LLM-assisted Forensic and Compliance Auditing for Public Sector Organizations

João Santos[1], Rodrigo Correia[2], Jesús Betancourt[2], Tiago Cruz[1], João Henriques[2], Filipe Caldeira[2] and Paulo Simões[1]

[1]University of Coimbra, CISUC, DEI, Portugal

[2]CISeD—Research Centre in Digital Services, Polytechnic Institute of Viseu, Portugal

phdsantos@student.dei.uc.pt
pv23006@alunos.estgv.ipv.pt
pv22987@alunos.estgv.ipv.pt
tjcruz@dei.uc.pt
joaohenriques@estgv.ipv.pt
caldeira@estgv.ipv.pt
psimoes@dei.uc.pt

Abstract: From an IT perspective, modern organizations often exhibit a dichotomy between day-to-day operational practice and formal policy enforcement for security, quality, and regulatory compliance. Reconciling these dimensions remains challenging, particularly in large-scale infrastructures that generate high-volume, heterogeneous, and unstructured data, while relevant policies and standards are typically expressed in natural language and therefore difficult to translate into actionable rules and automated compliance checks. This paper presents FOCUS-PA, a project that develops methods and tools to strengthen digital forensics and continuous auditing for compliance management in the Public Administration (PA) sector. FOCUS-PA delivers a Forensic and Compliance Auditing (FCA) platform tailored to public-sector environments, explicitly accounting for the specific characteristics of administrative information systems, their data sources, workflows, and domain-driven legal and privacy constraints. The platform is designed for the continuous ingestion and analysis of operational data, enabling both ongoing compliance monitoring and the forensic investigation of security incidents. To address these requirements, FOCUS-PA introduces an agentic framework that leverages Large Language Models (LLMs) via a Retrieval-Augmented Generation (RAG) approach. By adopting the Model Context Protocol (MCP), the platform provides a unified mechanism to connect natural-language policy documents with structured operational logs, supporting consistent interpretation, correlation, and evidence-driven auditing. A central contribution of FOCUS-PA is reducing the cost of security policy engineering in compliance-auditing solutions. Today, experts must translate regulatory texts and internal procedures into machine-readable specifications - a process that is slow, expensive, and quickly becomes outdated as policies and regulations evolve. We investigate how large language models (LLMs) can help extract and structure policy specifications from unstructured documents (e.g., procedure manuals) and observed operational data, facilitating their conversion into machine-verifiable rules ("Policy as Code"). This approach aims to accelerate policy onboarding and maintenance and to enable the timely adoption of emerging regulations and standards (e.g., NIS2) by streamlining the translation of high-level requirements into actionable compliance checks. We detail requirements for PA deployments and LLM-assisted policy engineering, describe the platform architecture and implementation, and discuss integration aspects, demonstrating how FOCUS-PA can improve compliance governance and accelerate incident investigation through scalable analytics and expert-supervised automation.

Keywords: Large language models, Log-based anomaly detection, Digital forensics, Regulatory compliance, Model context protocol, Retrieval-augmented generation

1. Introduction

The digital transformation of Public Administration (PA) has resulted in the accumulation of vast amounts of data, bringing with it complex regulatory obligations. Organizations are now subject to regulations such as the GDPR and NIS-2, which have elevated compliance to a core operational requirement. Consequently, non-compliance represents a critical risk, leading to substantial financial penalties and the erosion of institutional reputation (Mandal and Mazumdar, 2018).

Despite this, compliance assessment remains largely manual and infrequent. This approach is inherently slow and unable to scale with the massive volumes of digital footprints generated by modern infrastructures (Buddhadev et al., 2025). While a significant portion of non-compliant behaviour is recorded in system logs, the sheer volume and heterogeneity of these records render manual inspection infeasible. This limitation has motivated the exploration of Large Language Models (LLMs), which allow for the interpretation of semi-structured log data with a level of contextual understanding previously unattainable by automated systems.

The challenge of interpreting these digital traces is shared by both compliance auditing and security forensics. As noted by Henriques et al. (2024), both domains rely on the same evidentiary foundations to identify deviations from defined baselines—whether those baselines are legal regulations or internal security policies. Because a single anomalous log entry can simultaneously signal a regulatory failure and a security threat, there is a clear opportunity for a unified analytical approach.

Building on this premise, we propose FOCUS-PA, an LLM-based framework that treats compliance and forensics as a singular orchestration task. Unlike traditional "passive" RAG systems, FOCUS-PA leverages the Model Context Protocol (MCP) to allow LLMs to autonomously and securely query log streams and policy stores. By using a unified pipeline, the framework equips auditors with evidence of regulatory breaches while providing forensic analysts with a signalling mechanism for suspicious activity.

2. Background and Related Work

2.1 Log-based Anomaly Detection and Forensics

Anomaly detection involves the analysis of data samples to identify deviations from expected operational patterns (Rella, 2019). Given the critical nature of organizational security, a diverse range of methodologies has been explored.

Traditional approaches to this problem often rely on the rule-based solutions (Bhatt, Manadhata and Zomlot, 2014), where audit trails are compared against pre-defined, technology-specific signatures designed to flag suspicious behaviour. While effective for identifying known attack vectors, these systems are inherently rigid; they struggle to detect zero-day exploits and require the laborious manual creation of new rules as the threat landscape evolves.

To address these limitations, other solutions like Du et al. (2017) and Meng et al. (2019) resort to machine learning, specifically using Long Short-Term Memory Neural Networks to, given a string of logs, obtain the probability distribution of the next log to appear. In these frameworks, a log entry is flagged as anomalous if its probability of occurrence falls below a specific threshold. Other approaches, such as Henriques et al. (2022) describe a closed-loop process, from anomaly classification to policy generation, based on Decision Trees which support the generation of Policy-as-Code rules to be enforced by a policy engine.

With the advent of the LLMs, additional approaches were developed (Qi et al., 2023; Liu et al., 2024; Ren et al., 2024). From simply providing example logs together with human knowledge to the employment of more elaborate prompt engineering techniques present in the latter two. Not only do these models present promising research directions regarding anomaly classification, but they also have the great advantage of being able to provide reasonable explanations to why they were flagged as such.

The development of these models has also provided new research opportunities for the forensic investigation field, an important process to analyse security breaches and events. Recent research in cybersecurity forensics have leveraged open-source LLMs to enhance the analysis of security breaches. Research indicates that smaller, optimized models can match or exceed the performance of traditional machine-learning techniques, such as Random Forest and XGBoost, with the added benefit of generating human-readable justifications for their classifications (Palma et al., 2025). Central to several frameworks is Retrieval-Augmented Generation (RAG), which refers to the retrieval of relevant information from external knowledge bases before generating responses with LLMs to ensure factual grounding (Gao et al., 2023). Building on this paradigm, specialized frameworks like GenDFIR have been developed to combine rule-based systems with RAG-driven agents, facilitating the reconstruction of complex incident timelines from diverse data sources (Loumachi, Ghanem and Ferrag, 2025).

2.2 Compliance Auditing

Due to the inherent inefficiencies of manual compliance auditing, significant research has focused on the development of automated frameworks. Early solutions prior to LLMs include Chen et al. (2011) and Mandal et al. (2018). Chen et al. (2011) utilizes graph-based modelling to aggregate log features into community structures, followed by dimensionality reduction and k-nearest neighbour (k-NN) algorithms to isolate anomalous (possibly non-compliant) data points. While Mandal et al. (2018) follow a rule-based architecture in which raw logs are pre-processed and matched against compliance rules derived from machine-readable intermediate security policies, for which the authors propose an XML-based representation.

Exploring the new horizons opened by LLMs—though not yet specifically for log analysis—significant work has addressed document-level compliance auditing (Hassani, 2024; Buddhadev et al., 2025; Sun, Luo and Li, 2025).

Hassani's approach utilizes passage chunking, where each segment is analysed against predefined compliance rules (e.g., GDPR) by an LLM. By receiving only the rules and instructions, the model performs zero-shot learning.

Buddhadev et al. (2025) extend this concept by implementing an agentic compliance framework. Utilizing multiple task-specific models orchestrated via LangChain, the authors perform a comprehensive analysis of uploaded files. A key distinction of their work is the employment of RAG to fetch relevant rules from a knowledge store based on audit requirements.

Sun et al. (2025) also implement RAG but rely on two distinct graph knowledge bases: a Static graph for entities and concepts, and an "Eventic" graph for regulatory information following a custom ontology. Each document is segmented into chunks used to retrieve relevant nodes from the intersection of these graphs, thereby enriching the prompt provided to the model for compliance evaluation.

Beyond document classification, Chin et al. (2025) investigate the use of autonomous agents for password policy compliance on Windows systems. Their framework utilizes a ReAct (Reasoning and Acting) architecture, which enables an LLM to iteratively reason about compliance tasks—specifically checking system adherence to CIS recommendations—and execute diagnostic actions via a shell command interpreter. By bridging the gap between natural language policies and direct system queries, the agent demonstrates significantly higher efficiency than manual auditing. While the use of shell-based tools differs from the remote log auditing focus of this work, it provides a robust example of how agentic models can autonomously select and utilize tools based on prompt-detected needs.

Regarding conversational interfaces, Hillebrand et al. (2024) developed a RAG-enhanced chatbot for regulatory compliance. Their architecture employs a hybrid search strategy to identify relevant documents, combining vector similarity for semantic context with a BM25 algorithm for keyword precision. The retrieved results are re-ranked using Reciprocal Rank Fusion (RRF), and a 2x boosting factor ensures the model prioritizes retrieved knowledge over its internal weights, reducing hallucinations.

Similarly, Aushev et al. (2025) introduced RAGulator, which also utilizes a hybrid retrieval approach. A distinct contribution of RAGulator is the implementation of lightweight BM25 pre-filtering, which shortlists candidate passages before they are processed by more resource-intensive transformer models (such as NV-Embed-v2). This multi-stage retrieval process optimizes computational efficiency without sacrificing accuracy.

2.3 Model Context Protocol

The integration of the Model Context Protocol (MCP) allows a decoupled, interface between the LLM and the heterogeneous data sources. Traditional RAG implementations often suffer from integration debt, where data retrieval logic is hard coded to specific database schemas.

MCP architecture consists of the following elements (Hou et al., 2025):

- MCP Host: The execution environment (e.g., the FOCUS-PA application) that orchestrates the workflow.
- MCP Client: A component within the host that manages secure, persistent connections and capability discovery.
- MCP Server: A specialized service providing Tools (e.g., semantic search), Resources (e.g., raw logs), and Prompts (e.g., audit templates).

2.4 Natural Language to Policy as Code Translation

While previous approaches utilize LLMs primarily as compliance checkers, the works discussed in this section focus on automating the generation of Policy as Code (PaC)—a paradigm that encodes security and compliance policies into machine-readable formats (Romeo et al., 2026).

Research focusing specifically on generating rules for the Open Policy Agent (OPA) includes the distinct works of Romeo et al. (2026), Chowdhary et al. (2025), and Reed et al. (2025). These works target Rego, OPA's declarative query language designed to evaluate policies against structured data.

Romeo et al. (2026) present ARPaCCino, an agentic system that translates natural language policies into PaC rules. The system is equipped with a 'RAG tool' to fetch domain-specific knowledge, such as official OPA documentation and Rego code examples. Additionally, it incorporates a 'Rego rules checker tool' that provides iterative feedback, refining the output until the translation is successful. While ARPaCCino specifically targets

Infrastructure as Code (IaC) validation—placing it slightly outside the primary scope of this project—it effectively demonstrates the potential of LLMs to translate high-level enterprise policies into machine-verifiable code.

Similarly, Chowdhary et al. (2025) propose AutoPaC, a natural language-to-PaC framework. Unlike ARPaCCino's approach, AutoPaC utilizes an LLM specifically fine-tuned on Rego code to perform translations. The system incorporates the LLM-as-a-Judge paradigm—where a secondary model evaluates the generated output and provides corrective feedback. Subsequently, the code is formatted using a Rego linter and validated against test cases if any is provided. This iterative generation-evaluation cycle continues until the judge provides a positive evaluation, ensuring high-fidelity rule production.

Furthermore, Reed et al. (2025) investigate the automation of the policy deployment lifecycle through a structured, three-stage pipeline. The process begins with Semantic Segmentation, using fine-tuned transformer models to perform Named Entity Recognition (NER) on regulatory text. These extracted legal concepts are then mapped to a formal Regulatory Logic Ontology (RLO), which acts as a machine-readable intermediate representation. Finally, this structured data is used to synthesize executable code, targeting languages such as OPA/Rego and HashiCorp Sentinel.

Further contributions to LLM-driven configuration are presented by El Hachimi et al. (2025) and Wang et al. (2025). While these studies do not generate Rego policies, their methodologies are highly relevant. El Hachimi et al. leverage a retry-based correction loop, utilizing execution errors as feedback to refine model outputs autonomously. Meanwhile, Wang et al. focus on cross-platform compatibility, demonstrating how generating an Intermediate Representation (IR) first allows for a flexible, technology-agnostic translation into the final target syntax.

3. The FOCUS-PA Platform

This section details the architectural design and current implementation of the prototype. As illustrated in Figure 1, the infrastructure comprises four primary components. First, the Log Sources continuously ingest and forward audit events to the centralized forensic data store. Second, the Data Store Gateway (MCP Server) acts as a unified gateway, mediating the model's access to both the log and policy databases through tools. Third, an Authorization Server manages the issuance and validation of access tokens, ensuring secure, role-based execution of the available tools. Finally, the core FOCUS-PA platform orchestrates the overall analytical workflow and serves as the primary interface for user interaction.

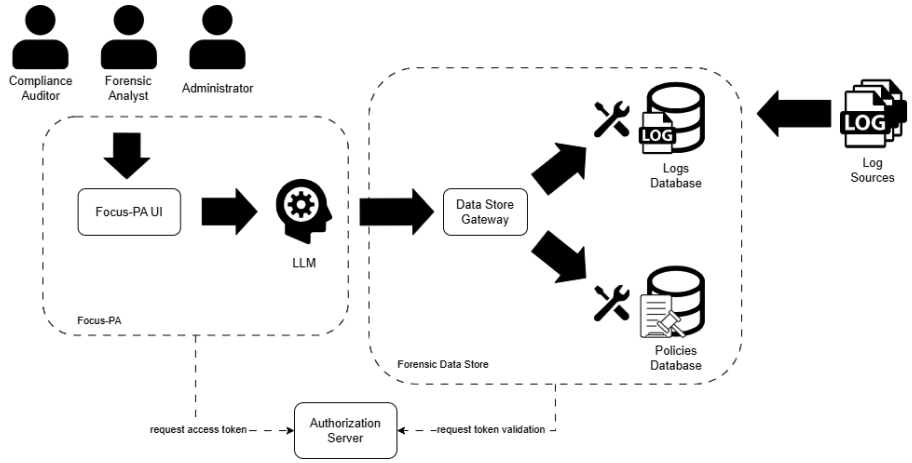

Figure 1: FOCUS-PA architecture

3.1 Data Sources and log Representation

Although on future iterations we aim to achieve a system that analyses a wide range of log types, the current development state only considers Windows Security logs. These logs are serialized in a JSON-like representation and contain the fields stated below:

- Timestamp, host, and process identifiers.
- Event ID and provider (e.g., Microsoft-Windows-Security-Auditing).
- Event-specific fields (e.g., user SID, target process, return code).
- And free-text description supplied by the operating system.

In addition to the logs, our current prototype enriches the prompts with relevant policy passages. These passages are generated using a sliding window chunking heuristic, designed to preserve the structural integrity of the regulatory and internal policy documents. The current implementation utilizes a window size of 800 words, a parameter optimized to capture entire policy clauses (which often exceed 500 words) while staying within the effective attention and context window limitations of the underlying LLM. We also consider an overlap of 80 words (10%), balancing computational storage requirements with the preservation of semantic continuity. This overlap ensures that logical requirements spanning across a chunk boundary are not lost, mitigating the risk of fragmented context during the retrieval phase.

3.2 Preliminary Data Store Gateway Implementation

Both databases are present in a centralized Data Store Gateway. This allows a standardized, secure way to present this data to the LLMs employed.

The current prototype provides two tools to the models, one for each database:

- Log Searching Tool: Receiving a database language specific query as input together with the maximum number of results to retrieve and returns a list of the fetched logs.
- Policy Querying Tool: Similarly to the previous tool, this tool takes a query and the limit of results. This natural language query is then embedded and used for semantic search, obtaining the passages with higher similarity values.

To mitigate syntactic errors in log database query generation, the corresponding tool definition includes explicit formatting constraints and schema hints within the description supplied to the model. This ensures that the model adheres to specific database language requirements—such as the mandatory nesting of temporal filters within range clauses—without requiring a separate fine-tuning step. By embedding these rules directly into the tool's description, the LLM treats the formatting requirements as a functional constraint during the tool-selection phase of the agentic loop.

To improve security of the system we employ the OAuth2.1 security standard. This way, the Data Store Gateway requires a valid, role-scoped access token for every request, effectively binding the LLM's analytical capabilities to the specific privileges of the authenticated user. The detailed operational workflow of a secured tool execution (in this case the log querying tool), from the initial challenge to final data retrieval, is illustrated in Figure 2, following the steps stated bellow:

1. Initial Request: The user provides a natural language prompt requesting information regarding specific log activity.
2. Unauthenticated Tool Call: The model attempts to invoke the required log querying tool. At this stage, no access token is attached to the request.
3. Access Challenge: The Data Store Gateway intercepts the request and, identifying the lack of credentials, blocks access with a 401 Unauthorized response.
4. Token Acquisition: The FOCUS-PA platform detects the challenge and requests an access token from the Authorization Server (AS), asserting the identity and role of the user who initiated the prompt.
5. Credential Issuance: The AS validates the request and grants a scoped access token to the platform.
6. Authenticated Tool Call: The model re-attempts the tool execution, this time including the valid access token in the request header.
7. Validation Request: Upon receiving the authenticated request, the Data Store Gateway contacts the AS to verify the token's integrity.
8. Introspection: The AS performs token validation, confirming that the token is active and has not been revoked.
9. Authorization and Data Retrieval: If the token contains the required role for the specific tool (e.g., Sec. Analyst for log searching), the Data Store Gateway executes the query and provides the requested data to the model.

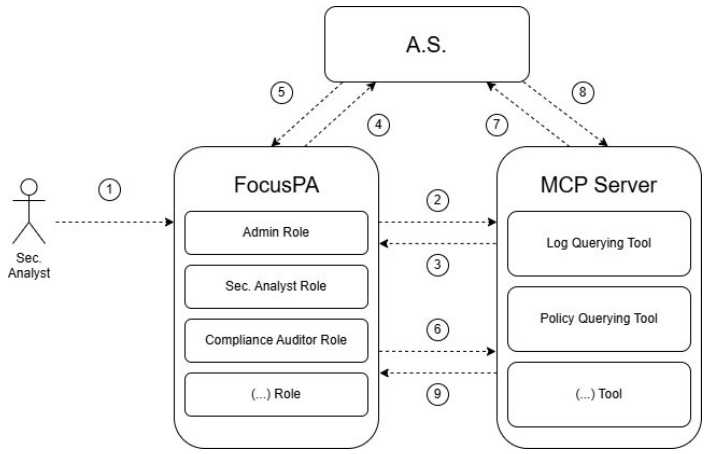

Figure 2: Tool calling workflow example

3.3 RAG Controller

The RAG controller is responsible for communicating with the model and present discovered tools from the Data Store Gateway to it. Ideally, the advantage of using the model to call these tools is for it to tailor each query according to the goals present in the user's prompt.

The interaction with the model first starts by stating the role of the model (White et al., 2023) prompting the model to act as a Windows Event Logs expert. This context is vital for the semantic translation of raw log data (such as mapping numerical Event IDs to their corresponding security actions) before evaluating them against the retrieved policy fragments.

Following the initial persona setup, the user's prompt is presented to the model. The system is then expected to autonomously identify any information gaps required to address the query and invoke the relevant knowledge enrichment tools to retrieve the missing context.

3.4 Metrics Logging

To ensure reproducibility and facilitate diagnostic analysis, FOCUS-PA incorporates a centralized observability layer managed by a dedicated "MetricsLogger". This component records every LLM interaction as a standardized entry in a local CSV repository, creating a persistent audit trail of system behavior across different experimental runs. The captured data is organized into four primary domains:

- Execution Metadata, including ISO8601 timestamps, service identifiers (e.g., chat vs. RAG), and the model configuration utilized for the call.
- Resource Consumption and Throughput. Tracks token and character counts for prompts and completions, alongside wall-clock duration and generation speed (tokens per second/minute).
- Retrieval Statistics, quantifying the volume of log entries from Elasticsearch and policy fragments from ChromaDB effectively injected into the context, providing visibility into the RAG retrieval thresholds.

4. Conclusion

In summary, the FOCUS-PA architecture bridges the semantic gap between low-level system audit data and high-level natural language policies, specifically within the constraints of public-sector environments. The current implementation utilizes an agentic chatbot framework equipped with specialized context-enrichment tools. By correlating these disparate data sources, the system facilitates an active RAG paradigm, allowing compliance auditors and forensic analysts to investigate non-compliant or suspicious activity through intuitive natural language queries. These tools are integrated via the Data Store Gateway, ensuring a secure, standardized interface for database interactions while enforcing role-based access control (RBAC) to limit tool execution based on user privileges within the organization.

Preliminary development observations indicate that optimizing tool-calling performance remains a critical challenge. Future refinements should focus on enhancing prompt engineering strategies, providing more granular tool descriptions, or exploring domain-specific fine-tuning to improve the model's situational awareness during the reasoning step.

Additionally, our team is exploring how LLMs can automate the generation of Rego policies directly from system logs. Building on recent work (e.g., Wang et al., 2025), a promising approach is to first produce an Intermediate Representation (IR): a high-level, stepwise plan that is explicitly grounded in the log schemas being analysed. Following the iterative "rule checker" paradigm (e.g., Romeo et al., 2026), the IR can be refined through review cycles until it is deemed correct—either via a human-in-the-loop expert, or with an LLM-as-a-Judge (e.g., Chowdhary et al., 2025), with final approval always performed by an expert.

Once validated, the IR can be compiled into Rego either in a single pass or incrementally. In the incremental setting, each step can be validated immediately, and any syntax or type errors can be fed back to the model for automated repair, in line with prior self-correction workflows (e.g., El Hachimi et al., 2025). Because LLMs may not consistently adhere to Rego-specific constraints, generation quality can be improved by augmenting the prompt with targeted documentation excerpts, curated examples, and explicit coding rules.

This workflow supports an efficient operational model: the LLM is primarily used during **policy authoring**, while the resulting Rego rules can be executed continuously for monitoring and auditing—reducing repeated model invocations and lowering overall runtime costs.

Acknowledgements

This work was partially funded by the FOCUS-PA project (2024.07695.IACDC/2024), through the FCT—Foundation for Science and Technology under call FCT PRR RE-C05-i08.

Ethics Declaration: No ethical clearance was needed for the research documented in this paper.

AI Declaration: This document was reviewed and refined with the assistance of Large Language Models, such as ChatGPT, or similar tools, which helped check grammar, correct typos, and enhance clarity. The overall content and ideas remain solely the responsibility of the authors.

References

Aushev, I. *et al.* (2025) 'RAGulator: Effective RAG for regulatory question answering', *Proceedings of the 1st Regulatory NLP Workshop (RegNLP 2025)*, pp. 114–120. Available at: https://aclanthology.org/2025.regnlp-1.18/ (Accessed: 5 March 2026).

Bhatt, S., Manadhata, P.K. and Zomlot, L. (2014) 'The operational role of security information and event management systems', *IEEE security & Privacy*, 12(5), pp. 35–41.

Buddhadev, Y. *et al.* (2025) 'Automated Compliance Checking of Unstructured Data Using Large Language Models and Langchain.', *Mathematical Modelling of Engineering Problems*, 12(9). Available at: https://www.iieta.org/journals/mmep/paper/10.18280/mmep.120927 (Accessed: 3 March 2026).

Chen, Y. and Malin, B. (2011) 'Detection of anomalous insiders in collaborative environments via relational analysis of access logs', *Proceedings of the first ACM conference on Data and application security and privacy*. *CODASPY '11: First ACM Conference on Data and Application Security and Privacy*, San Antonio TX USA: ACM, pp. 63–74. Available at: https://doi.org/10.1145/1943513.1943524.

Chin, J.H. *et al.* (2025) 'Automating Security Audit Using Large Language Model based Agent: An Exploration Experiment'. arXiv. Available at: https://doi.org/10.48550/arXiv.2505.10732.

Chowdhary, N. *et al.* (2025) 'Autopac: Exploring llms for automating policy to code conversion in business organizations', *2025 17th International Conference on COMmunication Systems and NETworks (COMSNETS)*. IEEE, pp. 667–675. Available at: https://ieeexplore.ieee.org/abstract/document/10885751/ (Accessed: 5 March 2026).

Du, M. *et al.* (2017) 'DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning', *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. *CCS '17: 2017 ACM SIGSAC Conference on Computer and Communications Security*, Dallas Texas USA: ACM, pp. 1285–1298. Available at: https://doi.org/10.1145/3133956.3134015. (Accessed: 6 March 2026).

El Hachimi, A. *et al.* (2025) 'Flow-Rule Generation for SDN Using LLMs with Retry-Based Deployment Validation', *2025 21st International Conference on Network and Service Management (CNSM)*. IEEE, pp. 1–5. Available at: https://ieeexplore.ieee.org/abstract/document/11297487/ (Accessed: 6 March 2026).

Gao, Y. *et al.* (2023) 'Retrieval-augmented generation for large language models: A survey', *arXiv preprint arXiv:2312.10997*, 2(1), p. 32.

Hassani, S. (2024) 'Enhancing legal compliance and regulation analysis with large language models', *2024 IEEE 32nd International Requirements Engineering Conference (RE)*. IEEE, pp. 507–511. Available at: https://ieeexplore.ieee.org/abstract/document/10628489/ (Accessed: 3 March 2026).

Henriques, J. *et al.* (2022) 'An automated closed-loop framework to enforce security policies from anomaly detection', *Elsevier Computers & Security*, 123. DOI: 10.1016/j.cose.2022.102949

Henriques, J. *et al.* (2024) 'A survey on forensics and compliance auditing for critical infrastructure protection', *IEEE Access*, 12, pp. 2409–2444. DOI: 10.1109/ACCESS.2023.3348552

Hillebrand, L. *et al.* (2024) 'Advancing risk and quality assurance: A rag chatbot for improved regulatory compliance', *2024 IEEE International Conference on Big Data (BigData)*. IEEE, pp. 8668–8670. Available at: https://ieeexplore.ieee.org/abstract/document/10825431/ (Accessed: 5 March 2026).

HOU, X. *et al.* (2025) 'Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions', *arXiv preprint arXiv:2503.23278* [Preprint]. Available at: https://library.qiangtu.com/download/1194/pdf/1194.pdf (Accessed: 3 March 2026).

Liu, Y. *et al.* (2024) 'LogPrompt: Prompt Engineering Towards Zero-Shot and Interpretable Log Analysis', *Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings*. *ICSE-Companion '24: 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings*, Lisbon Portugal: ACM, pp. 364–365. Available at: https://doi.org/10.1145/3639478.3643108.

Loumachi, F.Y., Ghanem, M.C. and Ferrag, M.A. (2025) 'Advancing cyber incident timeline analysis through retrieval-augmented generation and large language models', *Computers*, 14(2), p. 67.

Mandal, D. and Mazumdar, C. (2018) 'Automating information security policy compliance checking', *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*. IEEE, pp. 1–4. Available at: https://ieeexplore.ieee.org/abstract/document/8470420/ (Accessed: 3 March 2026).

Meng, W. *et al.* (2019) 'Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs.', *Ijcai*, pp. 4739–4745. Available at: https://nkcs.iops.ai/wp-content/uploads/2019/06/paper-IJCAI19-LogAnomaly.pdf (Accessed: 3 March 2026).

Palma, G. *et al.* (2025) 'Leveraging large language models for scalable and explainable cybersecurity log analysis', *Journal of Cybersecurity and Privacy*, 5(3), p. 55.

Qi, J. *et al.* (2023) 'Loggpt: Exploring chatgpt for log-based anomaly detection', *2023 IEEE international conference on high performance computing & communications, data science & systems, smart city & dependability in sensor, cloud & big data systems & application (HPCC/DSS/SmartCity/DependSys)*. IEEE, pp. 273–280. Available at: https://ieeexplore.ieee.org/abstract/document/10466820/ (Accessed: 3 March 2026).

Reed, E. *et al.* (2025) 'Leveraging Natural Language Processing to Translate Regulatory Text into Enforceable Policy-as-Code'. Available at: https://www.researchgate.net/profile/Christianah-Ayoade/publication/400081471_Leveraging_Natural_Language_Processing_to_Translate_Regulatory_Text_into_Enforceable_Policy-_as-Code (Accessed: 5 March 2026).

Rella, N. (2019) 'AnomalY detection', *Computer Fraud & Security* [Preprint]. Available at: https://www.researchgate.net/profile/Bhanu-Reddy-Rella/publication/390556413_Anomaly_Detection/links/67f3fce8e8041142a16d2da9/Anomaly-Detection.pdf (Accessed: 4 March 2026).

Ren, H. *et al.* (2024) 'CLogLLM: A Large Language Model Enabled Approach to Cybersecurity Log Anomaly Analysis', *2024 4th International Conference on Electronic Information Engineering and Computer Communication (EIECC)*. IEEE, pp. 963–970. Available at: https://ieeexplore.ieee.org/abstract/document/10929078/ (Accessed: 3 March 2026).

Romeo, F. *et al.* (2026) 'ARPaCCino: An Agentic-RAG for Policy as Code Compliance', in P.K. Chrysanthis et al. (eds) *New Trends in Database and Information Systems*. Cham: Springer Nature Switzerland (Communications in Computer and Information Science), pp. 467–481. Available at: https://doi.org/10.1007/978-3-032-05727-3_39.

Sun, J., Luo, Z. and Li, Y. (2025) 'A compliance checking framework based on retrieval augmented generation', *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 2603–2615. Available at: https://aclanthology.org/2025.coling-main.178/ (Accessed: 3 March 2026).

Wang, H. *et al.* (2025) 'RulePilot: An LLM-Powered Agent for Security Rule Generation'. arXiv. Available at: https://doi.org/10.48550/arXiv.2511.12224.

White, J. *et al.* (2023) 'A prompt pattern catalog to enhance prompt engineering with chatgpt', *arXiv preprint arXiv:2302.11382* [Preprint]. Available at: https://omekas-test.sba.unipi.it/files/original/9473424cea8d562f876a4bca4bedd9e2336910af.pdf (Accessed: 3 March 2026).