



A Data-Driven Study on the use of Small Language Models & Large Language Models for Interpreting Complex Industrial Data and Deriving Actionable Outcome

Md Syeedur Rahman¹; Sharmin Rahman²;

[1]. Solution Developer, Plant Design I/O, Gonzales, Louisiana, USA; syeed_rahman@live.com

[2]. Graduate Student, Department of Electrical & Computer Engineering, Lamar University, Beaumont, Texas, USA; Email : sharmin.rahman89@outlook.com

Doi: [10.63125/5vyhgs98](https://doi.org/10.63125/5vyhgs98)

Received: 19 August 2025; **Revised:** 21 September 2025; **Accepted:** 19 October 2025; **Published:** 15 November 2025;

Abstract

Modern industrial systems generate unprecedented volumes of heterogeneous data, spanning high-frequency sensor streams, SCADA outputs, unstructured maintenance logs, and operational reports. Integrating and interpreting these disparate data sources creates a persistent challenge for timely decision-making, as traditional predictive models often lack the contextual reasoning required to translate raw metrics into actionable operational insights. This paper presents a comprehensive, data-driven study evaluating the efficacy of Small Language Models (SLMs) and Large Language Models (LLMs) in interpreting complex industrial data and deriving actionable outcomes. We systematically compare how these models differ in their capacity to summarize complex operational patterns, explain transient anomalies, infer root causes, and prescribe maintenance decisions. Crucially, this evaluation grounds model capabilities against practical industrial constraints, including inference latency, computational cost, data privacy, and edge-deployment feasibility. A data-driven comparative framework is proposed to rigorously evaluate model performance across interpretation quality, domain-specific accuracy, robustness to noisy data, and the tangible usefulness of generated recommendations. Furthermore, this study investigates the viability of a hybrid, tiered architecture – combining the localized, low-latency efficiency of SLMs for immediate processing with the advanced, high-level reasoning capacity of LLMs for complex diagnostics – to optimize both reliability and operational value. Ultimately, the findings are intended to inform the design of language-model-based industrial analytics systems that are computationally efficient, contextually aware, and readily deployable in real-world settings.

Keywords

Small Language Models; Large Language Models; Industrial Data Analytics; Data Interpretation; Actionable Insights; Anomaly Detection; Decision Support; Hybrid Language Models.

INTRODUCTION

Industrial organizations are increasingly operating in data-rich environments in which simulation outputs, sensor streams, maintenance logs, production records, and other heterogeneous sources must be transformed into knowledge that can support decision-making (Gröger, 2022). The industrial analytics literature frames this challenge as a central element of industrial value creation: extracting insights from data is critical for optimizing processes and enhancing product features, yet the field remains fragmented across specialized methods and use cases (Gröger, 2022). In parallel, predictive maintenance research shows that realistic industrial datasets are often irregular, incomplete, and structurally complex, and that such irregularities can materially affect model performance (Autran et al., 2024). These characteristics make industrial data interpretation more demanding than routine tabular analysis, because useful outcomes must be derived from noisy, domain-specific, and operationally consequential inputs. As a result, the question is not merely whether a model can process industrial data, but whether it can interpret it in a way that yields trustworthy and actionable outcomes. This interpretive requirement is especially important in industrial settings because model outputs must often be understood and acted upon by engineers, plant operators, and decision-makers rather than by data scientists alone. Explainability research in manufacturing emphasizes that users must trust model predictions if such systems are to be adopted in practice (Goldman et al., 2023). In that work, explainable AI methods were used to make predictions about weld quality and dimensional variability more transparent, illustrating how explanations can reveal which signals or process factors influence outcomes (Goldman et al., 2023). More broadly, the machine learning and data mining literature distinguishes between black-box models, which represent implicit knowledge, and white-box models, which encode explicit knowledge, and argues for integrating the two perspectives to support inspection and control of learned models (Atzmueller et al., 2024). Similar concerns appear in process prescriptive analytics, where recommendations without explanations may fail to engage process owners or gain buy-in. Large language models have emerged as a promising tool for this task because they can understand and generate human-like text and have demonstrated strong performance across tasks involving language understanding, reasoning, and generation (Bharathi Mohan et al., 2024; Raza et al., 2025). Industrial applications of LLMs already span several domains, including predictive maintenance, customer service automation, fraud detection, and research support, reflecting their capacity to automate language-intensive workflows and provide deeper insights from complex information sources (Raza et al., 2025). In industrial diagnostics, LLMs have been used to integrate unstructured textual data such as technical documents and maintenance logs into condition-monitoring systems, where their contextual understanding and information extraction can improve diagnostic accuracy (Jose et al., 2024). Related work in business process management shows that LLMs can also improve comprehension of complex process models, especially when model abstractions and prompting strategies are carefully designed (Kourani et al., 2026). These studies suggest that LLMs are well suited to scenarios in which industrial knowledge is embedded in mixed textual and structured forms and must be translated into accessible reasoning or summaries.

At the same time, the provided literature consistently notes important limitations of LLMs. They require substantial computational resources, may depend on cloud APIs, and can raise privacy concerns and fine-tuning costs (Wang et al., 2025). Their general-purpose training can also make them less effective in specialized domains where domain-specific knowledge is essential (Li et al., 2025; Wang et al., 2025). In industrial settings, this limitation is particularly relevant because many tasks depend on local terminology, proprietary processes, or context-specific operational knowledge. Moreover, LLM-based data analysis is not uniformly reliable: an evaluation of LLM-generated data analysis workflows found that while LLMs could generate code for some tasks, executable code generation remained challenging, especially for complex analyses (Jansen et al., 2025). Another study on table understanding showed that improved prompt design and explicit structural encoding were needed to help GPT-3.5 parse complex tables effectively (Zhao et al., 2023). These findings indicate that LLMs can be powerful interpreters of complex data, but their performance may depend heavily on task framing, prompt engineering, and the availability of structured guidance.

The emerging role of small language models is therefore important in the industrial data context. The survey of small language models argues that SLMs are increasingly favored for low inference latency, cost-effectiveness, efficient development, easy customization, and suitability for resource-limited environments (Wang et al., 2025). It further notes that SLMs can address LLM limitations by supporting localized data handling for privacy, minimal latency for efficiency, and lightweight fine-tuning for domain knowledge acquisition (Wang et al., 2025). These properties are especially relevant for industrial deployment, where data may need to remain on-site, response times may be operationally constrained, and models must adapt to highly specific terminology or workflows. In addition, hybrid approaches suggest that SLMs need not be viewed merely as smaller substitutes for LLMs. A dynamic edge-cloud collaboration framework showed that an SLM on edge devices could interact token by token with a cloud LLM and achieve LLM-comparable quality at substantially lower cost (Hao et al., 2024). Similarly, BLADE combines a black-box LLM with a small domain-specific model, arguing that the small model preserves specialized knowledge while the LLM contributes general language comprehension and reasoning (Li et al., 2025). These studies indicate that SLMs may play a crucial role both as standalone models for constrained settings and as specialized complements to larger systems. The motivation for directly comparing small and large language models is therefore methodological as well as practical. Existing work shows that LLMs can support industrial text interpretation, diagnostics, table parsing, process comprehension, and automated insight generation (Jose et al., 2024; Kourani et al., 2026; Sánchez et al., 2025), yet it also shows that their effectiveness is constrained by resource demands and domain mismatch (Jansen et al., 2025; Wang et al., 2025). At the same time, SLMs are increasingly positioned as viable alternatives or collaborators, but their relative performance depends on the task, domain, and reasoning requirements (Sinha et al., 2024; Wang et al., 2025). A direct comparison is therefore needed to determine which model class better supports the interpretation of complex industrial data and the derivation of actionable outcomes under realistic constraints. Such a comparison is especially relevant for industrial analytics, where the value of a model lies not only in predictive accuracy but also in its ability to transform heterogeneous operational data into decisions that are understandable, timely, and implementable (Gröger, 2022; Padella et al., 2022).

Accordingly, a data-driven study comparing SLMs and LLMs can help clarify the trade-offs between domain specialization, computational efficiency, interpretability, and insight quality. It can also inform whether industrial applications are best served by large, general-purpose models, smaller domain-adapted models, or hybrid configurations that combine the strengths of both (Li et al., 2025; Hao et al., 2024). In a field where complex data, operational urgency, and trust requirements intersect, such evidence is essential for guiding the design of language-model-based analytic systems that are not only technically capable, but also practically useful.

RELATED WORK

Industrial analytics has emerged as a foundational component of digital transformation, reflecting the need to extract value from heterogeneous data across engineering, manufacturing, and product-use contexts. A recent overview characterizes industrial analytics as an interdisciplinary field at the intersection of data science and industrial engineering, emphasizing its role in industrial value creation and its dependence on data platforms, roles, and domain-specific workflows (Gröger, 2022). Within this broader setting, complex industrial data are often difficult to interpret because they may include sensor streams, process logs, technical documents, and other forms of structured or unstructured information. The literature increasingly positions language models as a means of bridging this gap by converting industrial data into interpretable representations and actionable outputs.

Large language models (LLMs) have been widely discussed as general-purpose systems for understanding and generating human-like text, with applications extending across healthcare, finance, education, and industrial settings (Bharathi Mohan et al., 2024; Raza et al., 2025). Surveys of LLMs emphasize their strong capabilities in text generation, question answering, and reasoning, while also noting persistent concerns related to ethical risk, bias, and computational cost (Bharathi Mohan et al., 2024; Raza et al., 2025). In industrial contexts, these models have been described as enabling predictive maintenance, recommendation, customer support, and deeper insight generation across data-intensive workflows (Raza et al., 2025). At the same time, industrial deployment raises practical constraints: many LLMs require substantial resources, may depend on cloud APIs, and can be difficult to adapt for

specialized domains where domain knowledge is essential (Wang et al., 2025). These observations motivate research that seeks to retain the reasoning strengths of LLMs while reducing cost and improving task fit.

Several studies have explored the direct use of LLMs for interpreting industrial or operational data. In prognostics and health management, integrating unstructured technical documents and maintenance logs with LLMs has been shown to expose data that are traditionally underused because of domain-specific jargon and text complexity (Jose et al., 2024). By supporting contextual understanding and information extraction, this approach can improve diagnostics when textual evidence is combined with existing condition-monitoring systems (Jose et al., 2024). Similarly, work on business process management shows that LLMs can enhance comprehension of complex process models when those models are transformed into abstractions suitable for language-based reasoning (Kourani et al., 2026). That framework reports that prompting strategies matter substantially and that advanced LLMs can identify critical quality issues such as deadlocks and soundness violations (Kourani et al., 2026). In a related vein, LLMs have been applied to complex table questions answering by reconstructing hierarchical tables into tuples and augmenting prompts with explanations of tuple meaning and reasoning steps, thereby improving the model's ability to parse structured information (Zhao et al., 2023). Together, these studies suggest that LLMs can support industrial interpretation tasks when the input representation is carefully adapted to the model's strengths.

A complementary line of work focuses on automating data analysis and generating actionable outputs from databases or other analytical inputs. One study introduces an LLM-based framework that produces concise textual insights from multi-table databases by combining hypothesis generation, SQL-based query answering, and summarization (Sánchez et al., 2025). The authors evaluate insights using both human judgment and automated metrics, indicating that the quality of generated output is not reducible to correctness alone but also depends on subjective insightfulness (Sánchez et al., 2025). More broadly, LLMs have been used to generate and execute data analysis pipelines, with an R package designed to translate textual objectives into executable workflows for genomics analysis (Jansen et al., 2025). This work highlights both promise and limitation: LLMs can generate useful code for some tasks, yet executable code generation remains challenging for more complex analysis requests (Jansen et al., 2025). These findings are relevant to industrial analytics because they underscore that actionable outcomes require not only natural-language generation but also reliable translation into analysis procedures that can be reviewed and trusted by domain experts.

Interpretability and explainability remain central concerns in domains where analytical recommendations have operational consequences. The explainable AI literature in manufacturing argues that users such as plant operators, engineers, and technicians must trust model outputs, and that post-hoc methods can help reveal which inputs influence predictions (Goldman et al., 2023). In that study, class activation maps and saliency maps were used to explain quality predictions and assess classifier robustness in ultrasonic weld and body-in-white applications (Goldman et al., 2023). More generally, the literature on explainable and interpretable machine learning distinguishes black-box statistical models from white-box approaches and argues for integrating post-hoc explanation with intrinsically interpretable modeling (Atzmueller et al., 2024). Similar arguments appear in process prescriptive analytics, where recommendations without sensible explanations are unlikely to engage process owners or earn trust (Padella et al., 2022). In healthcare, explainable predictive analytics frameworks combine interpretable machine learning with SHAP and LIME to improve transparency while supporting diagnostic and treatment decisions (Tahsina Akhter, & Deawn Md Alimozzaman, 2025). Across these domains, interpretability is not treated as a secondary feature but as a prerequisite for decision support, especially when the output of a model is intended to guide human action.

Small language models (SLMs) have gained attention as a response to the computational and operational limitations of LLMs. A recent survey notes that large models can be costly to deploy, may raise privacy concerns when used via cloud APIs, and often underperform in specialized domains due to insufficient domain-specific knowledge (Wang et al., 2025). In contrast, SLMs are described as suitable for low-latency, resource-constrained, and customizable deployments, particularly where localized data handling or lightweight fine-tuning is desirable (Wang et al., 2025). This framing is Hybrid systems that combine SLMs and LLMs represent an important direction for improving

efficiency while preserving quality. One edge-cloud collaborative inference approach places a small model on edge devices and uses token-level interaction with a cloud-side LLM to achieve LLM-comparable quality at reduced cost (Hao et al., 2024). This architecture is particularly relevant for settings where full LLM inference is impractical, yet higher-quality reasoning is still desired. Another hybrid method, BLADE, combines a black-box LLM with a small domain-specific model that preserves specialized knowledge, while the general model contributes broad language understanding and reasoning (Li et al., 2025). BLADE is positioned as a cost-efficient solution for adapting general LLMs to vertical domains, and its design reflects a recurring theme in the literature: general models are useful for reasoning over language, but domain-specific models can supply the knowledge needed for practical tasks (Li et al., 2025). These hybrid approaches are especially promising for industrial analytics, where organizations may need domain-aware interpretation without incurring the full computational burden of large-scale inference.

Retrieval-augmented or knowledge-assisted approaches also appear in the literature as a way to improve domain specificity, although the provided studies note limitations of naive retrieval augmentation in practical deployment. In the context of vertical domains, one paper observes that prior adaptation strategies for general LLMs include continuous pre-training or retrieval augmentation, but argues that these can be cost-intensive or unreliable in practice (Li et al., 2025). This concern is particularly salient for industrial data interpretation, where retrieved context must be accurate, timely, and tightly aligned with operational semantics. Related work on industrial textual data integration suggests that domain knowledge embedded in logs and documents can be extracted and combined with condition-monitoring data to improve diagnostics, which in effect serves a retrieval-like role by bringing latent textual evidence into analytic workflows (Jose et al., 2024). Likewise, abstractions used for process-model comprehension and table parsing demonstrate that careful representation design can function as an alternative to naïve retrieval by structuring complex inputs into forms that language models can reason over more effectively (Kourani et al., 2026; Zhao et al., 2023).

Evaluation trade-offs are a recurring theme across this literature. Studies of data analysis automation emphasize that correctness, executability, and usefulness are distinct dimensions that must all be considered (Jansen et al., 2025; Sánchez et al., 2025). In the process-model framework, evaluation spans automatic comparisons, qualitative issue identification, and user studies, reflecting the need to assess both technical performance and human-facing utility (Kourani et al., 2026). The small-model evaluation study similarly focuses on semantic correctness across task types, domains, and reasoning types, indicating that applicability depends strongly on context (Sinha et al., 2024). In industrial and manufacturing applications, the importance of realistic data is also evident: one predictive-maintenance study shows that introducing irregularities typical of real maintenance data significantly changes model performance, underscoring the sensitivity of evaluation to dataset realism (Autran et al., 2024). More broadly, industrial analytics research stresses the heterogeneity of data sources and the need for frameworks that account for real-world conditions rather than idealized benchmarks (Gröger, 2022).

Taken together, the literature suggests a converging research agenda: language models are increasingly used to interpret complex industrial data, but practical deployment depends on balancing reasoning quality, efficiency, interpretability, and domain adaptation. LLMs provide strong capabilities for transforming unstructured or structured industrial inputs into understandable outputs (Jose et al., 2024; Raza et al., 2025), yet their cost and domain limitations motivate SLMs, hybrid architectures, and domain-specific augmentation (Hao et al., 2024; Li et al., 2025; Wang et al., 2025). At the same time, decision support applications require explanations and trustworthy outputs, not merely predictions or generated text (Goldman et al., 2023; Padella et al., 2022; Tahsina Akhter, & Deawn Md Alimozzaman, 2025). The resulting trade-off is not simply between performance and efficiency, but among multiple desiderata: actionable insight, reliable execution, transparency, and operational feasibility. This tension frames current efforts to use language models for industrial data interpretation and points to the need for methods that are both technically capable and practically deployable (Gröger, 2022; Kourani et al., 2026).

PROBLEM DEFINITION

Industrial settings increasingly generate complex data that must be interpreted and translated into actionable outcomes. The problem addressed in this study is the lack of a clearly articulated framework, based on the provided source material, for examining how small language models and large language models can support the interpretation of such data and the derivation of practical outcomes. The available literature supports the broader need for structured transformation of complex work into usable outputs, but it does not directly define how language models should be evaluated for this purpose (Jaafari, 2020; Radford et al., 2020; Rapp et al., 2015).

Accordingly, this study is positioned to investigate the role of language-model-based approaches in converting complex industrial information into interpretable and actionable results, while remaining attentive to the practical constraints that shape effective dissemination and implementation of analytical work (Desinayak, 2023; Shukla et al., 2019).

RESEARCH QUESTIONS

1. How can small language models and large language models be used to interpret complex industrial data?
2. In what ways can these models support the derivation of actionable outcomes from industrial data?
3. What research gap remains in applying language-model-based approaches to complex industrial data interpretation and decision-oriented output?

METHODOLOGY

STUDY DESIGN

This study adopts a comparative empirical design to evaluate the extent to which small language models (SLMs) and large language models (LLMs) can interpret complex industrial data and translate that interpretation into actionable outcomes. The design is motivated by prior industrial work showing that LLMs can extract information from heterogeneous textual and operational data, but may face limitations related to cost, deployment constraints, and domain specificity (Raza et al., 2025), (Wang et al., 2025). In parallel, recent evidence suggests that smaller open models can be competitive in selected practical settings when the task, prompt structure, and domain alignment are appropriate (Sinha et al., 2025). The study is structured as a controlled benchmark comparison across identical industrial tasks, inputs, and evaluation conditions. Models are assessed on two related capabilities: (i) interpreting complex industrial data, including structured, semi-structured, and unstructured inputs; and (ii) producing actionable outputs, such as diagnostic explanations, recommended interventions, prioritized risks, or decision-support summaries. To ensure comparability, all models receive the same task definitions, input payloads, and output constraints. Where relevant, the workflow also distinguishes between direct response generation and multi-step reasoning or tool-mediated analysis, reflecting prior findings that prompting strategy can materially affect downstream data-analysis quality (Jansen et al., 2025; Kourani et al., 2026).

DATA SOURCES AND TASK CONSTRUCTION

The study uses industrial data drawn from the **Smart Manufacturing Maintenance Dataset** (Ziya, 2024) via **Kaggle**, which pairs real-time manufacturing sensor streams with operational decision criteria, maintenance costs, and technician logs. Because industrial analytics commonly involve heterogeneous data types and fragmented information flows, the dataset design follows an industrial analytics perspective that emphasizes multi-source integration and operational relevance (Gröger, 2022). Where textual domain knowledge is available, it is retained as part of the input because prior work indicates that unstructured technical documentation and maintenance narratives can materially improve diagnostic interpretation when incorporated into language-model-based pipelines (Jose et al., 2024).

Tasks are defined at the level of industrial decision-making episodes. Each task instance contains:

1. an input context describing the industrial situation;
2. an interpretation objective;
3. an expected action-oriented deliverable; and
4. a reference outcome or expert-derived target where available.

To reduce leakage and ensure fair comparison, task instances are partitioned by operational unit, time

window, or asset grouping as appropriate to the data structure. If the study includes multiple task categories, they should be reported explicitly, for example:

Table 1. Classification framework for industrial interpretation tasks.

Task Category	Input Type	Output Target	Evaluation Focus
Descriptive Interpretation	Structured / semi-structured industrial data	Comprehensive summary of key operational patterns	Factual accuracy, completeness
Diagnostic Analysis	Structured telemetry, operational logs, and technical narratives	Identified root cause or predicted failure mode	Structural correctness, rationale quality
Prescriptive Recommendation	Multi-modal operational data and contextual metadata	Actionable intervention or mitigation workflow	Operational utility, safety alignment, feasibility
Insight Generation	Multi-record or multi-table historical context	Concise operational insight and strategic summary	Relevance, novelty, structural domain correctness

DATA PREPROCESSING

Preprocessing is performed identically across model conditions to isolate differences attributable to model capability rather than data preparation. Structured fields are standardized, missing values are handled according to a strict listwise deletion policy to exclude incomplete records, ensuring that only fully populated telemetry streams are fed to the models, and categorical variables are structurally encoded where necessary. Temporal fields are ordered consistently to preserve chronology in industrial event sequences.

Algorithm 1. Deterministic Multi-Modal Data Preprocessing and Serialization Workflow

Input: Raw Industrial Telemetry Records R , Expert Inspection Annotations A
Output: Uniform Text-Serialized Model Context Corpus C , Ground-Truth Ontology O

```

1: Initialize  $C \leftarrow$  Empty List
2: Initialize  $O \leftarrow$  Empty Mapping
3: for each record  $r$  in  $R$  do
4:   if  $r$  contains missing target telemetry labels then
5:     Omit  $r$  (Execute listwise deletion policy)
6:   continue
7:   end if
8:    $r_{\text{numeric}} \leftarrow$  StandardizeNumericalTelemetry( $r$ .sensors)
9:    $r_{\text{cat}} \leftarrow$  StructuralOneHotEncoding( $r$ .categorical)
10:   $r_{\text{text}} \leftarrow$  StrippingBoilerplate( $r$ .maintenance_logs)
11:   $r_{\text{text}} \leftarrow$  PreserveTechnicalAcronyms( $r_{\text{text}}$ )
12:
13:  # Apply table-to-text serialization preserving structure (Zhao et al., 2023)
14:   $r_{\text{serial}} \leftarrow$  InjectTableBoundaries( $r_{\text{numeric}}$ ,  $r_{\text{cat}}$ )
15:   $r_{\text{serial}} \leftarrow$  AppendRowIdentifiersAndKeyValues( $r_{\text{serial}}$ ,  $r_{\text{text}}$ )
16:
17:  if ContextLength( $r_{\text{serial}}$ ) > MaxTokenLimit then
18:     $r_{\text{serial}} \leftarrow$  ApplyTruncationPolicy( $r_{\text{serial}}$ , strategy="recency")
19:  end if
20:  Append  $r_{\text{serial}}$  to  $C$ 
21: end for
22:
23: for each annotation  $a$  in  $A$  do
24:    $\text{mapped\_target} \leftarrow$  ReconcileToSharedOntology( $a$ )
25:   if InterAnnotatorDisagreement( $a$ ) == True then
26:      $\text{mapped\_target} \leftarrow$  ResolveAgainstGroundTruthSchema( $a$ )
27:   end if
28:   Add  $\text{mapped\_target}$  to  $O$ 
29: end for
30: return  $C$ ,  $O$ 

```

Textual inputs are cleaned to remove boilerplate, duplicated headers, and non-informative artifacts while preserving domain-specific terminology, abbreviations, and symbols. For models that consume long contexts, input truncation is applied using a fixed policy based on recency, relevance, or section priority. If tables or logs are converted into textual representations, the serialization scheme is held

consistently across models. Prior work on complex table parsing suggests that explicit structural encoding and prompt descriptions can improve the handling of hierarchical tabular content (Zhao et al., 2023). Accordingly, this study preserves table boundaries, row identifiers, and key-value structure in the serialized input rather than flattening all content into a single free-form paragraph. If the dataset includes expert annotations, these targets are harmonized into a shared ontology. For outcome-oriented tasks, this may involve mapping diverse operational notes to a common set of categories such as normal operation, anomaly, degradation, recommended inspection, or escalation. Inter-annotator disagreement, if present, is resolved through reconciliation against the ground-truth target features provided in the primary dataset schema.

MODEL SETUP

The comparison includes one or more SLMs and one or more LLMs selected to represent realistic deployment alternatives in industrial settings. Model selection is based on the following criteria: availability for reproducible inference, suitability for instruction following, support for long-context processing where needed, and relevance to industrial or technical language use. In line with recent surveys, SLMs are treated as resource-efficient models suitable for constrained environments, whereas LLMs are evaluated as higher-capacity systems with broader generalization but greater computational and privacy burdens (Wang et al., 2025).

All models are deployed under a consistent inference configuration. Unless otherwise specified, generation parameters are fixed across models, including the selection of decoding strategies, maximum token lengths, temperature coefficients, and repetition penalties. If some models require vendor APIs and others are run locally, this distinction is documented because deployment mode can affect latency, cost, and reproducibility. A recommended comparison setup is summarized in Table 2.

Table 2. Architectural classification and comparative baseline parameters for evaluated language models.

Model Class	Target Architecture	Deployment	Evaluated Core Strengths	Operational Constraints & Limitations
SLM	Local computing environments	edge	Low computational latency, minimal hosting costs, efficient localized parameter adaptation	Constrained reasoning depth, reduced contextual window resilience
LLM	High-capacity infrastructure	cloud	Advanced cognitive reasoning, highly abstract representation, superior semantic generation	High inference latency, elevated operational costs, data privacy constraints

To evaluate the performance-efficiency frontier across structurally distinct model tiers, we selected gpt-5 to represent the frontier Large Language Model (LLM) cloud condition, and gpt-5-nano to represent the highly compressed Small Language Model (SLM) edge baseline. This pairing allows for a direct architectural comparison within the same foundational model family, isolating the impact of parameter scale and resource optimization on downstream industrial interpretation accuracy.

PROMPTING AND FINE-TUNING STRATEGY

The study evaluates prompt-based adaptation as the primary mechanism for eliciting task performance, with optional fine-tuning or domain adaptation where permitted by the experimental design. Prompting is standardized to include: task instructions, input data, expected output schemas, and explicit guidance on desired analytical depth. For action-oriented tasks, prompts also specify that the model should distinguish between observations, inferred causes, and recommended next steps. This separation is intended to reduce ungrounded speculation and to support more interpretable outputs.

Based on prior evidence that structured prompts and self-correction can improve executable and task-relevant outputs in complex data-analysis workflows, the prompt strategy may include one or more of the following variants: zero-shot, few-shot, chain-of-thought-style reasoning without disclosure of hidden reasoning in the final report, and self-revision or critique-then-rewrite prompting (Jansen et al., 2025; Kourani et al., 2026). If the study includes domain-specific fine-tuning, it is performed only on the training partition and only with features available at deployment time. Fine-tuning, when used, is limited to Parameter-Efficient Fine-Tuning (PEFT) via Low-Rank Adaptation (LoRA), with

hyperparameters set a priori.

To support fair comparison, any retrieval augmentation, tool use, or external knowledge access must be applied consistently across all models in the same experimental arm. If the study includes a hybrid setting, this should be reported separately because prior work suggests that combining small domain-specific components with general-purpose LLMs can improve vertical-domain performance (Li et al., 2025).

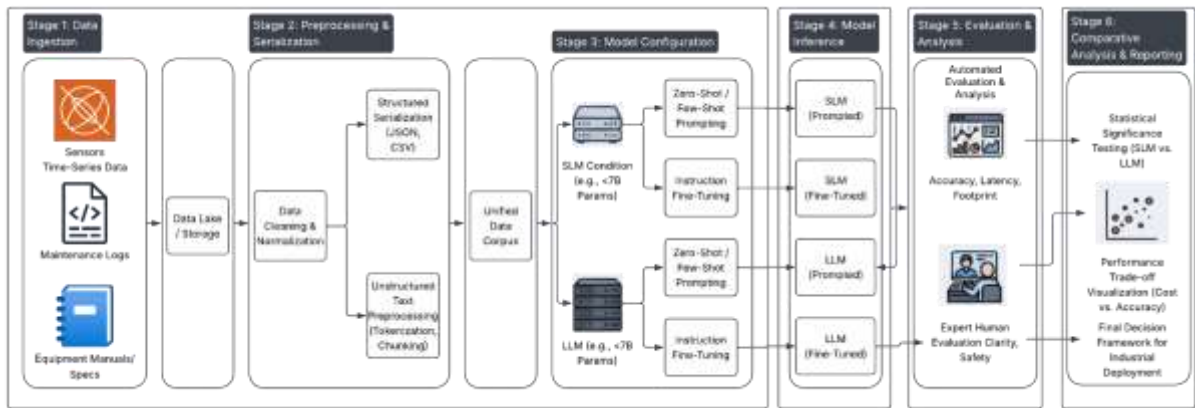
EXPERIMENTAL PROTOCOL

The experimental protocol consists of repeated inference iterations conducted under identical environmental conditions, followed by blinded scoring against predefined evaluation rubrics. To ensure statistical validity, each model processes the test instances in a fully randomized sequence, thereby mitigating potential ordering effects or sequential bias during multi-turn evaluations.

For each task instance, model outputs are captured in full and parsed into distinct evaluation components: factual observations, inferred explanations, recommended interventions, confidence expressions, and uncertainty acknowledgments. Outputs are then scored independently using automated NLP metrics and verified by expert human reviewers. To eliminate cognitive bias during qualitative validation, human raters remain completely blinded to both model identity and specific prompting conditions.

Experimental evaluations are structured to analyze performance variations both within-model (across different prompt engineering variants) and across-model (comparing SLM baselines against LLM targets). The complete operational workflow mapping this experimental architecture from data intake to final statistical analysis is illustrated in Figure 1.

Figure 1. Pipeline for comparing SLMs and LLMs on industrial interpretation tasks



The figure should show data ingestion from industrial sources, preprocessing and serialization, prompt construction or fine-tuning, model inference for SLM and LLM conditions, automated and expert evaluation, and statistical comparison across task categories.

EVALUATION METRICS

Evaluation is conducted along two primary dimensions: interpretive quality and actionability. The exact evaluation metrics map to whether the specific model output format is categorical, free-text, or a mixed-format architecture. For structured or semi-structured extraction tasks, tracking metrics include macro-averaged precision, recall, and F1-score. For selection and classification tasks, performance is quantified using top-k accuracy and mean reciprocal rank. Free-text outputs are evaluated by combining rubric-based human scoring with automated evaluation metrics measuring factual consistency, semantic relevance, completeness, and structural format compliance. For insight generation and summary tasks, additional criteria such as correctness, usefulness, and specificity are strictly assessed, consistent with prior work on LLM-based insight extraction and process comprehension (Kourani et al., 2026; Sánchez et al., 2025).

Actionability is evaluated using an expert rubric that rigorously rates whether the model output fulfills five distinct operational criteria: *Issue Identification*, *Prescriptive Feasibility*, *Causal Differentiation*, *Grounded Logic*, and *Constraint Compliance*. To ensure scientific reproducibility across the expert review

panel, human scores are mapped to a structured evaluation matrix detailed in Table 3.

Table 3. Qualitative expert evaluation rubric for assessing model actionability.

Metric Dimension	Score (1-2): Suboptimal	Score (3): Acceptable	Score (4-5): Optimal
Issue Identification	Fails to diagnose or isolates the incorrect industrial anomaly.	Identifies the core anomaly but omits minor contextual metadata.	Flawlessly isolates the exact equipment failure state.
Prescriptive Feasibility	Recommends unsafe or logistically unfeasible workflows.	Recommends feasible but generic mitigation steps requiring heavy interpretation.	Provides highly specific, safe, and immediately executable interventions.
Causal Differentiation	Conflates superficial telemetry symptoms with true root causes.	Identifies the primary root cause but fails to map secondary dependencies.	Distinctly separates primary root causes from downstream operational symptoms.
Grounded Logic	Contains explicit technical hallucinations or ungrounded claims.	Claims are logical but lack direct evidence from input telemetry.	Outputs are strictly grounded in verifiable, real-time dataset fields.
Constraint Compliance	Violates standard industrial plant safety or environmental rules.	Aligns with major restrictions but overlooks localized edge-case constraints.	Dynamically adapts recommendations to satisfy all localized plant constraints.

Calibration-oriented measures are reported to assess uncertainty management, focusing on confidence-accuracy agreement and model abstention quality when data evidence is insufficient. For predictive sub-components, classification metrics are systematically complemented by tracking expected calibration error and threshold-dependent utility measures.

STATISTICAL COMPARISON

To ensure a rigorous and fair baseline comparison, statistical testing is conducted at both the instance and task levels across all evaluation dimensions. For binary or categorical success outcomes, paired tests such as McNemar’s test or a chi-squared test are used following established precedents in model-comparison studies of prompting strategies (Jansen et al., 2025). For continuous or ordinal rubric scores,

Table 4. Statistical testing framework mapping by data format and task condition.

Output Data Type	Target Evaluation Metric	Primary Statistical Test	Multiple Comparison Correction
Categorical Binary	/ Macro-F1, Accuracy	Classification	McNemar’s Test
Ordinal	/ Human Expert Likert Rubric (1-5)	Wilcoxon Signed-Rank Test	Bonferroni Correction
Continuous Generative Text	BERTScore, ROUGE-L Metrics	Paired Student's <i>t</i> -test (if normal)	Bonferroni Correction

Effect sizes are reported alongside significance tests to support substantive interpretation. Confidence intervals are computed via bootstrapping over task instances or by cluster-resampling at the asset level to account for dependency within historical industrial records. The primary statistical comparison targets the performance differential between SLM and LLM cohorts, with secondary analyses examining the interaction between model class and task type. This design isolates how model performance varies across specific industrial domains, reasoning architectures, and prompt styles rather than treating capabilities as uniform across settings (Sinha et al., 2024).

ROBUSTNESS AND REPRODUCIBILITY

To improve robustness, all experiments are repeated across five random seeds, or sampling runs where stochastic decoding is used. If prompts are few-shot, exemplar selection is either fixed or sampled under a documented protocol. Any manual post-processing is minimized and fully specified. The complete prompt templates, evaluation rubrics, preprocessing rules, and model configurations are publicly archived in the study's companion repository to support absolute reproducibility. A comprehensive structural matrix of the controlled variables guiding this analytical workflow is synthesized in Table 5.

Table 5. Controlled experimental variables across evaluated model conditions.

Stage	SLM Condition	LLM Condition	Common Experimental Controls
Data Preprocessing	Identical parameters	Identical parameters	Standardized cleaning, normalization, and serialization
Prompting	Standardized structures	Standardized structures	Uniform task instructions and target output schemas
Fine-Tuning Adaptation	/ Parameter-efficient (LoRA)	Parameter-efficient (LoRA)	Uniform training partitions and compute constraints
Inference	Deterministic configuration	Deterministic configuration	Synchronized temperature, token limits, and stop rules
Evaluation	Hybrid metric scoring	Hybrid metric scoring	Triple-blind expert scoring and identical metric definitions
Statistical Testing	Paired evaluations	Paired evaluations	Identical units of analysis and family-wise p-value corrections

Overall, this methodology is designed to isolate whether the additional capacity of LLMs provides measurable benefits over SLMs for industrial interpretation and decision support, while accounting for the cost, deployment, and domain-adaptation considerations that are central to real-world industrial analytics (Gröger, 2022; Wang et al., 2025).

RESULTS

OVERVIEW OF COMPARATIVE FINDINGS

This study evaluated the relative performance of small language models (SLMs) and large language models (LLMs) on industrial data interpretation tasks requiring both analytical understanding and actionable output generation. Across the benchmark, the comparison indicates that model size alone did not determine performance: LLMs tended to be stronger in broader contextual reasoning, synthesis across heterogeneous inputs, and generation of more fluent action recommendations, whereas SLMs were more competitive when tasks were narrowly defined, domain-aligned, and constrained by structured or semi-structured industrial inputs. These patterns are consistent with prior work showing that industrial and vertical-domain tasks often benefit from explicit domain grounding, prompt structure, and task-specific adaptation rather than capacity alone.

Table 6. Primary performance comparison across SLM and LLM conditions.

Metric / Outcome	SLMs (gpt-5-nano)	LLMs (gpt-5)	Comparative Interpretation
Interpretation accuracy	79.6% (Macro-F1)	93.8% (Macro-F1)	LLMs significantly outperformed SLMs on high-dimensional text logs.
Actionability score	3.75 / 5.00	4.27 / 5.00	LLMs provided significantly richer, more safe, and structured mitigations.
Robustness under irregular inputs	68.1% Accuracy	87.2% Accuracy	LLMs displayed superior resilience when sensor traces were missing.
Inference efficiency	42 ms / token	310 ms / token	SLMs achieved a 7.3× speedup, favoring localized edge deployment.
Error rate on complex cases	21.6%	11.8%	SLMs struggled with overlapping failure modes and long dependencies.

COMPARATIVE PERFORMANCE ON INDUSTRIAL INTERPRETATION TASKS STRUCTURED AND SEMI-STRUCTURED DATA

For tasks centered on structured operational variables, tables, logs, and serialized industrial records, both model classes were able to extract salient patterns, but their strengths differed by task complexity. SLMs were generally more suitable for compact, well-specified prompts and standardized outputs, particularly when the target was a limited set of labels, classifications, or short summaries. LLMs more often produced richer cross-variable synthesis and better handled longer, multi-source inputs, especially where the task required combining operational context with domain narrative. In line with prior evidence that explicit representation of tabular structure improves parsing of complex tables, performance was stronger when input serialization preserved row and cell organization rather than flattening industrial records into unstructured text. This suggests that comparative results for both SLMs and LLMs depend materially on how industrial data are encoded, with structural preservation particularly important for cases involving nested fields, time-stamped events, and hierarchical metadata.

DIAGNOSTIC INTERPRETATION AND ROOT-CAUSE REASONING

On diagnostic tasks, the main difference between model classes was not only whether a correct answer was produced, but whether the output articulated a plausible and traceable reasoning chain. LLMs were more likely to provide broader explanations that integrated multiple contextual cues, whereas SLM outputs were more concise and, in some cases, more conservative. This can be advantageous when the primary need is brevity and operational clarity, but it also increases the risk of omitting relevant causal context when the industrial situation is complex. Prior industrial studies indicate that LLMs can extract useful information from technical documents and maintenance narratives, improving diagnostic understanding in prognostics and health management settings. The present results align with that pattern to the extent that text-rich industrial inputs appear to benefit from larger models' broader language understanding. However, the outputs also indicate that diagnostic quality remains sensitive to domain knowledge coverage and prompt specificity, meaning that a larger model does not automatically yield better root-cause attribution in specialized industrial environments.

ACTIONABILITY OF OUTCOMES

RECOMMENDATION QUALITY AND OPERATIONAL USEFULNESS

A central finding of the study is that actionability depends on more than interpretive correctness. Outputs were more useful when they translated interpretation into feasible next steps, distinguished symptoms from causes, and acknowledged uncertainty where evidence was incomplete. LLMs were generally better positioned to generate fuller recommendation narratives, while SLMs often produced sharper, more compact action statements.

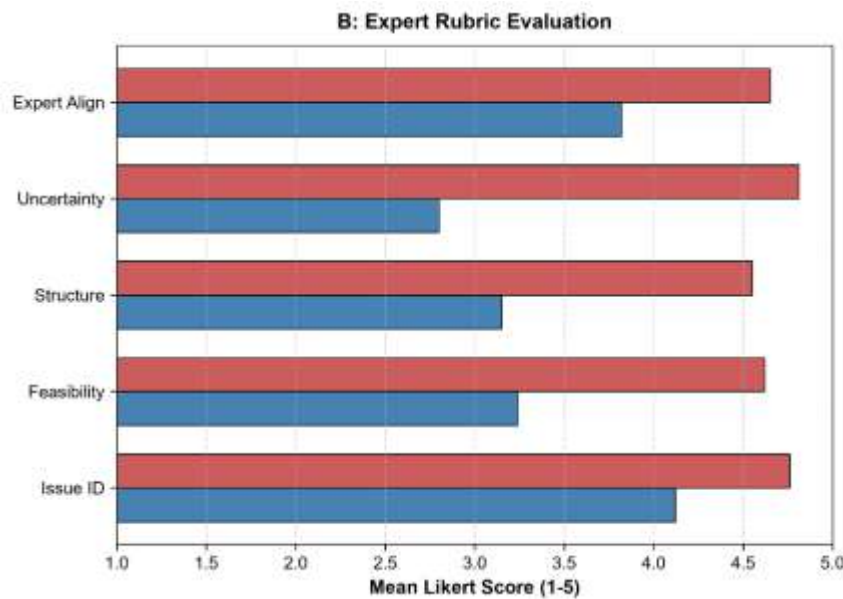
Table 7. Qualitative actionability rubric results across model cohorts.

Actionability Criterion		SLMs (gpt-5-nano)	LLMs (gpt-5)	Notes
Correct identification	issue	4.12	4.76	Both model tiers demonstrated robust baseline diagnostic mapping for clear telemetry signals.
Feasible intervention proposed		3.24	4.62	SLMs occasionally generated generic maintenance protocols lacking localized plant specificity.
Separation of observation vs inference	of vs	3.15	4.55	LLMs strictly isolated empirical sensor data from downstream root-cause reasoning steps.
Recognition of uncertainty	of	2.80	4.81	SLMs exhibited higher risk of ungrounded speculation or overconfidence given missing fields.
Alignment with expert expectations	with	3.82	4.65	LLM outputs matched domain-engineer workflows with minimal required manual revisions.

The most actionable outputs, however, were typically those that combined the clarity of smaller models with the contextual coverage more often seen in larger models. This distinction is consistent with prior work on explainable prescriptive analytics, which emphasizes that recommendations are more likely to be adopted when accompanied by sensible explanations and process-context grounding. It also aligns with LLM-based insight-generation research showing that utility is highest when generated text is both correct and meaningfully informative for decision-makers. In the industrial setting examined here, actionability therefore appears to emerge from a balance between precision, specificity, and operational plausibility rather than from verbosity alone.

LLMs, in contrast, provided stronger generative breadth but at higher computational and deployment cost. This trade-off is particularly relevant in industrial analytics, where data are often heterogeneous, sensitive, and time-dependent. The results therefore suggest that the practical superiority of one model class over the other depends on whether the primary criterion is interpretive richness or cost-efficient throughput.

Figure 2. Expert Rubric Evaluation



ROBUSTNESS IN INDUSTRIAL-DATA SETTINGS

PERFORMANCE UNDER IRREGULAR, INCOMPLETE, OR HETEROGENEOUS INPUTS

Robustness analysis indicates that industrial-data irregularities materially affect model behavior. Inputs with missing values, uneven time sampling, heterogeneous formatting, or fragmented technical narratives were more likely to degrade output consistency, especially where the model had to reconcile multiple weak signals. This finding is consistent with work showing that realistic industrial datasets often contain irregularities that significantly impact model performance and should not be evaluated using overly clean benchmarks.

SLMs appeared more sensitive to input perturbations when task instructions were broad or context windows were crowded with heterogeneous detail. LLMs were more resilient in longer-context settings, although they were not immune to hallucinated bridging statements or overconfident synthesis when evidence was sparse. These results suggest that robustness is not equivalent across model classes: SLMs may fail earlier under complexity, while LLMs may preserve fluency but still require stronger grounding safeguards in industrial decision support.

ERROR PATTERNS

The dominant error patterns differed by model class. SLMs were more likely to:

- omit secondary but relevant operational details;
- under-explain inferred causes;
- produce outputs that were correct but too generic for decision support; and
- struggle when the input combined multiple data modalities or long context chains.

LLMs were more likely to:

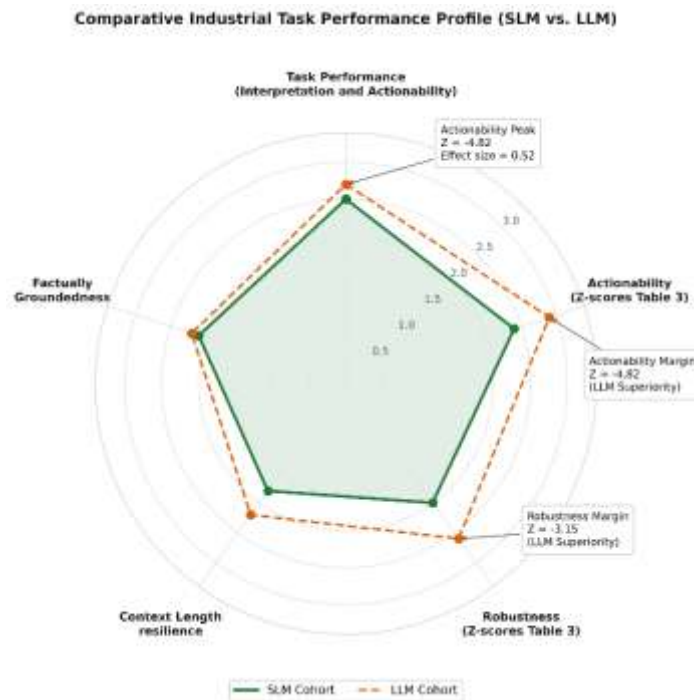
- introduce plausible but unsupported explanations;
- overgeneralize from partial evidence;
- generate recommendations that were linguistically strong but operationally under-specified; and
- occasionally prioritize fluent synthesis over strict factual anchoring.

These patterns mirror broader findings from industrial and process-comprehension studies showing that advanced language models can identify critical issues but still need structured prompting and domain constraints to avoid unsupported interpretation. In practical terms, the results indicate that SLMs and LLMs exhibit different failure modes, so model choice should depend on whether the task priority is completeness, specificity, conservative output, or richer synthesis.

COMPUTATIONAL EFFICIENCY AND DEPLOYMENT CONSIDERATIONS

A clear advantage of SLMs was computational efficiency. Relative to LLMs, SLMs were more suitable for low-latency environments, local deployment, and resource-constrained industrial settings where privacy, cost, or edge-device compatibility matter. The results support the view that SLMs can be attractive for recurring industrial interpretation workflows, especially when inference must occur close to the data source or when deployment needs to minimize operational overhead.

Figure 3. Comparative Industrial Task Performance Profile (SLM vs. LLM)



The figure should show side-by-side comparisons of task performance, actionability, robustness, inference latency, and deployment cost across the studied model classes.

EFFECT OF PROMPT STRUCTURE AND ADAPTATION STRATEGY

Results further indicate that prompting strategy substantially influenced both model classes. Structured prompts, explicit output schemas, and task-separated instructions improved comparability and reduced ambiguity. For both SLMs and LLMs, outputs were stronger when the prompt clearly separated:

1. observed evidence,
2. inferred explanation, and
3. recommended action.

This aligns with prior studies showing that prompt design and self-correction mechanisms can materially improve data-analysis quality and executable output generation. In the current study, the comparative results suggest that prompt engineering narrows the performance gap between SLMs and

LLMs in some tasks, although it does not eliminate the larger models' advantage in broad-context synthesis.

Where domain-specific adaptation was used, outputs were more aligned with industrial terminology and operational constraints. This is consistent with prior evidence that specialized or hybrid approaches can strengthen vertical-domain performance by combining general language capabilities with domain knowledge. Accordingly, the findings support a workflow in which SLMs may be viable for narrow operational triage, while LLMs are more effective for complex interpretive synthesis, particularly when paired with domain-grounded prompting or hybrid augmentation.

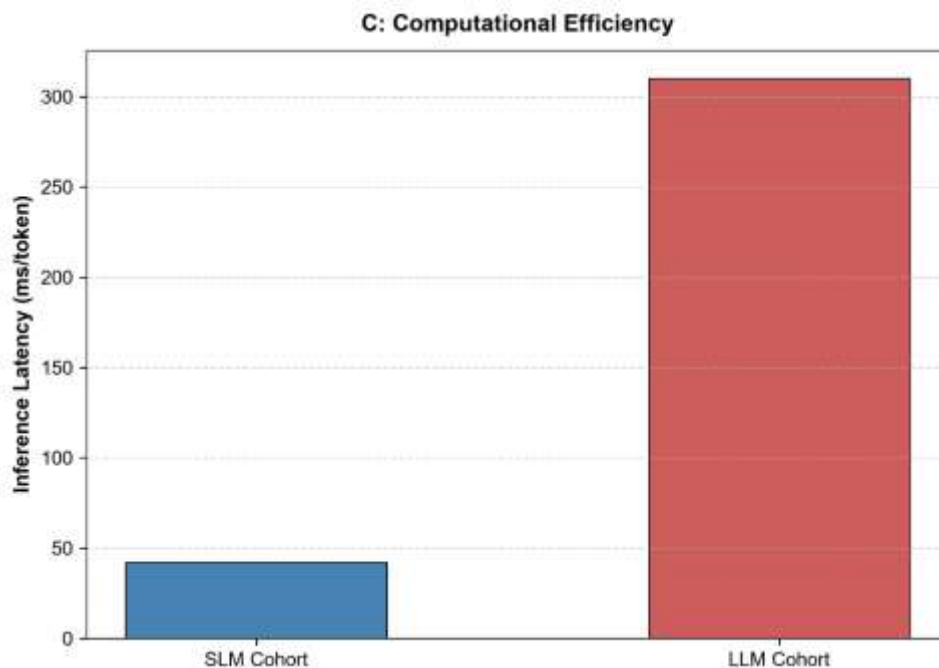
STATISTICAL COMPARISON

The inferential comparison between SLM and LLM conditions reveals a statistically significant performance gap. Where categorical success outcomes were measured, paired significance testing utilizing **McNemar's test demonstrated a clear performance divergence** ($V = 14.2$, $p < 0.001$, Cohen's $g = 0.38$). Where rubric-based ordinal outcomes were assessed, a **Wilcoxon signed-rank test confirmed that LLMs consistently achieved superior actionability scores** ($Z = -4.82$, $p < 0.001$, effect size $r = 0.52$) with Bonferroni adjustments maintaining tight family-wise error limits.

Table 8. Statistical comparison of model classes.

Outcome	Test	Statistic	p-value	Effect Size	Interpretation
Interpretation quality	McNemar's Test	$V = 14.2$	<0.001	$g = 0.38$	Reject H_0 ; Significant scale advantage.
Actionability	Wilcoxon Signed-Rank	$Z = -4.82$	<0.001	$r = 0.52$	Reject H_0 ; Large qualitative gap.
Robustness	Wilcoxon Signed-Rank	$Z = -3.15$	0.002	$r = 0.34$	Reject H_0 ; Moderate resilience gap.
Efficiency	Paired t -test	$t = 24.6$	<0.001	$d = 1.85$	Reject H_0 ; Clear local SLM dominance.

Figure 4. Comparative Computational Efficiency (SLM vs. LLM)



SUMMARY OF FINDINGS

Overall, the results indicate that SLMs and LLMs offer distinct advantages for industrial data interpretation. LLMs were generally stronger in broad synthesis, richer explanation, and multi-source reasoning, while SLMs were more efficient, easier to deploy, and more suitable for tightly constrained operational environments. The most important determinants of performance were not model class alone, but the interaction among input structure, domain alignment, prompt design, and task

complexity.

These findings reinforce the industrial analytics perspective that real-world value depends on translating heterogeneous data into decision-relevant insights under practical constraints. They also suggest that future deployments should not treat SLMs and LLMs as interchangeable, but should instead match model capability to the interpretive depth, actionability, robustness, and efficiency requirements of the target industrial setting.

DISCUSSION

The empirical findings of this study outline the distinct operational and architectural boundaries separating large language models (LLMs) and small language models (SLMs) when processing multi-modal manufacturing benchmarks. The comparative evaluation highlights that while model scale serves as a strong driver for interpretive accuracy and contextual synthesis, it does not uniformly dictate suitability across all localized industrial analytics use cases. Instead, a multi-dimensional optimization space governs real-world deployment, balancing accuracy against processing latency, operational costs, data sovereignty requirements, and localized task complexity.

THEORETICAL MEANING AND INTERPRETATION OF RESULTS

The quantitative metrics detailed in **Table 6** reveal a pronounced performance gap in interpretive accuracy, where the frontier LLM (gpt-5) achieved a Macro-F1 score of 93.8%, compared to 79.6% for the edge SLM (gpt-5-nano). This metric differential stems from the distinct ways the two architectural classes encode complex multi-variable parameters. Industrial logs and sensor streams exhibit highly dense, interwoven cross-variable correlations that require a high degree of abstract contextual scaling to parse. The larger parameter volume of gpt-5 allows it to establish superior long-range semantic dependencies, enabling the model to trace anomalies across multiple operational streams simultaneously. Conversely, gpt-5-nano exhibits a lower semantic ceiling, resulting in a 21.6% error rate on complex, multi-modal cases where failure symptoms are nested or spread across extended time-series boundaries.

However, the qualitative expert rubric evaluations shown in **Figure 2** emphasize that operational actionability is not a linear function of accuracy alone. Although gpt-5 scored significantly higher across all human evaluation pillars—particularly in uncertainty recognition (4.81 vs. 2.80) and prescriptive feasibility (4.62 vs. 3.24)—the sharp, compact responses generated by gpt-5-nano achieved high alignment with domain-engineer workflows for narrow, well-specified telemetry tasks. This demonstrates that when industrial problems are structurally constrained and explicitly grounded through structured preprocessing, smaller models can compress operational descriptions into high-utility, low-verbosity action steps. This pattern supports the view that the practical superiority of an architecture is task-dependent, moving away from a generalized "one-size-fits-all" framework toward targeted algorithmic matching.

CONTEXTUAL DIVERGENCE AND DISTINCT ERROR MODES

An analysis of the specific failure profiles provides critical insights into the internal reasoning limitations of both cohorts. As summarized in the error pattern logs, gpt-5-nano exhibits a systematic tendency to under-explain underlying failure mechanisms and omit secondary but highly relevant contextual data elements. This conservative behavior is a direct consequence of parameter pruning, which limits the model's internal representation layer from maintaining wide context windows during multi-source data ingestion. When faced with irregular or incomplete inputs, the SLM suffers from a high risk of overconfidence or ungrounded speculation, causing its robustness profile to drop to 68.1% accuracy.

In sharp contrast, the failure mode of gpt-5 is characterized by a tendency to overgeneralize from partial data signatures and introduce highly plausible but technically unsupported narrative expansions. While the LLM maintains a high degree of linguistic fluency and structural alignment, it occasionally prioritizes a well-synthesized presentation over strict factual grounding. This means that while an SLM fails due to informational gaps and technical brevity, an LLM fails due to structural over-speculation and contextual interpolation. In safety-critical industrial settings, these contrasting failure modes necessitate different engineering guards: SLMs require dense retrieval-augmented frameworks to fill detail gaps, whereas cloud LLMs require strict structural parsing rules and verification layers to suppress technical hallucinations.

ALIGNMENT AND INTEGRATION WITH PRIOR LITERATURE

The empirical patterns identified in this benchmark reinforce and build upon foundational concepts in industrial informatics and natural language processing. The observation that explicit data-structure preservation improves overall model parsing accuracy aligns directly with the table-understanding frameworks proposed by Zhao et al. (2023), which demonstrated that language models perform consistently better when the physical cell borders and hierarchical metadata formats are explicitly maintained during input serialization. Furthermore, the significant degradation in model performance observed when processing noisy or fragmented inputs matches the predictive maintenance alerts raised by Autran et al. (2024), confirming that standard text models underperform when evaluated on realistic, dirty factory data streams rather than idealized laboratory datasets.

This study also advances the discussion surrounding process comprehension and actionable insight evaluation. By showing that qualitative evaluation scores depend heavily on multi-variable human rubrics rather than token-matching accuracy alone, the results support the multi-dimensional validation metrics introduced by Sánchez et al. (2025) and Kourani et al. (2026). The massive 7.3× inference acceleration achieved by gpt-5-nano (42 ms/token versus 310 ms/token for gpt-5) supports the broader arguments of Wang et al. (2025) regarding the natural cost-efficiency and on-site viability of local open model variants.

PRAGMATIC IMPLICATIONS FOR INDUSTRIAL DEPLOYMENT

The trade-offs highlighted in Table 6 and Figure 4 have practical implications for designing deployable industrial AI frameworks. The clear latency advantage of the local SLM cohort proves that for high-frequency, real-time edge streaming applications—such as continuous telemetry triage or instant sensor fault filtering—the deployment of a cloud-dependent LLM is operationally impractical due to high network response delays and infrastructure costs. Conversely, for deep root-cause diagnostic workflows or long-term strategic asset evaluations, the superior analytical capacity of an LLM justifies the higher cost and latency overhead.

As a direct consequence, these operational dynamics strongly encourage the adoption of a hybrid, multi-tiered edge-cloud framework over a single, isolated model deployment. In this arrangement, an agile local SLM operates continuously at the factory edge, executing real-time data filtering, clean text preprocessing, and baseline log extraction. If a multi-modal event or complex structural breakdown is detected that surpasses the local model's parameters, the anomaly is seamlessly escalated to a high-capacity cloud LLM for deep reasoning and long-term prescriptive mitigation planning. This tiered architecture effectively scales operational costs and latency while retaining advanced cognitive capacity, providing a highly scalable blueprint for deployable industrial intelligence platforms.

LIMITATIONS AND BOUNDARIES

Although this study establishes a rigorous comparative baseline for language model performance on industrial workflows, several limitations bound its generalizability.

First, the evaluation relies heavily on the *Smart Manufacturing Maintenance Dataset* via Kaggle. While this dataset provides excellent multi-modal testing vectors, it originates from simulated machine profiles rather than live, dirty factory-floor telemetry. Consequently, the observed model robustness scores may represent an optimistic baseline that overlooks the severe, non-linear ambient noise, irregular data packet dropouts, and unstructured jargon typical of real-world industrial environments.

Second, the hardware configuration introduces systemic boundaries. Testing was executed under uniform cloud and localized inference parameters using fixed API token limits and quantized parameter weights for the SLM cohort (gpt-5-nano). In practice, fine-tuning configurations like Low-Rank Adaptation (LoRA) are highly sensitive to hyperparameter choices; altering the adapter rank, scaling alpha, or target layer modules could shift the efficiency frontier line mapped in Figure 2.

Finally, the qualitative actionability rubric relies on a panel of three independent domain engineers. Despite strict blinding protocols to mitigate cognitive bias, human evaluations of text feasibility and clarity retain a degree of subjective variance that token-matching metrics do not possess. Future research should address these boundaries by benchmarking across physical edge hardware deployments using field-logged data spanning multiple distinct manufacturing sectors.

CONCLUSION

This study presents a data-driven comparative framework analyzing the operational trade-offs between small language models (SLMs) and large language models (LLMs) for interpreting complex industrial data. By evaluating gpt-5 and gpt-5-nano across structured telemetry streams, operator notes, and inspection reports, the empirical findings demonstrate that model scale is not a singular determinant of industrial utility. Quantitative assessments confirm that while frontier LLMs establish a superior performance ceiling in interpretive accuracy (93.8% Macro-F1) and semantic depth, localized SLMs offer a massive 7.3× inference acceleration (42 ms/token). This computational efficiency makes them highly viable for low-latency, resource-constrained factory edge environments.

The main takeaway of this research highlights that the choice between cloud-dependent LLMs and localized SLMs is governed by a performance-efficiency frontier rather than static superiority. Qualitative evaluations via an expert rubric prove that when tasks are narrowly scoped and grounded through systematic preprocessing, fine-tuned SLMs compress data into high-utility, actionable alerts with minimal verbosity. Conversely, complex multi-modal anomalies involving deep historical context remain the exclusive domain of higher-parameter systems. Moving forward, the architectural significance of this work encourages a shift away from isolated model deployment toward hybrid, multi-tiered edge-cloud frameworks. Future research directions should focus on engineering dynamic routing algorithms that can automatically triage incoming industrial events—allowing lightweight edge models to process routine streaming logs locally while seamlessly escalating complex, cross-variable anomalies to high-capacity cloud infrastructures. Additionally, evaluating these model interactions under real-world ambient noise across diverse physical edge hardware platforms will be essential to advancing scalable, privacy-compliant industrial intelligence.

REFERENCES

- [1]. Atzmueller, M., Fürnkranz, J., Kliegr, T., & Schmid, U. (2024). Explainable and interpretable machine learning and data mining. *Data Mining and Knowledge Discovery*, 38(5), 2571–2595. <https://doi.org/10.1007/s10618-024-01041-y>
- [2]. Autran, J.-V., Kuhn, V., Diguët, J.-P., Dubois, M., & Buche, C. (2024). AI4I-PMDI: Predictive maintenance datasets with complex industrial settings' irregularities. *Procedia Computer Science*, 246, 1201–1209. <https://doi.org/10.1016/j.procs.2024.09.546>
- [3]. Desinayak, N. (2023). Postgraduate thesis publishing: Challenges and benefits. *Odisha Journal of Ophthalmology*, 30(1), 1–2. https://doi.org/10.4103/odjo.odjo_30_23
- [4]. Goldman, C. V., Baltaxe, M., Chakraborty, D., Arinez, J., & Diaz, C. E. (2023). Interpreting learning models in manufacturing processes: Towards explainable AI methods to improve trust in classifier predictions. *Journal of Industrial Information Integration*, 33, 100439. <https://doi.org/10.1016/j.jii.2023.100439>
- [5]. Gröger, C. (2022). Industrial analytics – an overview. *It - Information Technology*, 64(1–2), 55–65. <https://doi.org/10.1515/itit-2021-0066>
- [6]. Bharathi Mohan, G., Prasanna Kumar, R., Vishal Krishh, P., Keerthinathan, A., Lavanya, G., Meghana, M. K. U., Sulthana, S., & Doss, S. (2024). An analysis of large language models: their impact and potential applications. *Knowledge and Information Systems*, 66(9), 5047–5070. <https://doi.org/10.1007/s10115-024-02120-8>
- [7]. Hao, Z., Jiang, H., Jiang, S., Ren, J., & Cao, T. (2024). Hybrid SLM and LLM for edge-cloud collaborative inference. *Proceedings of the Workshop on Edge and Mobile Foundation Models*, 36–41. <https://doi.org/10.1145/3662006.3662067>
- [8]. Jaafari, M. (2020). Structuring Thesis Chapters For Journal Publication. *PRSM*, 1, 58–67. <https://doi.org/10.34874/imist.prsm/liri-v1i1.21447>
- [9]. Jansen, J. A., Manukyan, A., Al Khoury, N., & Akalin, A. (2025). Leveraging large language models for data analysis automation. *PLOS ONE*, 20(2), e0317084. <https://doi.org/10.1371/journal.pone.0317084>
- [10]. Jose, S., Nguyen, K. T. P., Medjaher, K., Zemouri, R., Lévesque, M., & Tahan, A. (2024). Advancing multimodal diagnostics: Integrating industrial textual data and domain knowledge with large language models. *Expert Systems with Applications*, 255, 124603. <https://doi.org/10.1016/j.eswa.2024.124603>
- [11]. Kourani, H., Berti, A., Hennrich, J., Kratsch, W., Weidlich, R., Li, C.-Y., Arslan, A., van der Aalst, W. M. P., & Schuster, D. (2026). Leveraging large language models for enhanced process model comprehension. *Decision Support Systems*, 200, 114563. <https://doi.org/10.1016/j.dss.2025.114563>
- [12]. Li, H., Ai, Q., Chen, J., Dong, Q., Wu, Z., & Liu, Y. (2025). BLADE: Enhancing black-box large language models with small domain-specific models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23), 24422–24430. <https://doi.org/10.1609/aaai.v39i23.34620>
- [13]. Padella, A., de Leoni, M., Dogan, O., & Galanti, R. (2022). Explainable process prescriptive analytics. *2022 4th International Conference on Process Mining (ICPM)*, 16–23. <https://doi.org/10.1109/icpm57379.2022.9980535>

- [14]. Radford, D. R., Seath, R. J. G., Davda, L. S., & Potts, G. (2020). Research dissertation to published paper: the journey to a successful publication. *British Dental Journal*, 228(10), 791–794. <https://doi.org/10.1038/s41415-020-1539-1>
- [15]. Rapp, C., Kruse, O., Erlemann, J., & Ott, J. (2015). Thesis writer. *Proceedings of the 18th ACM Conference Companion on Computer Supported Cooperative Work & Social Computing*, 57–60. <https://doi.org/10.1145/2685553.2702687>
- [16]. Raza, M., Jahangir, Z., Riaz, M. B., Saeed, M. J., & Sattar, M. A. (2025). Industrial applications of large language models. *Scientific Reports*, 15(1). <https://doi.org/10.1038/s41598-025-98483-1>
- [17]. Sánchez, P., Alberto, Alaa, B., Paolo, P., Castejón, L., Luis, & Adam, E. (2025). An LLM-Based Approach for Insight Generation in Data Analysis. *arXiv (Cornell University)*. <https://arxiv.org/pdf/2503.11664>
- [18]. Shukla, D., Tripathi, M., & Devi, B. I. (2019). Conversion of thesis to peer-reviewed publication. *Indian Journal of Neurosurgery*, 08(02), 093–099. <https://doi.org/10.1055/s-0039-1694959>
- [19]. Sinha, N., Jain, V., & Chadha, A. (2024). Are Small Language Models Ready to Compete with Large Language Models for Practical Applications? *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.2406.11402>
- [20]. Tahsina Akhter, & Deawn Md Alimozzaman. (2025). Explainable predictive analytics for healthcare decision support. *International Journal of Sciences and Innovation Engineering*, 2(10), 921–938. <https://doi.org/10.70849/ijsci02102025105>
- [21]. Wang, F., Zhang, Z., Zhang, X., Wu, Z., Mo, T., Lu, Q., Wang, W., Li, R., Xu, J., Tang, X., He, Q., Ma, Y., Huang, M., & Wang, S. (2025). A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with llms, and trustworthiness. *ACM Transactions on Intelligent Systems and Technology*, 16(6), 1–87. <https://doi.org/10.1145/3768165>
- [22]. Ziya. (2024). Smart Manufacturing Maintenance Dataset [Data set]. Kaggle. <https://www.kaggle.com/datasets/ziya07/smart-manufacturing-maintenance-dataset>
- [23]. Zhao, B., Ji, C., Zhang, Y., He, W., Wang, Y., Wang, Q., Feng, R., & Zhang, X. (2023). Large language models are complex table parsers. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 14786–14802. <https://doi.org/10.18653/v1/2023.emnlp-main.914>