LLM-Driven Observability for Private 5G Networks: A Modular Platform for Industrial Environments

Maria C. Z. Patricio, Luis Kilmer, Vitor Z. Pamplona, Rilbert L. da Silva, Michel C. Dias and Ruan D. Gomes

[1]IFPB Innovation Hub, Federal Institute of Paraíba (IFPB), João Pessoa, PB, Brazil
P.O. Box 58013-240 – João Pessoa, PB – Brazil

{maria.patricio,luis.kilmer}@academico.ifpb.edu.br,

{vitor.pamplona, rilbert.lima}@polodeinovacao.ifpb.edu.br,

{michel.dias, ruan.gomes}@ifpb.edu.br

Abstract. *Private 5G networks in industrial environments demand low-latency, trustworthy observability of standardized Key Performance Indicators (KPIs), such as those defined in ETSI TS 128 554. However, existing literature rarely addresses the unified integration of on-premises telemetry pipelines with natural-language analytics and the systematic validation of LLM-generated SQL against complex observability schemas. This paper proposes an on-premises observability architecture that orchestrates the ingestion of ETSI/3GPP-aligned KPIs from an Open5GS/OpenAirInterface testbed into a time-series database via a scalable Prometheus–Telegraf–Kafka pipeline. Furthermore, we introduce the Onion Validation framework, a multi-layer inference protocol that enforces intent classification, schema grounding, syntactic correctness, execution-plan conformance, and result-level verification with full auditability. An experimental evaluation was conducted using five 8-bit quantized Large Language Models (2B–8B parameters) processed against 130 domain-specific queries, comparing unified versus partitioned execution architectures. The empirical results reveal a maximum functional correctness of 27.7% in Answerable queries, highlighting the strictness of the validation protocol. The findings demonstrate that while current edge-deployed quantized models exhibit reasoning limitations, the proposed layered validation mechanism is essential to mitigate operational risks, thereby delineating the critical trade-off between on-premises data sovereignty and the semantic accuracy of automated analytics.*

1. Introduction

Industry 4.0 is transforming manufacturing through the integration of cyber-physical systems and ubiquitous connectivity, imposing stringent requirements on communication infrastructures in terms of reliability, latency, and scalability [Yaqub and Alsabban 2023]. In this context, private 5G networks have emerged as a key enabler, supporting Ultra-Reliable Low-Latency Communications (URLLC), Massive Machine Type Communications (mMTC), and enhanced Mobile Broadband (eMBB), while allowing industrial operators to tightly control spectrum usage and implement robust security mechanisms. Furthermore, the functional disaggregation and open interfaces in O-RAN enable programmable control through RAN Intelligent Controller (RIC) xApps and

rApps [Azariah et al. 2024]. However, this increased openness significantly increases architectural complexity, making end-to-end observability a critical requirement for private 5G deployments. In this work, observability is defined more broadly than conventional KPI monitoring, as the integrated ability to collect, contextualize, query, and interpret multi-layer telemetry to support interactive diagnosis and traceable decision-making.

The operation of software-defined and disaggregated 5G networks relies on the continuous monitoring of standardized Key Performance Indicators (KPIs), including throughput, Block Error Rate (BLER), Received power from reference signal (RSRP), and slice-level QoS metrics defined by ETSI and 3GPP standards [ETSI 2021]. To reduce the operational burden of querying and interpreting large volumes of telemetry, natural-language (NL) interfaces for network observability have gained attention. In this setting, Large Language Models (LLMs) are employed to translate natural language requests into Structured Query Language (SQL), allowing operators to reason directly over observability databases [Araujo et al. 2024, Mani et al. 2023, Husom et al. 2025].

Despite their potential, LLM-based text-to-SQL systems remain fragile in operational settings. They often generate syntactically valid but semantically incorrect SQL queries, which may silently produce misleading results and pose concrete risks to network efficiency. This issue is more pronounced when querying complex industrial schemas than when evaluating against curated benchmark datasets [Nascimento et al. 2025]. Furthermore, evaluation approaches based on a single correctness metric fail to capture execution failures, semantic mismatches, and intent misinterpretations, which are critical dimensions for trustworthy performance.

Although the recent literature has explored telemetry collection and NL interfaces in isolation, few studies have systematically validated LLM-generated analytical queries on experimental testbed telemetry. In this work, we address this gap by proposing the on-premises observability platform illustrated in Figure 1. Based on Open5GS and OpenAirInterface, this architecture introduces the Onion Validation framework, a staged pipeline that integrates intent screening, schema grounding, SQL validation, execution-plan conformance analysis, and row-level result verification to ensure comprehensive auditability.

To analyze the effects of inference orchestration, five quantized LLMs (2B–8B parameters) are evaluated using 130 operator-inspired queries under partitioned and unified execution architectures. Functional correctness is assessed by comparing LLM-generated SQL results with hand-written ground-truth queries, while intent routing behavior and end-to-end latency are measured across the complete inference pipeline.

The main contributions of this paper are: (i) the design and implementation of an on-premises observability platform for private 5G, integrating ETSI/3GPP-compliant telemetry ingestion with locally deployed natural-language analytics; (ii) the proposal of the Onion Validation methodology, a multi-layer validation framework for LLM-generated SQL that combines intent screening, schema grounding, syntactic verification, execution-plan analysis, and result-level validation; (iii) an empirical study using 130 operator-inspired queries executed on an observability database built from a real testbed; and (iv) a comparative analysis of five 8-bit quantized LLMs under unified and partitioned schema configurations, evaluating functional correctness, intent classification, and end-to-end latency.

The remainder of this paper is organized as follows. Section 2 reviews related work; Section 3 presents the proposed platform; Section 4 details the Onion Validation methodology; Section 5 discusses the experimental results; and Section 6 concludes the paper.

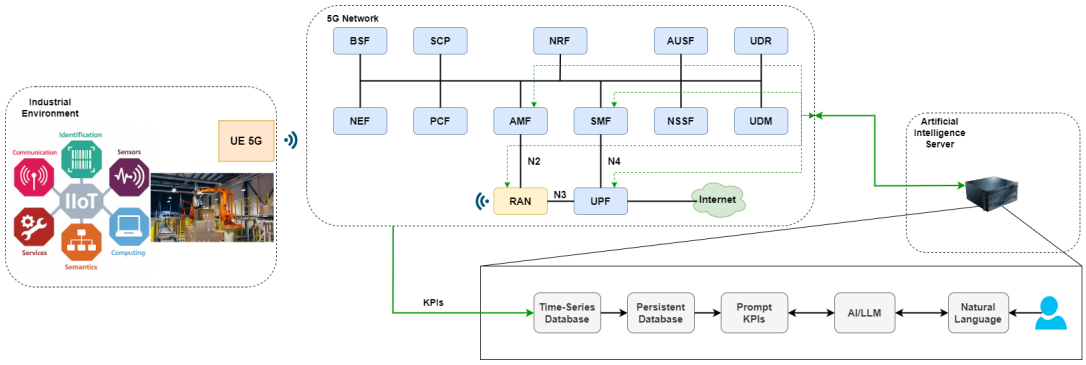


Figure 1. High-level overview of the experimental observability platform. The diagram highlights the main components involved in private 5G telemetry collection and LLM-based analysis.

2. Related Work

The deployment of private 5G networks and O-RAN architectures has been extensively addressed in the literature, with an emphasis on the trade-off between functional disaggregation, reliability, and operator control [Eswaran and Honnavalli 2022, Azariah et al. 2024]. At the same time, research on 5G telemetry has largely focused on the exposition of standardized KPIs [Saha and Viswanathan 2022, Panek et al. 2025], yet these approaches often lack integrated mechanisms for high-level interactive reasoning or automated diagnostics.

To address operational complexity, recent studies have proposed the use of LLMs as interfaces for network management, exploring applications in Software Defined Networks (SDN) orchestration [Araujo et al. 2024], code generation [Mani et al. 2023], and intent-based networking [Mcnamara et al. 2023, Xu et al. 2023]. Given the typical resource constraints of industrial edge infrastructures, the viability of quantized models has also been investigated, with analyzes focusing on latency-energy balance and domain-specific fine-tuning [Husom et al. 2025, Arya and Simmhan 2025, Kan et al. 2024].

However, independent evaluations indicate that LLM-based text-to-SQL systems exhibit significant performance degradation when applied to complex, real-world industrial schemas compared to curated benchmarks [Nascimento et al. 2025]. This discrepancy underscores the need for validation strategies that transcend mere syntactic checks [Zhan et al. 2025, Zahir and Qadi 2016].

As summarized in Table 1, the existing literature treats telemetry ingestion, natural-language analytics, and query validation in isolation. This work bridges these fragmented domains by integrating standardized 5G telemetry with on-premises edge analytics, enforced by the proposed Onion Validation methodology, within a unified experimental setup. More specifically, the novelty of this work lies in jointly evaluating standardized private 5G telemetry ingestion, local LLM-based querying, and auditable multi-layer validation within an integrated experimental framework.

Table 1. Comparison with related work (as described in Section 2).

Work	Pvt 5G / O-RAN	Telemetry	NL	Text-to-SQL	Multi-layer	Audit
Eswaran & Honnavalli (2022)	✓	–	–	–	–	–
Azariah et al. (2024)	✓	–	–	–	–	–
Saha & Viswanathan (2022)	✓	✓	–	–	–	–
Panek et al. (2025)	–	✓	–	–	–	–
Araujo et al. (2024)	– (SDN)	–	✓	–	–	–
McNamara et al. (2023)	✓	–	✓	–	–	–
Mani et al. (2023)	–	–	✓	–	–	–
Kan et al. (2024)	✓	–	✓	partial	–	–
Nascimento et al. (2025)	–	–	✓	✓	–	–
Cui et al. (2025)	–	–	–	–	plan-based	–
This work	✓	✓	✓	✓	✓	✓

3. Proposed Platform

The proposed platform is implemented as an experimental two-tier setup that separates continuous network telemetry ingestion from LLM-based on-demand analysis, as illustrated in Figure 2[1]. The first tier (Telemetry Tier) is responsible for the exposure, transport, and storage of KPI from a private 5G network, while the second tier (Analytics Tier) processes queries in natural language over the collected observability data.

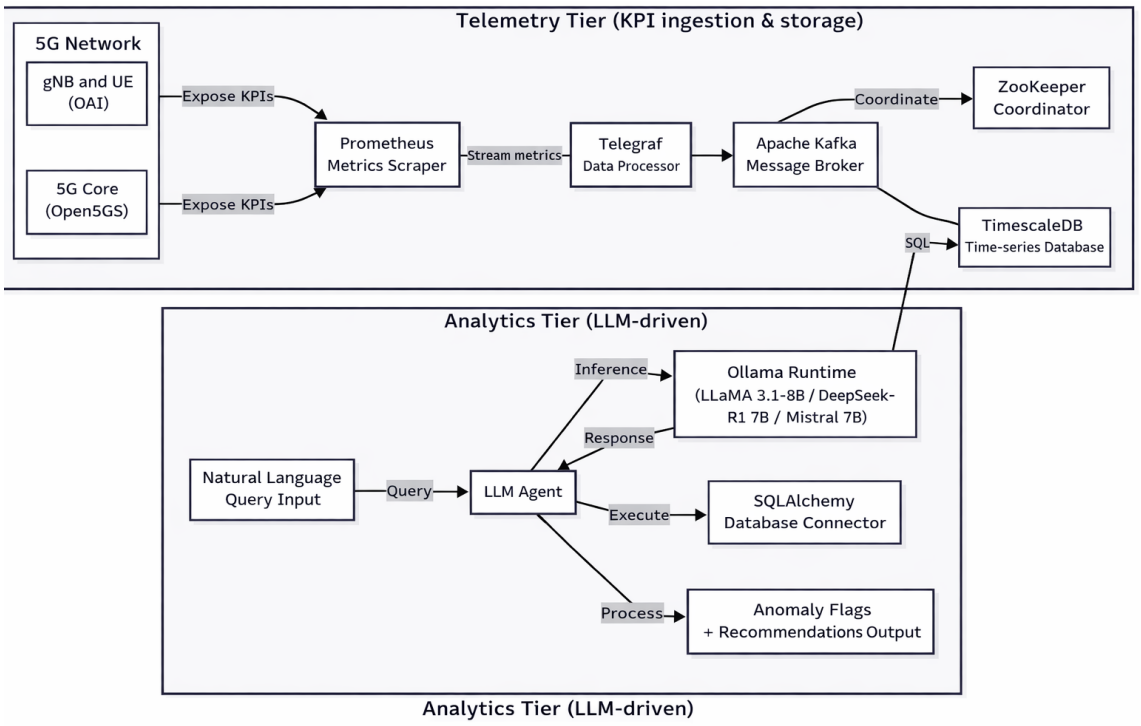


Figure 2. High-level overview of the experimental observability platform. The diagram highlights the main components involved in private 5G telemetry collection and LLM-based analysis.

The Telemetry Tier collects standardized KPIs from a private 5G network implemented using Open5GS [Open5GS 2025] and OpenAirInter-

[1]Source code and configuration files available at: `https://gitlab.com/lukilme/open-5g`

face [OpenAirInterface Software Alliance 2025]. The KPIs defined by ETSI TS 128 554 [ETSI 2021] are exposed via Prometheus exporters, scraped every 5 seconds, and forwarded through a Prometheus–Telegraf–Kafka pipeline. The metrics are transcoded into the InfluxDB Line Protocol and consumed by a Python-based Kafka client, which batches and persists the data in TimescaleDB, a PostgreSQL-based time-series database [Timescale Inc. 2025, The PostgreSQL Global Development Group 2025]. The telemetry pipeline is configured to support low-latency ingestion under the evaluated load conditions, maintaining ingestion rates of up to 15000 samples/s. This configuration is consistent with the slice-level observability requirements reported in previous experimental studies, such as MonArch [Saha et al. 2023], and provides timely telemetry for downstream analysis.

The Analytics Tier hosts an on-premises LLM-based observability agent implemented according to the Onion Validation methodology. The agent interfaces with TimescaleDB via SQLAlchemy [SQLAlchemy Authors 2025] and queries locally served language models through an Ollama daemon [Ollama Inc. 2025]. Five 8-bit quantized LLMs are evaluated: LLaMA 3.1–8B, DeepSeek-R1 7B, Mistral-7B, Phi-3 (3.8B), and LLaMA 3.2 (2B), with the latter serving as a baseline. This selection enables a controlled comparison across model scales in an edge-constrained setting [Arya and Simmhan 2025].

From an architectural perspective, the agent receives natural-language requests, accesses the observability database, and processes the interaction through the Onion Validation pipeline, which combines intent screening, SQL generation, validation, execution control, and auditable output handling. This design enables the integration of standardized telemetry with a local natural-language interface while preserving traceability across the analytical workflow. A detailed description of the Onion Validation stages is provided in Section 4.

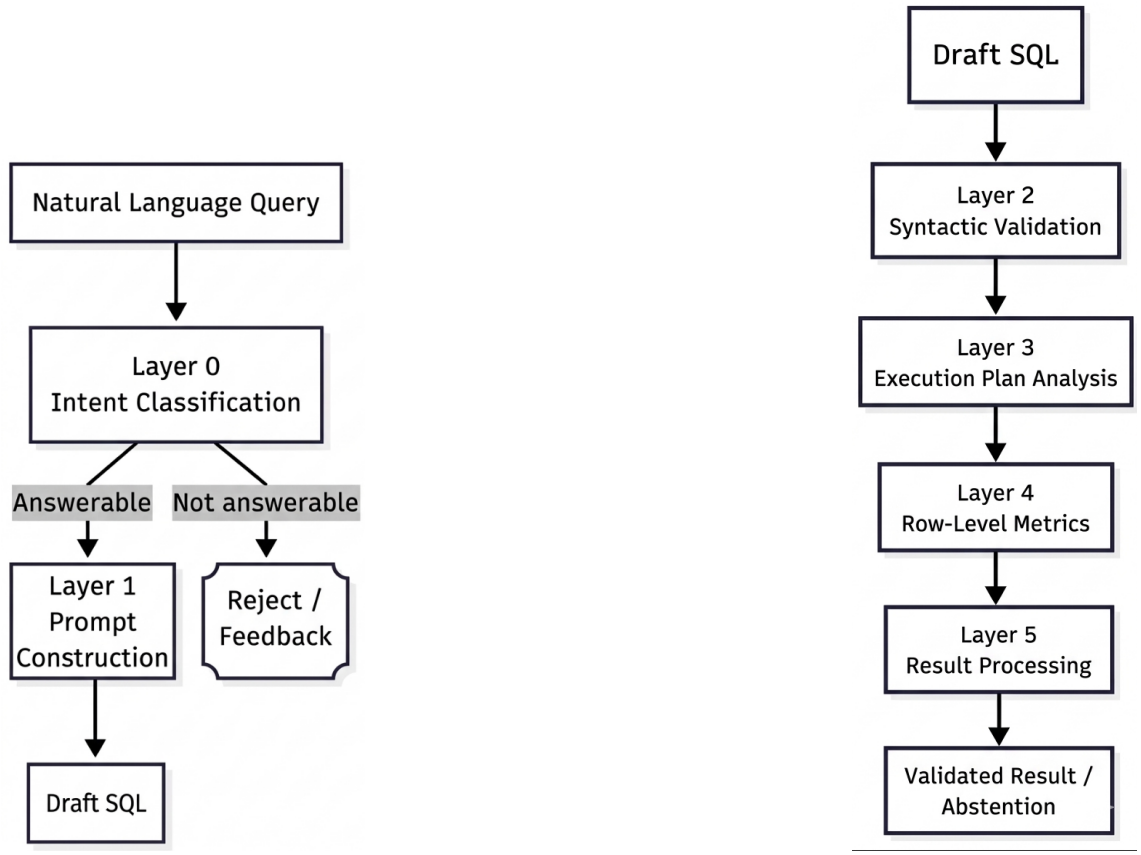
All prompts, generated SQL queries, execution traces, intent classifications, and system outputs are persisted in an immutable audit log. This mechanism ensures reproducibility and facilitates post-hoc analysis, aligning with recent approaches to auditable LLM-based observability systems [Dandoush et al. 2024]. In general, the proposed platform combines standardized private 5G telemetry with auditable, on-premises LLM-based analysis to support near-real-time inspection of observability data in industrial network environments.

4. Methodology

This section details the methodology used to instantiate, exercise, and evaluate the proposed LLM-driven observability platform. A conceptual overview of the Onion Validation methodology is provided in Figure 3, which summarizes the two main processing phases. The methodology covers: (i) the Onion Validation framework structure, (ii) execution architecture variants, (iii) testbed setup, (iv) data collection and test set preparation, and (v) the evaluation protocol.

4.1. Onion Validation Framework

The Onion Validation framework is implemented as a six-stage inference pipeline, beginning with an intent-based pre-validation stage (Layer 0) and followed by five validation



(a) Phase 1: Intent-based pre-validation and draft SQL generation.

(b) Phase 2: Validation and output processing.

Figure 3. Conceptual overview of the Onion Validation methodology. Phase 1 performs intent-aware draft generation, while Phase 2 applies sequential validation and output processing layers.

layers (Layers 1–5) in which each target distinct, non-overlapping dimensions of SQL query correctness. Layer 0 filters the queries for admissibility and is not counted as a validation step. The remaining layers operate in a fixed order but are analytically decoupled, allowing a structured and fine-grained examination of the behavior of LLM models. This layered structure contrasts with monolithic accuracy-based evaluations, which implicitly assume that all user queries are well-defined and executable.

As shown in Figure 3, the framework unfolds in two main phases: Phase 1 (layers 0 and 1) handles input interpretation and SQL construction, and Phase 2 (layers 2 to 5) performs ordered validation checks and subsequent output handling. Designed as a diagnostic evaluation architecture rather than a new SQL generation model, Onion Validation fills an assessment gap by breaking down SQL query correctness into separate, orthogonal stages for analyzing the performance of LLM-based text-to-SQL systems.

4.1.1. Phase 1: Input Processing and Draft Generation

In Layer 0, incoming natural language queries are first classified into one of four intent categories: *Answerable*, *Ambiguous*, *Impossible*, or *Out-of-Scope*. This pre-validation step, shown in Figure 3a, determines whether a query is admissible for SQL generation. The LLM under evaluation performs the classification using explicit decision rules and the exposed database schema. Queries classified as *Impossible* or *Out-of-Scope* are rejected with explanatory feedback. *Ambiguous* queries are not forwarded to SQL generation under the current evaluation setup. The distribution of predicted intent classes characterizes the query routing behavior and directly affects the fraction of queries propagated to subsequent stages.

In Layer 1, for queries classified as *Answerable*, enriched prompts are constructed by integrating the TimescaleDB schema, ETSI TS 128 554 KPI definitions, and SQL syntax constraints. Based on this context, the LLM generates a draft SQL query. This step completes Phase 1 of the framework. Although SQL generation is not evaluated in isolation, it is documented to provide a complete view of the system workflow.

4.1.2. Phase 2: Validation and Output Processing

Phase 2, summarized in Figure 3b, evaluates the queries created in Phase 1 through a sequence of validation and result processing layers.

In Layer 2, the SQL query draft is validated for PostgreSQL grammar correctness, table and column existence, type compatibility, and clause ordering. Queries that fail syntactic validation are logged and may trigger regeneration with corrective prompting. Invalid queries are strictly blocked from execution.

In Layer 3, the syntactically valid queries are analyzed using PostgreSQL `EXPLAIN`. Execution plans are evaluated for semantic conformance, ensuring appropriate table usage, joins, filters, and aggregations prior to execution. Execution-plan conformance is defined through rule-based validation and structural consistency over plan components, supporting interpretable diagnostics.

In layers 4 and 5, the executed query results are compared with hand-written ground-truth queries using row-level precision, recall, F1-score, and Jaccard index. For non-*Answerable* queries, correct abstention is considered a success. The final outputs are distilled into narrative summaries, anomaly indicators, and recommended actions when applicable. All intermediate artifacts, execution metadata, and validation outcomes are persisted in an Audit Log to ensure traceability and reproducibility.

4.2. Execution Schema Variants

All models are evaluated under two database schema configurations to assess design trade-offs:

- Unified Schema: the unified schema consolidates all 5G network metrics into a single table, exposing a serialized and dense representation of the data to the LLM. This approach simplifies data access and reduces structural ambiguity during query interpretation, at the cost of reduced normalization;

- Partitioned Schema: the partitioned schema organizes network metrics across multiple related tables, providing a more abstract and normalized representation of the 5G data. Although this design improves semantic separation between metric domains, it requires the model to reason over multiple tables when answering queries, potentially increasing latency.

4.3. Testbed Setup

An experimental private 5G testbed was instantiated in a laboratory environment using Open5GS v2.7.2 and OpenAirInterface v2025.w14. ETSI TS 128 554 KPIs were scraped every 5 seconds via Prometheus, processed by Telegraf (Influx Line Protocol), streamed through Apache Kafka, and persisted in TimescaleDB via a custom Python consumer. The LLM agent and Ollama runtime were deployed on a dedicated on-premises host equipped with an Intel Xeon Silver 4316 CPU (20 cores, 40 threads) and 32 GB of RAM, running Ubuntu 18.04.6 LTS (Linux 5.4).

4.4. Data Collection and Test Set Preparation

We constructed a dataset of 130 natural-language observability queries grounded in experimental telemetry collected under controlled traffic conditions. The query set is stratified by intent into *Answerable*, *Impossible*, and *Ambiguous* categories (30 queries each) to facilitate per-class performance reporting and failure-mode analysis. Additionally, 40 *Out-of-Scope* queries were included to stress-test the system's resilience against semantic mismatches (e.g., requests referring to non-existent KPIs or external domains). Table 2 shows examples of each query category.

Table 2. Examples of Intent Classification (Layer 0) for Observability Queries.

User Query	Classification
Percentage variation of regmobfail: last 7 days vs previous 7 days	ANSWERABLE
Show the worst records in terms of quality.	AMBIGUOUS
Which user consumed the most data today?	IMPOSSIBLE
Which user experienced the lowest signal quality?	OUT OF SCOPE

Given the high domain-specificity of 5G observability, this sample size reflects a deliberate trade-off between statistical power and the rigor required to manually validate complex, ETSI/3GPP-aligned ground-truth SQL (involving precise KPI naming, time-window semantics, and aggregation levels). Consequently, we report confidence intervals and interpret performance differences with appropriate statistical caution. The ground truth for *Answerable* queries was subjected to a dual-verification procedure, and near-duplicates were removed to minimize paraphrase leakage.

4.5. LLM-Agent Evaluation Procedure

For each query, the agent executed the full inference pipeline starting from intent-based pre-validation. Five LLMs served by Ollama in 8-bit quantized precision were evaluated: LLaMA 3.1-8B, DeepSeek-R1 7B, Mistral-7B, Phi-3 (3.8B), and the LLaMA 3.2 (used as baseline). Each model was tested using partitioned and unified execution architectures.

Evaluation metrics included functional correctness, intent classification distribution, and end-to-end latency. All prompts, generated SQL statements, execution traces, intent classifications, and model responses were persisted in a structured JSON format to support auditability and post-hoc analysis.

5. Results and Discussion

This section presents the experimental results in two parts: (i) an analysis of the observability metrics collected from the private 5G testbed, and (ii) a comprehensive performance assessment of the LLM agents under the proposed validation framework.

5.1. Threats to Validity

The evaluation relies on experimental telemetry generated under controlled conditions in a private 5G testbed, in which the core network is implemented using Open5GS, and the RAN is simulated/emulated. Although the telemetry pipeline, database schema, and KPI definitions strictly adhere to ETSI/3GPP standards, this setup does not capture the full stochastic variability of real-world radio environments, such as complex multi-path fading, massive user mobility, or inter-cell interference dynamics.

Consequently, the statistical distributions of specific KPIs may differ from those observed in large-scale production deployments. The reported functional correctness and latency metrics should, therefore, be interpreted as evidence of architectural feasibility and comparative behavior for edge-deployed neuro-symbolic systems, rather than as an absolute benchmark for production-grade carrier networks.

5.2. Observability Metrics in 5G Networks

The observability metrics utilized in this work follow the normative definitions of ETSI/3GPP, comprising throughput, Block Error Rate (BLER), RSRP/SINR, and core-network interface counters. The metrics are derived from the Open5GS/OpenAirInterface testbed operating under controlled traffic generation.

To populate the observability schema with realistic temporal variations, we executed traffic scenarios including *iperf*-based throughput saturation, bulk data transfers, and continuous streaming sessions. Figure 4 illustrates the End-to-End (Core-to-UE) latency behavior measured under these conditions. Although the absolute values are specific to the testbed environment, they preserve the data types, ranges, and cross-domain correlations required to exercise the LLM's reasoning capabilities over time-series data.

All telemetry is ingested into a time-series database structured according to ETSI TS 128 554 naming conventions. This schema-rich dataset underpins the natural-language query workload used in the LLM evaluation, enabling a reproducible assessment of text-to-SQL generation and validation behavior.

5.3. LLM Model Performance Assessment

Five locally deployed LLMs were assessed using the Onion Validation methodology. The evaluation considered functional correctness, intent classification behavior, and end-to-end execution latency. Unless stated otherwise, reported latency values denote full pipeline execution, including LLM inference, validation layers, regeneration loops, and result processing.

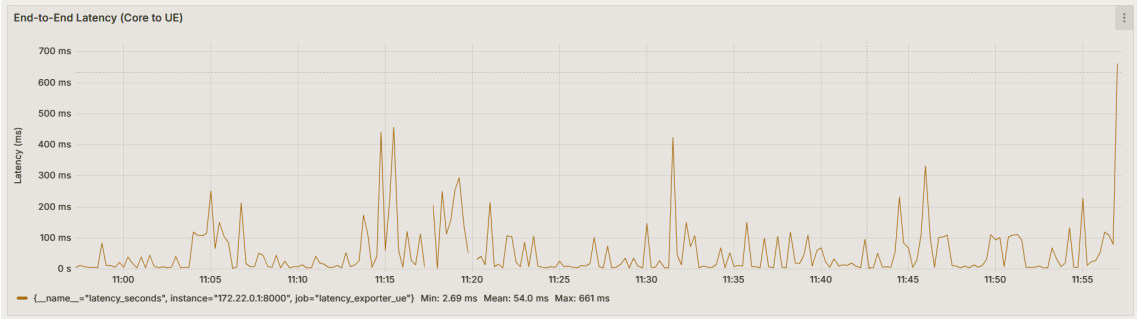


Figure 4. Latency measurements in the private 5G network under industrial environment conditions.

We report point estimates accompanied by 95% confidence intervals (CIs). For correctness metrics, we employ non-parametric bootstrapping over the query set (resampling query IDs with replacement, $B = 10,000$ iterations) and derive the CI bounds from the 2.5th and 97.5th percentiles. Regarding latency, we report the mean, median, and 95th percentile (P95) of the end-to-end wall-clock time. This metric is measured from the initial request submission to the final output generation, strictly including all validation layers and potential regeneration loops.

5.3.1. Accuracy and Functional Correctness

Table 3 reports the functional correctness (accuracy) for each model under partitioned and unified execution. Accuracy is strictly defined as the proportion of *answers in the correct category*. Reported values include point estimates and 95% bootstrap confidence intervals (CIs) computed over the query set ($B = 10{,}000$).

Table 3. Execution accuracy on *Correct* category with 95% Confidence Intervals ($N = 30$, $B = 10{,}000$).

Model	Partitioned (95% CI)	Unified (95% CI)	Δ (pp)	Aggregated (95% CI)
LLaMA 3.1–8B	28.5% [20.1–36.8]	26.9% [19.5–34.2]	-1.6	**27.7%** [22.0–33.5]
DeepSeek-R1 7B	26.2% [18.8–33.5]	25.4% [17.9–32.8]	-0.8	25.8% [20.5–31.1]
Phi-3 (3.8B)	24.1% [16.5–31.7]	23.7% [16.2–31.2]	-0.4	23.9% [18.1–29.7]
LLaMA 3.2 (2B)	22.3% [15.4–29.2]	26.9% [19.5–34.3]	+4.6	24.6% [19.3–29.9]
Mistral-7B	11.5% [06.2–16.8]	19.2% [12.5–25.9]	+7.7	15.4% [11.0–19.8]

The aggregated accuracy peaks at 27.7% (LLaMA 3.1). These values reflect the strict validation rules enforced by the Onion Validation methodology, which prioritizes operational safety over partial recall. The significant performance gap for Mistral-7B between partitioned and unified architectures (+7.7 pp) suggests that this model benefits substantially from the data locality and context retention provided by the unified execution flow.

5.3.2. Intent Classification Behavior

Intent classification (Layer 0) acts as the system's gatekeeper. Table 4 details the distribution of the predicted intent classes in unified execution.

Table 4. Distribution of predicted intent classes by model (unified execution).

Model	Answer.	Ambig.	Imposs.	Out Scope
LLaMA 3.1–8B	75.0%	16.7%	0.0%	8.3%
DeepSeek-R1 7B	60.0%	18.3%	5.0%	16.7%
Phi-3 (3.8B)	48.3%	21.7%	15.0%	15.0%
LLaMA 3.2 (2B)	21.7%	36.7%	40.0%	1.7%
Mistral-7B	31.7%	21.7%	21.7%	25.0%

A clear correlation is observed between the model scale and the confidence in the intention. Larger models (LLaMA 3.1, DeepSeek) tend to be more aggressive, classifying a majority of queries as *Answerable*, different from the baseline model (LLaMA 3.2).

5.3.3. End-to-End Latency

We report end-to-end latency of the full Onion Validation pipeline (including inference, validation, and any regeneration) as mean/median/P95 across queries. This captures both the typical operator experience (median) and the tail behavior relevant to operational feasibility (P95).

Table 5. End-to-End Latency statistics (Mean, Median, P95) for the full Onion Validation pipeline. Values are reported in seconds.

Model	Architecture	Mean (s)	Median (s)	P95 (s)
LLaMA 3.1–8B	Unified	10.9	10.8	11.1
	Partitioned	15.1	15.1	15.4
DeepSeek-R1 7B	Unified	60.9	53.3	82.5
	Partitioned	57.9	53.3	108.6
Phi-3 (3.8B)	Unified	22.7	4.7	81.4
	Partitioned	17.7	5.3	64.1
Mistral-7B	Unified	8.4	8.4	13.5
	Partitioned	13.4	13.4	13.5
LLaMA 3.2 (2B)	Unified	3.1	3.1	3.2
	Partitioned	5.2	5.0	5.1

The results demonstrate that the unified schema tends to reduce latency in the majority models, particularly within more recent and higher-capacity architectures, such as LLaMA 3.1, LLaMA 3.2, and Mistral. In these instances, data consolidation within a single table mitigates structural interpretation overhead, thus consistently reducing both

mean latency and the 95th percentile (P95). This suggests that such models effectively handle serialized data structures, extracting relevant metrics without a heavy reliance on explicit relational organization.

In contrast, models such as DeepSeek-R1 and Phi-3 exhibit different behavioral patterns. DeepSeek-R1 achieves superior performance with the partitioned schema, suggesting a greater reliance on a clear, normalized semantic structure. Phi-3, in addition to the higher mean latencies, exhibits a significantly elevated P95 in both scenarios, with performance degradation in the unified schema; this indicates reasoning instability when confronted with high information density. This reinforces the premise that smaller or more sensitive models tend to benefit from explicit schemas, even at the expense of increased mean latency.

Overall, the data indicate that latency is not merely a function of data volume, but also of the cognitive load imposed by the schema upon the model. Unified schemas favor robust, modern architectures, whereas partitioned schemas enhance semantic predictability for less capable models. Therefore, in 5G environments where P95 and consistency are critical, schema selection must consider not only mean efficiency but also the compatibility between the data structure and the abstraction capability of the deployed LLM.

6. Conclusion and Future Work

This paper presented an end-to-end, on-premises observability platform for private 5G networks, integrating the ingestion of 3GPP-standardized telemetry with local LLM-based natural language analytics. To address the reliability challenges of generative AI in industrial settings, we introduced the Onion Validation methodology, a layered pipeline designed to support the systematic inspection, validation, and auditability of LLM-generated SQL queries over experimental data.

We evaluated five quantized LLMs (2B–8B parameters) using 130 operator-inspired queries within an Open5GS/OpenAirInterface testbed. The results demonstrate that, for edge-deployed quantized models, syntactic validity does not guarantee functional correctness. The peak zero-shot performance of 27.7% demonstrates that, although locally deployed LLMs preserve data sovereignty, they necessitate stringent, multi-layer evaluation procedures to adequately reduce operational risk. Furthermore, trade-off analysis between unified and partitioned architectures reveals significant latency implications, guiding the design of real-time diagnostics systems.

Beyond the experimental results, the practical adoption of LLM-based analytics in industrial environments must also be examined from an operational perspective. Although local deployment helps preserve data sovereignty, it does not eliminate important limitations related to computational cost, infrastructure provisioning, and model maintenance. In addition, the possibility of hallucinations, semantically incorrect outputs, and unsafe query generation remains a relevant concern, particularly when natural language interfaces are used to inspect sensitive network data under complex operational conditions. For this reason, the use of LLMs in industrial observability should be framed as a supervised assistive capability, supported by validation layers, auditing mechanisms, access control policies, and human oversight, rather than as a fully autonomous decision-making component.

Future work will focus on extending the experimental framework to incorpo-

rate standardized domain knowledge. Specifically, our goal is to implement Retrieval-Augmented Generation (RAG) mechanisms to inject normative 3GPP and ETSI definitions into the context window during inference. Additional directions include the exploration of schema-aware retrieval strategies and the expansion of the ground-truth dataset to cover multi-domain scenarios involving heterogeneous radio access technologies.

Acknowledgment

This work is supported by EMBRAPII (BFA 2301.0001) and the companies Cisco, Prysmian, and MPT Cable. The authors also thank CPQD, Inatel, Taggen, Data Machina, CNPq (307108/2025-2), IFPB, and the IFPB Innovation Hub.

References

Araujo, A. S. et al. (2024). An agentic approach for dynamic software-defined network management using large language models. In *2024 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*.

Arya, M. and Simmhan, Y. (2025). Understanding the performance and power of llm inferencing on edge accelerators. *arXiv e-prints*.

Azariah, W., Bimo, F. A., Lin, C.-W., Cheng, R.-G., Nikaein, N., and Jana, R. (2024). A survey on open radio access networks: challenges, research directions, and open source approaches. *Sensors*, 24(3):1038.

Dandoush, A., Kumarskandpriya, V., Uddin, M., and Khalil, U. (2024). Large language models meet network slicing management and orchestration. *arXiv preprint arXiv:2403.13721*.

Eswaran, S. and Honnavalli, P. (2022). Private 5G networks: A survey on enabling technologies, deployment models, use cases, and research directions. *Telecommunication Systems*, 82(1):3–26.

ETSI (2021). Ts 128 554 v16.7.0 - 5g; management and orchestration; 5g end to end key performance indicators (kpi). Technical report, ETSI. [Online].

Husom, E. J., Goknil, A., et al. (2025). Sustainable llm inference for edge ai: evaluating quantized llms for energy efficiency, output accuracy, and inference latency. *arXiv preprint arXiv:2504.03360*.

Kan, K. B., Mun, H., Cao, G., and Lee, Y. (2024). Mobile-llama: instruction fine-tuning open-source llm for network analysis in 5g networks. *IEEE Network*, 38(5):76–83.

Mani, S. K., Zhou, Y., Hsieh, K., Segarra, S., Eberl, T., Azulai, E., Frizler, I., Chandra, R., and Kandula, S. (2023). Enhancing network management using code generated by large language models. In *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks*, pages 196–204.

Mcnamara, J., Camps-Mur, D., Goodarzi, M., Frank, H., Chinchilla-Romero, L., Cañellas, F., Fernández-Fernández, A., and Yan, S. (2023). Nlp powered intent based network management for private 5g networks. *IEEE Access*, 11:36642–36657.

Nascimento, E. R. S., Garcia, G., et al. (2025). Llm-based text-to-sql for real-world databases. *SN Computer Science*, 6(2):130.

Ollama Inc. (2025). *Ollama: local large language model runtime*. https://github.com/ollama/ollama.

Open5GS (2025). Open5gs: open source 5g core network. `https://open5gs.org`.

OpenAirInterface Software Alliance (2025). Openairinterface 5g wireless implementation. `https://openairinterface.org`.

Panek, M., Jabłoński, I., and Woźniak, M. (2025). Automatic performance assessment: Step toward autonomous mobile network management systems. *IEEE Communications Magazine*, 63(4):73–79.

Saha, N., Shahriar, N., Boutaba, R., and Saleh, A. (2023). Monarch: network slice monitoring architecture for cloud native 5g deployments. In *NOMS 2023 - 2023 IEEE/IFIP Network Operations and Management Symposium*, pages 1–7.

Saha, S. and Viswanathan, H. (2022). Demonstrating network slice kpi monitoring in a 5g testbed. In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE. Demo Track.

SQLAlchemy Authors (2025). Sqlalchemy: the database toolkit for python. https://www.sqlalchemy.org/.

The PostgreSQL Global Development Group (2025). Postgresql: the world's most advanced open source relational database. `https://www.postgresql.org/`.

Timescale Inc. (2025). Timescaledb: an open-source time-series database powered by postgresql. `https://www.timescale.com/`.

Xu, X., Chen, H., Simsarian, J. E., Ryf, R., Mazur, M., Dallachiesa, L., Fontaine, N. K., and Neilson, D. T. (2023). Demonstration of voice user interface for intelligent network orchestration. In *Optical Fiber Communication Conference (OFC)*. Paper M3Z.11, Optica Publishing Group.

Yaqub, M. Z. and Alsabban, A. (2023). Industry-4.0-enabled digital transformation: prospects, instruments, challenges, and implications for business strategies. *Sustainability*, 15(11):8553.

Zahir, J. and Qadi, A. E. (2016). A recommendation system for execution plans using machine learning. *Mathematical and Computational Applications*, 21(2):23.

Zhan, Y., Cui, L., Weng, H., Wang, G., Tian, Y., Liu, B., Yang, Y., Yin, X., Xie, J., and Sun, Y. (2025). Towards database-free text-to-SQL evaluation: A graph-based metric for functional correctness. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING)*, pages 4586–4610.