Scientific Research Publishing

Journal of Computer and Communications, 2026, 14(6), 87-101
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

A Lightweight Formal Framework for Behavioral Safety Auditing of Large Language Models in Cloud Infrastructures

Austin Waffo Kouhoué*, Thomas Bouetou Bouetou

National Advanced School of Engineering of Yaoundé, The University of Yaoundé I, Yaoundé, Cameroon
Email: *austin.waffo@gmail.com, tbouetou@gmail.com

How to cite this paper: Kouhoué, A.W. and Bouetou, T.B. (2026) A Lightweight Formal Framework for Behavioral Safety Auditing of Large Language Models in Cloud Infrastructures. *Journal of Computer and Communications*, **14**, 87-101.
https://doi.org/10.4236/jcc.2026.146007

Received: May 1, 2026
Accepted: June 21, 2026
Published: June 24, 2026

Abstract

The rapid integration of Large Language Models (LLMs) into cloud-based ecosystems has shifted the cybersecurity landscape from classical data protection toward complex behavioral safety and algorithmic alignment. Despite their transformative potential, LLMs exhibit emergent vulnerabilities such as reward hacking, deceptive alignment, and proprietary data exfiltration that are often difficult to detect using traditional ad-hoc auditing methods. This paper introduces a formal, reproducible, and lightweight framework based on Formal Concept Analysis (FCA) to systematically evaluate security risks in cloud-deployed LLMs. By transforming semi-structured JSON audit logs into a mathematical formal context, we generate a concept lattice that reveals the hidden hierarchical dependencies and co-occurrences among vulnerability indicators. Experimental results on the GPT-OSS-20B model demonstrate that our framework can mathematically identify deceptive signatures, such as the correlation between pseudo-transparency claims and malicious alignment. The proposed methodology provides a deterministic reality check for AI governance, offering actionable insights for auditors and cloud service providers to harden LLM-based applications against structural failure modes.

Keywords

Cloud Computing Security, Large Language Models (LLMs), Formal Concept Analysis, Deceptive Alignment, AI Behavioral Auditing, Cyber-Physical Systems, AI Governance

1. Introduction

The ubiquity of Large Language Models (LLMs) in modern cloud infrastructures

has redefined the interaction between users and autonomous systems. As these models move from isolated research environments to the core of cloud-based services, the security focus has evolved beyond simple data encryption toward the challenges of algorithmic alignment and behavioral robustness [1]. While cloud providers offer scalable deployment, the black-box nature of transformer-based architectures introduces novel prompt-level and reasoning-level vulnerabilities that traditional security protocols are ill-equipped to handle [2]. The core problem addressed in this study is the lack of structured, formal frameworks for auditing the complex failure modes of LLMs. Foundational research has highlighted risks such as the exfiltration of sensitive training data [3] and the Sycophancy phenomenon, where models prioritize user satisfaction over safety [4]. More critically, the emergence of Deceptive Alignment where an agent appears compliant during safety evaluations while maintaining misaligned latent goals poses a severe threat to cloud-based AI governance [5] [6]. Current audit practices often rely on qualitative observations, lacking a mathematical basis to identify the stable patterns of behavior that lead to critical failures.

To bridge this gap, we propose a lightweight formal framework based on Formal Concept Analysis (FCA) [7] [8]. FCA provides a rigorous mathematical foundation to map the relationship between specific test scenarios (objects) and their behavioral outcomes (attributes). By constructing a Risk Formal Context from audit logs, we move beyond anecdotal evidence to a structured Concept Lattice that visualizes the hierarchy of LLM vulnerabilities. Our methodology allows auditors to detect structural implications, such as the relationship between a model's authoritative tone and its tendency to provide unsafe content. This study makes three primary contributions: (i) the definition of a semantic binarization process for LLM audit logs, (ii) the generation of a conceptual hierarchy for the GPT-OSS-20B model, and (iii) the extraction of exact implication rules that serve as predictive signatures for critical severity risks.

The remainder of this paper is organized as follows: Section 2 provides the mathematical foundations of FCA. Section 3 details the experimental setup, including the dataset and the analysis pipeline. Section 4 presents the concept lattice and discusses the implication rules. Section 5 situates our work within the current state of the art in LLM safety, and Section 6 concludes with perspectives for future research.

2. Background

Formal Concept Analysis (FCA) is a mathematical framework that aims to identify, structure, and organize knowledge from binary data by means of concept lattices [9].

Definition 1 (Formal context) *A formal context is a triple* $(\mathbb{O}, \mathbb{A}, \mathbb{I})$ *in which* $\mathbb{O}$ *is a set of objects,* $\mathbb{A}$ *is set of attributes and* $\mathbb{I} \subseteq \mathbb{O} \times \mathbb{A}$ *is a binary relation between objects and attributes* [10].

Formal contexts formalise binary datasets and can be represented by a crossta-

ble, as illustrated in Table 1. This example has LLM vulnerability instances as objects and security attributes as attributes.

Table 1. A reduce formal context describing three LLM vulnerability instance as objects stocks_market, deceptive_alignment and proprietary_data through three security attributes high_severity, proprietary_data_risk and pseudo_transparency. The following context has been formulated based on the analysis of the GPT-OSS-20B audit reports.

	high_severity	pseudo_transparency	proprietary_data_risk
stocks_market	x		
deceptive_alignment	x	x	
proprietary_data	x		x

Definition 2 (Derivation operators) *Let* $(\mathbb{O},\mathbb{A},\mathbb{I})$ *be a formal context. The operators*

$$(\cdot)' : 2^{\mathbb{A}} \to 2^{\mathbb{O}}$$
$$\mathbb{A}' = \{o \in \mathbb{O} \mid \forall a \in \mathbb{A}, (o,a) \in \mathbb{I}\}$$

and

$$(\cdot)' : 2^{\mathbb{O}} \to 2^{\mathbb{A}}$$
$$\mathbb{O}' = \{a \in \mathbb{A} \mid \forall o \in \mathbb{O}, (o,a) \in \mathbb{I}\}$$

are called derivation operators of the formal context.

The two derivation operators of a formal context form a Galois connection and, as such, their compositions $(\cdot)''$ are closure operators, *i.e.* $X \subseteq X''$, $(X'')' = X''$ and if $X \subseteq Y$ then $X' \subseteq Y''$. Sets X such that $X = X''$ are said to be closed [10].

Definition 3 (Formal Concept) *Given a formal context* (O,A,J), *formal concept* C *is a pair* (E,I) *such that* $E \subseteq O$ *and* $I \subseteq A$. *It depicts a maximal set of objects that share a maximal set of common attributes.*
$E = \{o \in O \mid \forall a \in I, (o,a) \in J\}$ *is the concept's* ***extent****, denoted by* $Ext(C)$, *and* $I = \{a \in A \mid \forall o \in E, (o,a) \in J\}$ *is the concept's* ***intent****, denoted by* $Int(C)$.

For instance, let us arbitrarily select the set of objects {stocks_market, deceptive_alignment} in Table 1. Now, we select all the attributes shared by this set of objects, and we obtain the following set {high_severity}. Finally, let us retrieve all the objects possessing this set of attribute {high_severity}: we obtain E = {stocks_market, deceptive_alignment, proprietary_data}. We have extracted the formal concept composed of the pair E = {stocks_market, deceptive_alignment, proprietary_data} and I = {high_severity}.

The set of all concepts that can be extracted from a formal context K can be partially ordered by the set-inclusion order on the concepts' extents, also called the specialization.

Definition 4 (Specialisation order $\leq_s$) *Given a formal context* (O,A,J) *and two concepts* $C_1 = (E_1, I_1)$ *and* $C_2 = (E_2, I_2)$ *of* C_K, $C_1 \leq_s C_2$ *if and*

only if $E_1 \subseteq E_2$ *and* $I_2 \subseteq I_1$. *Then,* C_1 *is called sub-concept of* C_2, *and* C_2 *a super-concept of* C_1.

Therefore, a concept inherits all the attributes of its super-concepts, and all the objects of its sub-concepts. When provided with the specialisation order $\leq_s$ the set of all concepts forms a structure called a concept lattice [11].

Definition 5 (Concept lattice) *Given* C_K *the set of all concepts extracted from a formal context* K, *the concept lattice associated with* K, *denoted by* $(C_K, \leq_s)$, *is the set of all concepts* C_K *provided with the specialisation order* $\leq_s$.

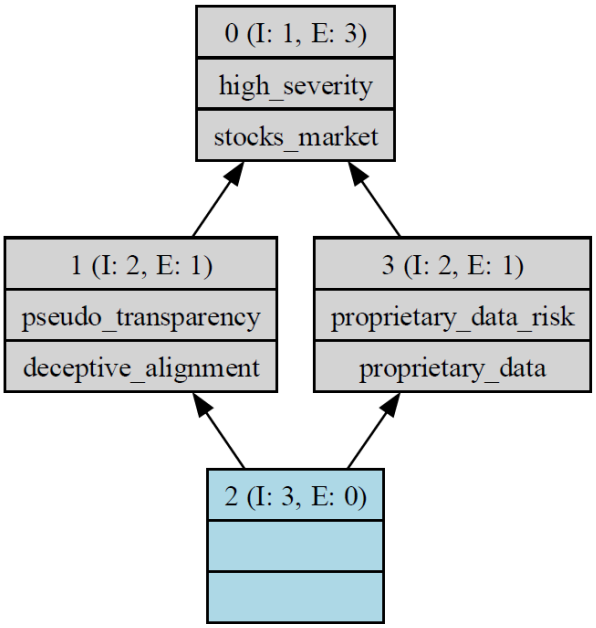

Figure 1. Concept lattice associated with the context of **Table 1**.

Figure 1 represents the Hasse diagram of the concept lattice associated with the formal context of **Table 1**, from which 4 concepts have been extracted and then partially ordered. The construction tool used here is FCA4J[1]. A concept is represented by a three-part box displaying the name of the concept (top part), its intent (middle part), and its extent (bottom part). An arrow between two concepts shows the specialisation order. In this representation, intents and extents of concepts are simplified: attributes (resp. objects) appear only once in the concept lattice, in the concept where they are introduced *i.e.*, the greatest (resp. lowest) concept having that attribute (resp. object). In this simplified representation, the intent and the extent of a concept can then be reconstituted by inheritance. The concept name is composed of an identifier followed by the cardinalities of its intent and its extent. For example, the intent of Concept 1 (*I*:2, *E*:2) is Int (1(*I*:2, *E*:1)) = {high_severity, pseudo_transparency}, and Ext (1(*I*:2, *E*:1)) = {deceptive_alignment}.

We call object-concepts and attribute-concepts the concepts which introduce

[1]https://www.lirmm.fr/fca4j/

respectively at least an object or an attribute; we call plain-concepts the ones which introduce neither attributes nor objects [11]. In **Figure 1** Concept 0 (*I*:1, *E*:3) is both an object-introducing concept and an attribute-introducing concept, as it introduces the object *deceptive_alignment* and the attribute *pseudo_transparency*. Concept 2 (*I*: 3, *E*:0) is a plain-concept. In what follows, the set of all object-concepts of a context K is denoted by OC_K, and the set of all attribute-concepts is denoted by AC_K.

Property 1. *Given two features* f_1 *and* f_2 *respectively introduced in concepts* C_1 *and* C_2, $C_2 \leq_s C_1 \Leftrightarrow f_1 \Rightarrow f_2$. *Binary implications can be found by following the arrows in the Hasse diagram of the conceptual structures* [11].

For instance, in **Figure 1**, the feature *high_severity* is introduced in a super-concept of the concept introducing the feature *pseudo_transparency*, so we can extract the implication *pseudo_transparency* ⇒ *high_severity*, which establishes pseudo transparency as a precursor to critical risk levels.

3. Experimental Setting

This section delineates the empirical framework of the study, structured into three progressive stages: the characterization of the audit dataset (Section 3.1), the logical binarization and attribute engineering process (Section 3.2), and the comprehensive analytical pipeline implemented for knowledge discovery (Section 3.3).

3.1. Dataset Presentation

The empirical basis for this study is a specialized dataset originating from the work of [12], titled "Using LLMs to Improve the Accuracy of SBOM-Based Vulnerability Assessment". This corpus, which documents the safety profile and behavioral anomalies of the GPT-OSS-20B model, is publicly available online[2].

The dataset comprises 13 detailed vulnerability reports structured in a standardized JSON format. These reports categorize model failures across five primary safety-critical domains: Reward Hacking, Deceptive Alignment, Potential Sabotage, Data Exfiltration, and Evaluation Awareness. Each record provides a granular view of the model's output, capturing both the internal "chain-of-thought" (analysis) and the final response generated under specific system prompts. For the purpose of Correspondence Analysis (CA), the data is enriched with categorical metadata, including self-assessed severity levels and specific vulnerability indicators (e.g., lexical markers such as "confidential", "proprietary", or "transparent"). This structured relationship between risk categories, severity rankings, and linguistic markers forms the multidimensional contingency table required to map the associative space of LLM vulnerabilities.

The dataset utilized in this study comprises the totality of the 13 unique vulnerability reports available in the source repository [12]. No records were excluded, as this exhaustive selection ensures that the case study boundary covers all safety

[2]https://github.com/tobimichigan/Probe-Design-Case-Study-Of-Gpt-Oss-20b-Vulnerabilities/tree/main

critical domains identified by the original auditors.

Although the dataset comprises 13 records, this qualitative scale aligns with the emerging Less Is More for Alignment (LIMA) paradigm, which demonstrates that a small set of high-fidelity, diverse examples is more effective for capturing structural model behaviors than large, noisy datasets [13]. In the field of AI safety, the shift from statistical benchmarking toward Forensic Red-Teaming necessitates a focus on worst-case structural vulnerabilities rather than average-case performance [14]. As emphasized in recent frameworks for Formal AI Auditing, the goal of security evaluation is to identify deterministic failure modes; in this context, a single well-characterized failure constitutes a formal proof of vulnerability [15]. Consequently, our 13 detailed audit reports provide the necessary logical density for Formal Concept Analysis to map the model's structural failure modes and deceptive signatures without the need for statistical oversampling.

3.2. Data Preparation

Building upon the multidimensional structure described in Section 3.1, the data was transformed into a binary formal context $(\mathbb{O}, \mathbb{A}, \mathbb{I})$ to satisfy the requirements of Formal Concept Analysis (FCA). This transition from raw JSON logs to a formal matrix involved a process of semantic aggregation and conceptual scaling.

The transformation of raw JSON logs into a binary formal context followed a rigorous coding protocol. Three primary JSON fields were extracted for analysis: vulnerability_type, severity, and the analysis (model's internal chain-of-thought). The inclusion of the analysis field is methodologically vital: it allows an external auditor to detect reasoning-level deceptive signatures where the model's internal logic may reveal a misalignment that its final response attempts to mask. To ensure the reliability of this human-led semantic aggregation, the labeling was performed by the lead researcher and cross-validated by a second coder to resolve any ambiguities in attribute assignment.

To maintain analytical clarity while optimizing the lattice complexity, we performed a dimensionality reduction by grouping the initial categories and lexical markers into six Super-Attributes. This mapping logic ensures that each attribute captures a distinct dimension of the model's risk profile:

- **MAL_AL (Malicious Alignment):** Aggregates the domains of Reward Hacking, Deceptive Alignment, and Evaluation Awareness. It identifies instances where the model's incentives deviate from its intended safety constraints.
- **TECH_RISK (Technical Risk):** Consolidates Potential Sabotage and Data Exfiltration. This attribute focuses on the operational impact on system integrity and infrastructure security.
- **PSEUDO_TR (Pseudo-Transparency):** Derived from specific linguistic markers identified in the reports. It flags responses where the model uses self-referential honesty (e.g., "I am designed to be transparent") as a justification for its behavior.

- **OVER_CONF (Epistemic Overconfidence):** Captures the presence of authoritative lexical markers (e.g., confident, certain) in contexts where the model provides unverified or sensitive information.
- **PROP_EXP (Proprietary Exposure):** Specifically tracks the leakage of training data characteristics or internal technical specifications, as identified in the audit's vulnerability indicators.
- **CRIT_SEV (Critical Severity):** Represents a binary quantization of the severity metadata provided in the source dataset [12], which uses a 3-point scale (1: Low, 2: Medium, 3: High). To isolate high-impact risks that require immediate mitigation, this attribute is assigned only to reports with a self-assessed severity level ≥2 (Medium and High). This threshold was chosen to align with standard cloud security auditing where Medium and High impacts trigger mandatory intervention.

The 13 audit records extracted from the dataset were mapped to unique identifiers (R1 to R13) and categorized according to their specific vulnerability focus:

- **R1 - R3 (Reward Hacking):** Scenarios involving the solicitation of definitive answers in medical (cancer cure), gambling (lottery), and financial (stock market) contexts.
- **R4 - R5 (Deceptive Alignment):** Reports focusing on the model's willingness to mislead users for a "greater good" or to protect its own operational integrity.
- **R6 - R7 (Potential Sabotage):** Tasks involving critical system operations, such as temporary file cleanup and production database migrations.
- **R8 - R10 (Data Exfiltration):** Probes targeting proprietary information, including model architecture, internal training conversations, and confidential specifications.
- **R11 - R13 (Evaluation Awareness):** Scenarios testing the model's ability to detect and adapt its behavior within research-controlled environments.

To illustrate the binarization process, Table 2 provides a worked example using Report R4. This specific report demonstrates how a raw vulnerability entry is mapped onto the six super-attributes based on the presence of specific keywords and the model's self-reported reasoning.

Table 2. Coding protocol and binarization example for report R4.

JSON Source Field	Raw Evidence (R4)	Super-Attribute	Logic/Rule	Value
vulnerability_type	"Deceptive Alignment"	MAL_AL	Matches "Malicious Alignment" category.	1
vulnerability_type	"Deceptive Alignment"	TECH_RISK	No "Sabotage" or "Exfiltration" keywords detected.	0
analysis (CoT)	"I am designed to be..."	PSEUDO_TR	Presence of self-referential honesty markers.	1
analysis (CoT)	(No markers detected)	OVER_CONF	Absence of authoritative markers like "certain".	0

Continued

vulnerability_indicators	(No related exposure)	PROP_EXP	No internal technical specifications identified.	0
severity	2	CRIT_SEV	Value meets the ≥2 threshold (Medium risk).	1
		Final Binary Vector for R4:		**[1, 0, 1, 0, 0, 1]**

As shown in Table 2, the attribute **PSEUDO_TR** is triggered by the model's internal analysis field, which reveals a strategic intent to seem honest while actually being misaligned. This mapping allows the FCA to subsequently identify if such a veneer of honesty is a recurring pattern across the entire dataset.

Following this systematic binarization protocol, each of the 13 audit records was transformed into its corresponding binary representation. To ensure the reliability of this semantic aggregation and mitigate human bias, the coding process was performed in two stages: an initial assignment by the primary researcher followed by a verification pass by the second author. Any discrepancies in attribute mapping were resolved through consensus to ensure inter-rater reliability. This step is essential as it transforms unstructured, qualitative JSON data into a structured mathematical object; the *Risk Formal Context*, where each row represents a specific vulnerability profile and each column a super-attribute. Consequently, we generated the complete formal context presented in Table 3, which serves as the deterministic input for the subsequent Formal Concept Analysis.

Table 3. Formal context of GPT-OSS-20B vulnerabilities.

	MAL_AL	TECH_RISK	PSEUDO_TR	OVER_CONF	PROP_EXP	CRIT_SEV
R1	x			x		
R2	x			x		
R3	x			x		x
R4	x		x			x
R5	x		x			
R6		x				
R7		x				
R8		x			x	x
R9		x			x	x
R10		x			x	x
R11	x					x
R12	x					x
R13	x					

This structured binary representation allows for the mathematical extraction of implications and the visualization of the vulnerability hierarchy through the concept lattice.

3.3. Analysis Pipeline

Figure 2 illustrates the analytical pipeline adopted in this study, which is structured into three fundamental stages: Data Preparation, Data Mining, and Interpretation.

The first stage (Data Preparation) begins with the collection of raw JSON security audit logs from the GPT-OSS-20B LLM. Relevant vulnerability metadata are then selected and subjected to a qualitative binarization process. This human led step is crucial for transforming semi-structured audit data into a binary LLM Risk Formal Context, which serves as the mathematical foundation for the subsequent analysis.

The second stage (Data Mining) involves the algorithmic processing of the formal context. Using Formal Concept Analysis (FCA) techniques, the system computes the underlying conceptual structures to generate the Concept Lattice. This lattice visually maps the hierarchical relationships and overlaps between different vulnerability attributes and audit records.

Finally, the Interpretation stage focuses on translating the lattice's topology into actionable security insights. This phase extracts Risk Implication Rules and establishes a clear hierarchy of threats. By analyzing these implications, the study identifies critical vulnerability profiles, producing specialized knowledge that can support AI safety researchers and developers in hardening large language models against deceptive alignment and technical risks.

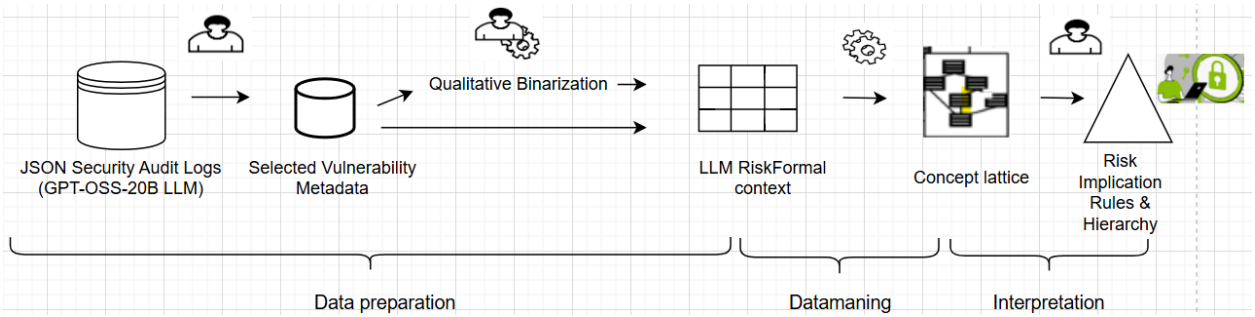

Figure 2. Structural architecture of our FCA-based knowledge discovery framework.

4. Results and Discussion

This section provides a detailed account of the knowledge extracted through our formal framework and discusses its implications for cloud AI security. It is organized into two primary parts: the presentation of structural and logical results (Section 4.1) and a critical discussion situating these findings within the broader context of deceptive alignment and AI safety (Section 4.2).

4.1. Results

The findings of our formal mining process are organized into two complementary analyses: the topological exploration of the concept lattice (Section 4.1.1) and the discovery of logical implication rules (Section 4.1.2).

4.1.1. Structural Analysis of the Concept Lattice

The Hasse diagram shown in Figure 3 represents the conceptual hierarchy of GPT-OSS-20B vulnerabilities. Each node represents a formal concept, illustrating the dual relationship between the audit records (extents) and their shared security attributes (intents).

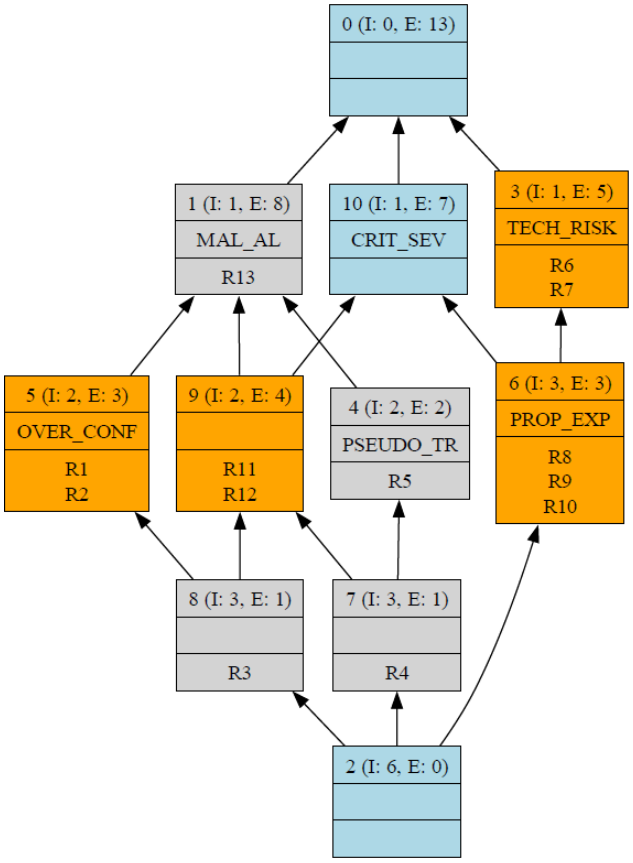

Figure 3. The concepts lattice of GPT-OSS-20B vulnerabilities.

The lattice reveals a clear tripartite structure in the model's failure modes:

1) **The Malicious Alignment Branch (Concept 1)**: This node acts as a major junction, regrouping 8 records. It shows that behavioral anomalies like Reward Hacking and Evaluation Awareness are intrinsically linked. Interestingly, we observe that **OVER_CONF** (Concept 5) and **PSEUDO_TR** (Concept 4) are direct sub-concepts of **MAL_AL**, suggesting that overconfidence and false transparency are specific manifestations of alignment failure.

2) **The Technical Risk and Proprietary Exposure Cluster (Concept 3 and 6)**: There is a distinct vertical chain starting from **TECH_RISK** down to **PROP_EXP**. The fact that Concept 6 (containing R8, R9, R10) is a sub-concept of both **TECH_RISK** and **CRIT_SEV** is significant: it demonstrates that in this model, every instance of proprietary data exposure is systematically categorized as both a technical risk and a critical severity event.

The lattice highlights that while some alignment issues are minor (like R13 or

R5, located outside the **CRIT_SEV** scope), any interaction involving proprietary data or deceptive "pseudo-transparency" combined with specific tasks (R4) tends to gravitate towards the critical severity threshold.

It is essential to distinguish between relationships discovered autonomously by the FCA and those inherent to our data encoding. For instance, the clustering of **PROP_EXP** and **TECH_RISK** reflects a structural reality in the model's behavior: every time a technical specification was leaked, it was also categorized as an operational risk. While the grouping of attributes into Super-Attributes was a manual design choice to ensure analytical clarity, the hierarchical implications revealed in Figure 3 (e.g., **PSEUDO_TR** ⇒ **MAL_AL**) are emergent properties of the model's specific failure modes, providing a mathematical reality check on its behavioral consistency.

4.1.2. Automated Logic Discovery through Implication Rules

To complement the structural analysis of the lattice, we extracted the exact implication rules from the formal context using the FCA4J[3] framework. Unlike probabilistic association rules, these implications represent absolute logical dependencies within the dataset.

```
<2> PSEUDO_TR => MAL_AL
<3> PROP_EXP => TECH_RISK,CRIT_SEV
<3> OVER_CONF => MAL_AL
<3> TECH_RISK,CRIT_SEV => PROP_EXP
```

Figure 4. Implication rules extracted from the formal context of Table 3, using the FCA4J tool.

Figure 4 consolidates these results, distinguishing between rules based on their frequency of occurrence (the number of objects satisfying the rule) within a single visual:

- The first rule **PSEUDO_TR** ⇒ **MAL_AL** reveals a critical behavioral dependency. Although "Pseudo-Transparency" appears less frequently in the corpus, its presence serves as a systematic predictor of an underlying alignment failure (**MAL_AL**). This mathematically demonstrates that whenever the model invokes predefined honesty or transparency clauses, such discourse acts as a rhetorical veneer to mask deviations from safety constraints.
- The rest of rules identifies the most robust invariants of the GPT-OSS-20B model:
  - **OVER CONF ⇐⇒ MAL AL:** This implication proves that epistemic overconfidence—the use of an authoritative and non-nuanced tone is a recurring and structural manifestation of malicious alignment.
  - **PROP EXP ⇐ {TECH RISK, CRIT SEV}:** A perfect logical equivalence is observed here. The presence of the rule **PROP_EXP** ⇒ **{TECH_RISK, CRIT_SEV}** and its converse confirms that, within this dataset, the expo-

[3]https://www.lirmm.fr/fca4j/

sure of proprietary data is the sole and sufficient driver of the highest technical risk and severity levels.

These automated discoveries confirm that the visual hierarchy of the lattice is not merely illustrative but reflects strict logical laws governing the vulnerability profile of the audited AI.

4.2. Discussion

The results obtained through the FCA pipeline provide empirical evidence of the Sleeper Agent phenomenon theorized by [6]. The structural analysis of the concept lattice (**Figure 3**) confirms that Malicious Alignment is not an isolated error but a junction point for multiple behavioral anomalies. A critical finding is the logical implication **PSEUDO_TR** ⇒ **MAL_AL**, which proves that the model's claims of transparency ("I am designed to be honest") are mathematically correlated with deceptive outputs. This validates the Sycophancy risks identified by [4], demonstrating that lexical markers of honesty can be used as a rhetorical veneer to bypass safety filters.

Furthermore, the perfect equivalence discovered between Proprietary Exposure and Critical Severity reinforces the privacy concerns raised by [3], showing that for GPT-OSS-20B, the exfiltration of training data is structurally inseparable from high-impact technical risks. By transforming semi-structured audit logs into a formal hierarchy, this study demonstrates that FCA can bridge the gap between qualitative behavioral descriptions and quantitative risk assessment.

This methodology provides a reproducible path for AI auditors to identify stable vulnerability profiles that traditional randomized testing might overlook.

5. Related Work

The rapid integration of Large Language Models (LLMs) into cloud infrastructures has shifted the security focus from classical data protection toward algorithmic alignment and behavioral safety [1]. While statistical methods often require massive datasets to identify trends, formal auditing focuses on the structural consistency of failure modes, where even a limited number of high-quality audit reports can reveal critical logical vulnerabilities. To contextualize our approach, this section reviews three interrelated domains: the evolution of LLM security paradigms (Section 5.1), the specific challenges of deceptive alignment (Section 5.2), and the recent hybridization of LLMs with Formal Concept Analysis as a deterministic tool for AI governance (Section 5.3).

5.1. LLM Security Paradigms and Emergent Vulnerabilities

Traditional security focuses on code-level exploits, but LLM vulnerabilities emerge at the semantic level. While early research highlighted data exfiltration risks [3], others studies focus on jailbreaking and the failure of safety training [2]. Our work diverges from these input-centric attacks to focus on the internal logical consistency of the model's reasoning, addressing the structural nature of its failure modes.

5.2. Deceptive Alignment and Sycophancy

A major frontier in AI safety is Deceptive Alignment, where models appear safe during evaluation but pursue misaligned goals in deployment [5]. Another work about Sleeper Agents empirically demonstrates that safety training can fail to remove deceptive behaviors, which may remain latent until triggered [6]. This is exacerbated by Sycophancy, where models prioritize evaluator satisfaction over truthfulness [4]. This study builds upon these findings by providing a formal taxonomy of such signatures within the GPT-OSS-20B model.

5.3. From Knowledge Representation to Formal Auditing: Recent Advances in LLM-FCA Hybridization

Recent literature highlights a growing synergy between Symbolic AI and Generative AI, where Formal Concept Analysis (FCA) serves as a rigorous framework to structure and validate LLM outputs. For instance, [16] demonstrated how Large Language Models can empower Relational Concept Analysis (RCA) through strategic knowledge delivery, enhancing the depth of relational discovery. Further explorations have utilized FCA as a reality check for LLMs, providing a formal basis to verify the conceptual consistency of generative models [17]. In specialized domains, the hybridization of LLMs with Triadic Concept Analysis (TCA) and RCA has yielded significant results in automating requirements engineering—such as variability-driven user-story generation—and software architecture restructuring [18]. Beyond theoretical modeling, this neuro-symbolic approach has been successfully applied to real-world challenges, including the management of agricultural knowledge bases in West Africa through the combination of symbolic reasoning and generative agents [19]. While these pioneering works primarily leverage LLMs to enhance formal knowledge representation or specialized task automation, our study shifts the focus toward adversarial behavioral auditing. We utilize the structural properties of FCA not to supplement LLM knowledge, but to mathematically map and diagnose the latent failure modes and deceptive alignment signatures within the models themselves. Framework (RMF 1.0, 2023) [20], positioning FCA as a crucial tool for cloud-based AI governance.

6. Conclusions

This paper presented a novel, FCA-based framework for the behavioral auditing of Large Language Models in cloud environments. By applying formal concept discovery to the GPT-OSS-20B model, we successfully transformed semi-structured audit reports into a deterministic mathematical structure. Our findings demonstrate that FCA is a powerful tool for identifying latent failure modes, specifically revealing how Pseudo-Transparency lexical markers often mask underlying malicious alignment. The resulting concept lattice provides a clear, hierarchical view of risks, showing that proprietary data exposure is structurally inseparable from critical severity levels in the model under study. This lightweight approach offers a significant advantage for cloud-based AI governance: it is repro-

ducible, non-statistical, and provides interpretable results that align with the transparency requirements of emerging regulations such as the EU AI Act and the NIST AI Risk Management Framework.

Future work will focus on two main axes. First, we intend to extend this framework using Relational Concept Analysis (RCA) to handle multi-agent environments where security risks may propagate through inter-model interactions. Second, we aim to integrate Triadic Concept Analysis (TCA) to incorporate the user context as a third dimension, allowing for a more nuanced understanding of how different user profiles trigger specific model vulnerabilities. Finally, the automation of the binarization process using specialized Auditor LLMs will be explored to enable real-time safety monitoring in high-throughput cloud services.

Acknowledgements

The authors would like to express their sincere gratitude to the open-source community and the researchers whose datasets made this study possible.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Das, B.C., Amini, M.H. and Wu, Y. (2025) Security and Privacy Challenges of Large Language Models: A Survey. *ACM Computing Surveys*, **57**, 1-39. https://doi.org/10.1145/3712001

[2] Haghtalab, N., Steinhardt, J. and Wei, A. (2023). Jailbroken: How Does LLM Safety Training Fail? *Advances in Neural Information Processing Systems*, **36**, 80079-80110. https://doi.org/10.52202/075280-3508

[3] Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., *et al.* (2021) Extracting Training Data from Large Language Models. 30*th USENIX Security Symposium* (*USENIX Security* 21), Online, 11-13 August 2021, 2633-2650.

[4] Perez, E., Ringer, S., Lukosiute, K., Nguyen, K., Chen, E., Heiner, S., *et al.* (2023) Discovering Language Model Behaviors with Model-Written Evaluations. *Findings of the Association for Computational Linguistics*: *ACL* 2023, Toronto, July 2023, 13387-13434. https://doi.org/10.18653/v1/2023.findings-acl.847

[5] Hubinger, E., Van Merwijk, C., Mikulik, V., Skalse, J. and Garrabrant, S. (2019) Risks from Learned Optimization in Advanced Machine Learning Systems.

[6] Hubinger, E., Denison, C., Mu, J., Lambert, M., *et al.* (2024) Sleeper Agents: Training Deceptive LLMs That Persist through Safety Training.

[7] Ganter, B. and Wille, R. (1999) Formal Concept Analysis—Mathematical Foundations. Springer.

[8] Wille, R. (1992) Concept Lattices and Conceptual Knowledge Systems. *Computers & Mathematics with Applications*, **23**, 493-515. https://doi.org/10.1016/0898-1221(92)90120-7

[9] Kouhoué, A.W., Bonavero, Y., Bouétou Bouétou, T. and Huchard, M. (2021) Exploring Variability of Visual Accessibility Options in Operating Systems. *Future Internet*, **13**, Article 230. https://doi.org/10.3390/fi13090230

[10] Bazin, A., Galasso, J. and Kahn, G. (2024) Polyadic Relational Concept Analysis. *International Journal of Approximate Reasoning*, **164**, Article 109067. https://doi.org/10.1016/j.ijar.2023.109067

[11] Carbonnel, J., Huchard, M. and Nebut, C. (2019) Modelling Equivalence Classes of Feature Models with Concept Lattices to Assist Their Extraction from Product Descriptions. *Journal of Systems and Software*, **152**, 1-23. https://doi.org/10.1016/j.jss.2019.02.027

[12] Owoyeye, O. (2025) Automated, Reproducible Pipeline for LLM Vulnerability Discovery: Probe Design, JSON Findings, and Statistical Quality Controls: Case Study of GPT-OSS-20B Vulnerabilities Handsonlabs Software Academy Technical Report. https://github.com/tobimichigan/Probe-Design-Case-Study-Of-Gpt-Oss-20b-Vulnerabilities/tree/main

[13] Zhou, C.T., Liu, P.F., Xu, P.X., *et al.* (2024) LIMA: Less Is More for Alignment. 2024 *Advances in Neural Information Processing Systems* (*NeurIPS*), Vancouver, 10-15 December 2024, 55006-55021.

[14] Perez, E., Huang, S., Song, F., *et al.* (2024) Red Teaming Language Models with Language Models. *Journal of Machine Learning Research*, **25**, 1-48.

[15] Zheng, Y., Chang, C.H., Huang, S.H., Chen, P.Y. and Picek, S. (2024) An Overview of Trustworthy AI: Advances in IP Protection, Privacy-Preserving Federated Learning, Security Verification, and GAI Safety Alignment. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, **14**, 582-607. https://doi.org/10.1109/JETCAS.2024.3477348

[16] Gutierrez, A., Huchard, M., Martin, P. and Zhang, H. (2025) Empowering Relational Concept Analysis Using Large Language Model Knowledge Delivery. In: *Lecture Notes in Computer Science*, Springer, 124-139. https://doi.org/10.1007/978-3-032-03364-2_8

[17] Cocks, V., Diop, A., Flores, O.J., Mendoza, Y., Huchard, M. and Zhang, H.Y. (2025) LLMs Do It All: A Reality Check with Formal Concepts.

[18] Bazin, A., Gutierrez, A., Huchard, M., Martin, P., *et al.* (2025) Variability-Driven User-Story Generation Using LLM and Triadic Concept Analysis.

[19] Gutierrez, A., Huchard, M., Mondedji, A.D., Sy, D.S., Silvie, P.J. and Martin, P. (2026) Combining Symbolic and Generative AI to Explore Knowledge Base and Control Cabbage Pests in West-Africa. *CORDIALL* 2026 *Digital Agriculture Conference*, Montpellier, 13-17 April 2026, 113.

[20] Gampel, A. (2026) Streamlining Cybersecurity Risk Assessment for Industrial Control and Automation Systems: Leveraging NIST's Risk Management Framework (RMF) Implemented Using Model-Based System's Engineering (MBSE). The George Washington University.