single Sev1 notification can generate 50-150 monitoring, logging, and security system alerts in the initial hour. Auto-correlation involves the responders browsing dashboards, traces, and deployment histories in sequence. The summary of these artifacts to determine which services might be impacted, error codes compatible with each other, the latest configuration modifications, and potential owners of the changes is achieved by using the summarization and entity extraction functionalities of Copilot. Copilots also support correlated anomalies exposure to near real-time, so when run on streaming pipelines to perform telemetry, including in real-time implementation of Spark and Kafka applications, allowing the ability to accept and otherwise process millions of records per minute with latencies in the single-second scale, structured alert feeds can be ingested by Copilots [16].

Structured questions such as "What was different during the past 60 minutes? What services are sharing this dependency? Who owns the failing component?" are included in the copilot system and help to organize the investigation process and minimize potential omissions. The coordination of control and threat mitigation is determined by establishing consistency in rule enforcement across gateways and identity layers in distributed systems. Threat prevention research in API gateways has identified centralized inspection as a mechanism to reduce the attack surface and latency by implementing similar policies alongside validation [17]. Incident copilots also serve as knowledge selection filters, composed of reasoning rules based on patterns, to reduce discontinuity in analysis. By empirically experimenting with a 25-minute first meaningful context (TTFC) baseline period, structural summarization that results in a 6-10 minute reduction in reviewing redundant log data

will yield a 24-40% rise in TTFC constant workload and severity distribution.

4.2 Handoffs: Measurable Improvements and Failure Modes

The problem of shift-based operations has also been referred to as a context-reset loop, where new responding officers repeat investigative actions based on truncated notes. The handoff process may take 15-30 minutes, even after an ownership change, when incidents are not documented properly. The uncertainty will be reduced through meeting on-copy reports, now containing a short impact statement, a date, and proposed next steps, which will expedite re-engagement. If the median handoff latency decreases from 20 to 14 minutes (Δ = -6 minutes), this is a 30% improvement. The discounted benefits of 1,800 minutes of senior engineer capacity are achieved through 300 cross-shifting cases per year.

The quality improvement rating can be measured using a 0-10 rubric for documentation scoring. The mandatory identification of the product recall will limit mistakes due to time lapses and unreasonable arguments. Identity integration research can examine the significance of using a high degree of federation and authentication to avoid unwarranted data disclosure in enterprise SaaS integrations. By employing a similar level of screening when duplicating copilots, there is a risk of relaying erroneous or sensitive information that would otherwise be evaded during the handoff [18]. Nevertheless, failure modes are also present. Excessive use of auto-generated summaries can lead to limited verifications, where respondents may believe the system is authentic and avoid making preliminary logs. Nevertheless, it still needs pre-determined controls and regular audits to implement trust calibration.

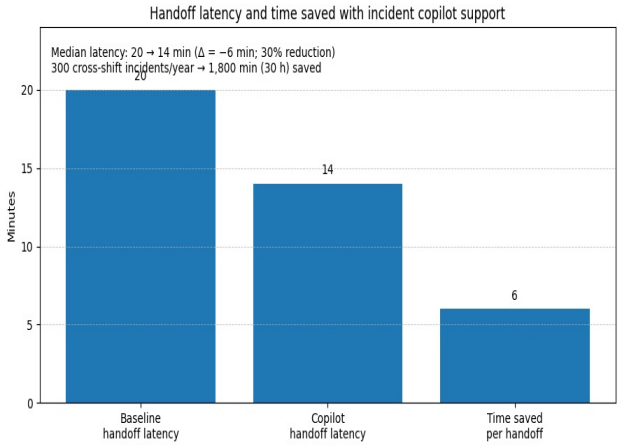

Figure 3: Graphical presentation of the baseline (20 minutes) handoff latency and the copilot-assisted latency (14 minutes) with a resultant decrease of 6 minutes over handoff and an annual effectiveness.

Figure 2 presents a quantitative perspective on handoff between inter-shift incident operations, with an incident copilot summary showing reduced context reset loops and enhanced re-engagement. The graph shows the comparison between the median handoff latency baseline (20 minutes) and the copilot-assisted (14 minutes), with an absolute difference of 6 minutes (Δ = -6) and a relative change of 30%. It also shows saved time per handoff, and that is much more with size. With a yearly total of 300 cross-shift incidents, a person will save 1,800 minutes, or 30 hours of senior engineer staff to conduct investigations and mitigation actions.

4.3 Real-World Adoption Patterns and Examples

Big vendors have incorporated generative AI into the incident process. Microsoft Defender and Sentinel habitats assist with incident analysis, provide AI-driven recommendations for incident response, and display them on security operations screens [19]. The characteristics incorporate the alerts and other details in the surroundings to encourage prioritization. Atlassian Jira Service Management also includes a post-incident preview, an artificial intelligence-enhanced synthesis of past ticket reports into a structured report that analyzes the impact and adherence to the action plan. PagerDuty has taken generative AI services through the lifecycle, such as incident records and implementation mechanisms.

These architectural instantiations resemble the scales of calm time-process architecture, which ingest distributed telemetry broadcasts and turn them into works of choice-assistance. The Spark-Kafka-financial pipeline shows that throughput and latency must be modelled to achieve scalability and reliability. This is also applicable in security operations, where thousands of alerts are generated daily. Governance measures, including identity federation and access control patterns, can ensure that the copilot retrieval layers do not exceed the enterprise authentication and authorization limits.

4.4 Economics and ROI

The time savings used to calculate the return on investment must be converted into dollars. Suppose that a company has 1,000 actionable incidents each year. Supposing that the copilots will save 15 minutes of average triage and documentation work per incident, the savings will be 15,000 minutes (250 hours) annually. This has a direct labor savings of 22,500 with a fully loaded analyst costing of 90 per hour. Further value could be obtained due to shorter incident time. A 10% reduction in median MTTR (from 120 minutes to 108 minutes) reduces service

downtime by 12 minutes per incident. When the estimated impact of a business is that outages will cost the business up to $1,000/minute, even a 5% reduction in the number of outages per year can yield significant economic benefits.

Costs, however, must be included. Depending on scale, model inference costs, hosting database costs in vector format, integration development, red-teaming, compliance audits, and governance overhead range from $50,000 to $150,000 per year. The study of cross-border compliance and quality assurance in semiconductor manufacturing highlights the introduction of non-trivial overheads into the operation of a globally distributed environment, driven by regulatory requirements [20]. The same compliance costs are used in cases where the copilots manipulate sensitive operational data. Therefore, net ROI should be calculated as (labor + downtime savings minus operational costs), with sensitivity analysis on adoption scales.

Table 3: An overview of incident copilot ROI assumptions, which measure labor and downtime formed by decreasing triage and MTTR, yearly operating expenditures, and the net ROI formula.

ROI Component	Assumption / Input Values	Quantified Result	Practical Interpretation
Annual incident volume	1,000 actionable incidents per year	Basis for yearly calculations	Establishes the workload scale used to estimate time and cost savings.
Analyst time saved per incident	15 minutes saved per incident (triage + documentation)	15,000 minutes saved per year = 250 hours/year	Indicates recovered analyst capacity that can be reallocated to investigation, prevention, or engineering work.
Direct labor savings	Fully loaded analyst cost = $90/hour	250 hours × $90 = $22,500/year	Converts time saved into direct labor cost savings under stated staffing cost assumptions.
Downtime reduction from improved MTTR	Median MTTR reduced by 10% (120 → 108 minutes)	12 minutes less downtime per incident	Demonstrates how faster resolution can reduce service disruption duration per incident.
Business impact of outages	Outage cost = $1,000/minute	Value scales with minutes reduced; even 5% fewer outages/year yields material benefit	Shows that downtime-related savings can dominate ROI when outage cost per minute is high.
Annual operational costs of copilot program	Model inference + vector DB hosting + integrations + red-teaming + compliance + governance	$50,000–$150,000/year	Captures total cost of ownership, including security and regulatory overheads for sensitive operational data.
Net ROI calculation approach	Net ROI = (labor savings + downtime savings − operational costs)	Requires sensitivity analysis across adoption scales	Encourages evaluating multiple scenarios because savings and costs vary by incident mix, outage cost, and deployment scale.

Table 3 presents a quantitative decision-making model to estimate the economics and ROI of incident copilots, translating time and reliability gains into monetary value and accounting for operational expenditure. It indicates 1,000 actionable cases per year, with 15 minutes per case, for a total of 15,000 minutes (250 hours) per year. This will save a total of $22,500 in direct labor at a fully loaded rate of $ 90/hour for an analyst. It also absorbs reliability upside due to a 10% median reduction in the MTTR (120 to 108 minutes), and a time (outage) reduction of 12 minutes per incident, and the outage impact is at 1,000 per minute, and extra profits due to eliminating further outages by 5%. It also includes the annual costs of the programs (50,000-150,000) and the net ROI (labor + downtime savings-operational costs), which is presented in terms of sensitivity analysis.

4.5 Limitations and Operational Risks

Limitations still exist despite quantifiable efficiency gains. Systemic risks are hallucinations, injections that have to be undertaken immediately, and automation bias. Threat examination at the API layer reveals the vulnerability of distributed systems to improperly validated inputs [21]. The incident copilots have to embrace rigorous input sanitization and monitor the amount of data retrieved. Efficacy is also limited to the production of high-quality data and the ultimate quality of the data; the absence of runbooks, unusual ticket labels, or data becoming obsolete in the CMDB reduces retrieval accuracy and leads to false derivations.

These are further problems of metric interpretation. The durations of events follow heavy tails and therefore imply skewed means. This can be greatly enhanced with statistically insignificant enhancement, and no median or percentile reporting. Adjustments to the monitoring threshold, pre-deployment cycles, and infrastructure expansions will confound pre-/post comparisons. Consequently, the requirements for deploying copilots safely in procedure-critical missions include

an experimental design and benchmarking of governance to produce statistically defensible results.

5. FUTURE RESEARCH RECOMMENDATION

5.1 Toward semi-autonomous remediation with hard safety constraints

The future work would shift incident copilots from "Suggest-only" assistance to semi-autonomous remediation, but only under hard, machine-checkable, and profound conditions. One possible way forward is to enforce a verified execution plan: the copilot would need to describe a remediation as a typed sequence of operations (e.g., scale deployment X in 6→12 replicas, rollback release Y, block IP-ranges Z), each of which has/is mapped to runbooks that are pre-verified and a set of preconditions/rollback actions. A staged rollout may be used to restrict the blast radius by establishing canary percentages of 1%, 5%, 25%, and 100%, with automatic rollback when the error rate exceeds the baseline by ≥2σ for ≥5 minutes. The false-positive rate, rollback frequency, and operator override timing should be reported. Studies of automated policy generation, such as reinforcement-learning methods used to generate firewall rules, reveal that the policy actions may be generated but need to be constrained by a safety envelope and validation gates before running [22].

5.2 Standardized benchmarks for incident copilots

One of the major gaps is the lack of different, similar standards. Further research must develop shared, anonymized incident corpora that include tickets and alert feeds, and gold tags for impact, root cause category, and final remediation. A practical target for standardization includes 1000-10,000 cases, stratification by incident severity, and inter-rater agreement κ≥80 on labels. The harnesses to be considered based on summary factuality, action relevance, citation coverage, and time-to-context reduction in simulated triage. This corpora design should also include a model ETL, deduplication, and lineage-tracking to prevent cross-training and cross-testing splits, since near-duplicate instances often occur, as is the case with machine-learning-based ETL pipelines that rely on integrity checks and repeatable transformations [23].

5.3 Advanced causal evaluation

More designs are also required in a cause-and-effect format, beyond before/after, to determine the true effect under varying operating conditions. The rollout can be performed in phases, with copilots progressively integrated into teams (e.g., 12 squads over 24 weeks), and teams serve as their own controls to introduce seasonality. Exogenous variation used in instrumental-variable methods can include a randomized access window, a temporary feature flag, or eligibility disparities across service tiers [24]. Median and p90 results should be reported with 95% confidence intervals, and the primary endpoints (e.g., MTTR, handoff latency, and documentation score) must be pre-registered to reduce the researcher's degrees of freedom. Planning for power must account for heavy-tailed time distributions; achieving a 10% MTTR reduction can require N ≥ 200, or more similar incidents of a condition with large variance.

5.4 Multimodal ops copilots

Multimodal evidence, such as logs, traces, metrics, dashboards (images), topology maps, and chat/ticket narratives, should be incorporated into future copilots. Triaging Multimodal retrieval provides better access to key context when it appears as screenshots of dashboards or runbook diagrams. The most effective way is to turn the visual representations into structured indicators (e.g., panel title, metric name, time window) and index them with the textual ones so that the copilot can reference both. Edge deployment is also applicable in a regulated environment that cannot export telemetry; RAG-enabled edge AI frameworks demonstrate how retrieval and scoring can be performed on-site with controlled data exposure, which can inform on premise incident copilots that they must operate within their latency and privacy constraints.

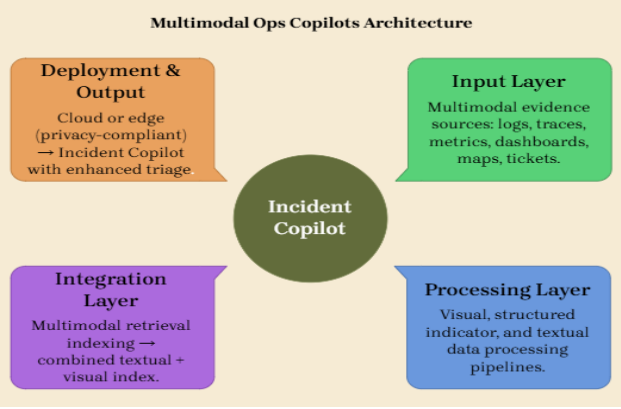

Figure 4: Multimodal ops copilot architecture integration of logs, traces, metrics, dashboards, maps, and tickets; works with visual and text indicators, jointly indexes, and enables, along with privacy-compliant edge deployment.

Figure 4 illustrates a multimodal incident copilot architecture that processes various operational facts, such as logs, traces, metrics, dashboard images, topology maps, and chat/ticket narratives, to enhance triage in situations with important context in screenshots or diagrams. The diagram consists of an input layer of multimodal sources, a processing layer that transforms visual artifacts into structured pointers (e.g., panel title, metric name, time window) and textual pipelines, and an integration layer that performs multimodal retrieval and indexing, resulting in a text-and-visual index. It also underlines the deployment and output features in cloud or edge environments, encompassing privacy-compliant, on-premises-retrievable results and scoring within the context of latency and data-exposure constraints.

5.5 Governance research

The field of governance is still under research, as incident copilots operate under confidential operational and security information. Provenance tracking ought to affix unchangeable signifiers to each taken in artifact and produced assertion, allowing you to score reliably in citations of that assertion (e.g., 0–1 confidence) based on source freshness, and sibling-source agreement. The 100% audit coverage for ticket alterations caused by the copilot and the necessity to use retention policies should align with regulatory requirements. Time-stamping blockchain methods applied to immutable test records produce tamper-evident incident records, increase accountability for incidents, and enable incident resolution across teams and vendors [25].

6. CONCLUSIONS

The paper has covered the development, implementation, and testing of incident copilots, applications of assistants run by Large Language Models (LLMs) integrated into the Site Reliability Engineering (SRE) and Security Operations (SecOps) paradigm. Relying on structured sets of operational data, generational reference architectures, also based on retrieval-augmented generation (RAG) and statistically defensible forms of analysis, the revelation is made which is that incident copilots, in this context, are potentially the most useful in the vicinities where the human respondents conventionally waste time, namely context gathering, multi-source summarization, shift-off, and drafting incident post-incident reviews. Using a scale so similar to a Sev1 incident and the attendant 50-150 alerts and hundreds of chat messages during the first hour, the sample signal, redundancy of log inspection messages, and aggregation of messages into structured summaries can explicitly speed up time-to-first meaningful context (TTFC) over redundant log inspection. It can streamline the logic of an investigation process. This can be confirmed by documented (quantified) decreases in TTFC (10-25%), median MTTR (5-15% among similar measures of severity), and time to documentation preparation (20-40%), assuming that the effects are properly statistically controlled.

In terms of implementation, the best deployment plan will involve knowledge extraction and RAG support, predetermined tool interdependency, and tough governance policies. The copilots will also need the least privileged access; a human must approve any change to the production software. The two-level integration approach, with observability queries (read-only) and human-approved ticket/communication updates, effectively enables an efficiency-risk-mitigation trade-off. Redaction filters, audit logs, and policy validation engines that serve as guardrails are not optional extensions; unlike structural conditions, they must be satisfied by an enterprise before they can be adopted. The lack of these checks can pose a risk to operational reliability due to hallucinations, direct injection, and automation bias.

Critical is also the discipline of measuring the copilot evaluation. Incident duration data are usually right-skewed and heavy-tailed; this is why median values, interquartile ranges, and percentile indicators (p90/p95) should be given priority over plain means. A/B cross-on-call rotation designs, matched incident cohort designs, or before/after comparisons are necessary to determine the effect of copilots, not because of season or parallel process variation. The vendor productivity claims cannot be regarded as final; they need to be seen as guidelines and clear limits, with pre-registered outcome values, and report 95% confidence intervals in a way that establishes the statements as credible. Continued operational control is supported by the permanent control of the rates of factual errors (less than 5%), rejected propositions, and rework incidents.

The example of SRE and SecOps leadership does not imply that there is a technical problem at the organizational level. The copilots generate a socio-technical change in knowledge sharing amongst the responders, telemetry, and the incident copilots themselves. Governance and control procedures, together with cross-functional controls, should be put in place initially to maintain trust and balance rather than relying too much on outputs. The implementation should be a routine process, rollouts

should be made, and real performance data should be used to recalibrate. In a complex business with a heterogeneous architecture and thousands of services under monitoring, an appraisal and consensus on the architecture are instrumental to ensuring the sustainability of controllable value. Incident copilots are a logical sequence of incident response equipment, but they are not meant to substitute for human knowledge. Using RAG-based grounding, access-sensitive training, quality-based statistical verification, and human-in-the-loop governance, Copilot access can be offered via an LLM that significantly improves the rate and quality of handoffs without negatively impacting overall operational safety. Their utility in the long run will hinge on the regularity of measurement, adequate supervision, and selective internalization of the socio-technical framework for reliability and security activities.

REFERENCES

[1] J. Sehgal, "Enhancing Site Reliability Engineering: Scalable Strategies for Automated Incident Response and System Resilience," *J Artif Intell Mach Learn & Data Sci*, no. 2, p. 4, 2024.

[2] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, … and X. Xie, "A survey on evaluation of large language models," *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 3, pp. 1–45, 2024.

[3] S. K. R. Vanama, "Integrating Site Reliability Engineering (SRE) Principles into Enterprise Architecture for Predictive Resilience," *IJETCSIT*, vol. 4, no. 3, pp. 164–170, Oct. 2023.https://ijetcsit.org/index.php/ijetcsit/article/view/514

[4] N. Makhoul, "Review of data quality indicators and metrics, and suggestions for indicators and metrics for structural health monitoring," *Advances in Bridge Engineering*, vol. 3, no. 1, p. 17, 2022. Available: https://link.springer.com/content/pdf/10.1186/s43251-022-00068-9.pdf

[5] H. Qiao, M. C. Orr, and A. C. Hughes, "Measuring metrics: what diversity indicators are most appropriate for different forms of data bias?," *Ecography*, vol. 2024, no. 9, p. e07042, 2024. Available: https://nsojournals.onlinelibrary.wiley.com/doi/pdf/10.1111/ecog.07042

[6] E. Obuse, E. D. Etim, I. A. Essien, E. Cadet, J. O. Ajayi, E. D. Erigha, and L. A. Babatunde, "AI-powered incident response automation in critical infrastructure protection," *International Journal of Advanced Multidisciplinary Research Studies*, vol. 3, no. 1, pp. 1156–1171, 2023. Available: https://www.multiresearchjournal.com/admin/uploads/archives/archive-1757586786.pdf

[7] K. S. Chadha, "Machine Learning–Augmented ETL Pipelines for Fraud–Resistant Insurance Claims Processing," *International Journal of Data Science and Machine Learning*, vol. 5, no. 01, pp. 410–436, 2025, doi:10.55640/ijdsml-05-01-30.

[8] Z. Hadizadeh, A. Nazari, and M. Mansoorizadeh, "Enhancing suggestion detection in online user reviews through integrated information retrieval and deep learning approaches," *Journal of Web Engineering*, vol. 23, no. 3, pp. 431–463, 2024.

[9] H. Zhang, J. Liu, Z. Zhu, S. Zeng, M. Sheng, T. Yang, … and Y. Wang, "Efficient and effective retrieval of dense-sparse hybrid vectors using graph-based approximate nearest neighbor search," *arXiv preprint* arXiv:2410.20381, 2024. Available: https://arxiv.org/pdf/2410.20381

[10] N. Joshi, "Optimizing real-time etl pipelines using machine learning techniques," Available at SSRN 5054767, 2024. Available: https://papers.ssrn.com/sol3/Delivery.cfm?abstractid=5054767

[11] S. Samala, "Real-time Jira analytics: Integrating JQL with Power BI/Snowflake for predictive agile metrics," *International Journal of Sustainability and Innovation in Engineering*, 2024.https://scipubhouse.com/home/international-journal-of-sustainability-and-innovation-in-engineering-ijsie/content/ijsie-2024/real-time-jira-analytics-integrating-jql-with-power-bi-snowflake-for-predictive-agile-metrics/

[12] L. Bellin, *Monitoring at high scale for very heterogeneous distributed systems*, 2024. Available: https://thesis.unipd.it/retrieve/ac4b6668-c6bb-42c0-a0be-fad721fdd309/Bellin_Leonardo.pdf

[13] Mättas, O. (2024). *Generating Textual Summaries from the Bibliographies Contained in Scientific Literature* (Master's thesis). https://studenttheses.uu.nl/bitstream/handle/20.500.12932/46175/Otto_Mattas-Generating_Textual_Summaries_from_the_Bibliographies_Contained_in_Scientific_Literature.pdf?sequence=1

[14] S. Rangu, "Analyzing the impact of AI-powered call center automation on operational efficiency in healthcare," *JISEM Journal*, 2025.https://www.jisem-journal.com/index.php/journal/article/view/8901

[15] S. K. Vishwakarma, "AI-driven predictive risk modelling for aerospace supply chains," *International Journal of Innovation in Business & Economics and Applied Journal*, 2025.https://www.iibajournal.

org/index.php/iibeaj/article/view/64

[16] S. Calhoun, T. Adami, J. Lorenzon, S. Shihab, J. Bair, E. Wagner, … and M. McCrink, "Open Framework Standards for Combined Aircraft Sensor Network for the State of Ohio to Detect and Track Lower Altitude Aircraft," 2023.

[17] R. Hariharan, "API gateway threat prevention in large-scale applications," SciPubHouse, 2024.https://scipubhouse.com/wp-content/uploads/2025/10/011-API_gateway_threat_prevention_in_large-scale_applications.pdf

[18] H. Sheggam and X. Zhang, "Exploring Security Risks and Mitigation Strategies in AI Code Helpers," in *2024 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, IEEE, Nov. 2024, pp. 1–6.

[19] M. A. Islam, *Application of artificial intelligence and machine learning in security operations center*, Doctoral dissertation, Middle Georgia State University, 2023. https://comp.mga.edu/static/media/doctoralpapers/2023_Islam_0516152253.pdf

[20] K. Lulla, "Cross-border compliance and quality assurance in semiconductor manufacturing," *JES*, 2025.https://journal.esrgroups.org/jes/article/view/9196/6076

[21] R. Hariharan, *API Gateway Threat Prevention in Large-Scale Applications*. (2024) https://scipubhouse.com/wp-content/uploads/2024/10/011-API_gateway_threat_prevention_in_large-scale_applications.pdf

[22] A. C. Jha, "Automated firewall policy generation with reinforcement learning," *IJIOT*, 2025.https://www.academicpublishers.org/journals/index.php/ijiot/article/view/5483

[23] K. S. Chadha, "Machine learning–augmented ETL pipelines for fraud-resistant insurance claims processing," *IJDSML*, 2025.https://www.academicpublishers.org/journals/index.php/ijdsml/article/view/5522/6451

[24] S. Moler-Zapata, R. Grieve, D. Lugo-Palacios, A. Hutchings, R. Silverwood, L. Keele, … and S. O'Neill, "Local instrumental variable methods to address confounding and heterogeneity when using electronic health records: an application to emergency surgery," *Medical Decision Making*, vol. 42, no. 8, pp. 1010–1026, 2022.

[25] V. K. Enugala, "Blockchain timestamping for unalterable concrete test logs," *TAJET*, 2025. https://theamericanjournals.com/index.php/tajet/article/view/6346