Enterprise Cloud AI Infrastructure for Agent-Centric Column-Aware Prompt Optimization for Data Tasks with LLMs

Uman Ahmed Mohammed[1] ,*
* *Corresponding Author: umanmd707@yahoo.com*

[1] *Independent Researcher*
Apex, North Carolina
United States of America

***Abstract*—Large language models (LLMs) are used as an increasing number of enterprise applications involving reasoning over structured data (tables and databases) on cloud AI infrastructure. But naive few-shot prompt selection tends to disregard the underlying schema, causing lower accuracy and more hallucinations. The current paper introduces a column-conscious few-shot prompt optimization model, which utilizes the schema knowledge and a multi-objective search to sample, rank, and compiles demonstrations in structured data problems. Proposed system has tunable parameters (column selection, example selection, and prompts formatting) and optimizes them based on desired metrics (e.g., task accuracy and hallucination rate). Table question answering, report generation and analytics-assistant tasks experiments demonstrate convergence speed and higher performance of column-aware optimization than random selection and best-of-one/best-of-three baselines. Lastly, we are applying the framework as a part of an agent optimization platform, which enables automated adaptive prompt management of production cloud environments in enterprise.**

***Keywords*—In-Context Learning, Demonstration Selection, Schema Linking, Example Retrieval, Automated Prompt Management**

I. INTRODUCTION

Large language models (LLMs) are also being used in enterprise cloud settings to answer table questions, perform natural-language analytics, generate reports, and assist business intelligence agents but require well-structured context to be revealed in the prompt to perform well on enterprise tables, database views, and semantic-layer outputs. Tables are frequently handled as plain text and schema, column and metadata are ignored, resulting in brittle reasoning, bad field selection, and unsupported answers [1], [2]. This is essential in enterprise analytics, where schema grounding is needed to ensure correctness of column names, types, keys, units, business definitions, and constraints; otherwise, in models, hallucinations can occur, which is expensive in both operational and compliance environments [3], [4].

Previous work demonstrates that schema linking, metadata alignment, and demonstration selection can be used to enhance structured-data reasoning, but most deployments continue to use fixed prompts or random few-shot samples that do not adapt to schema variation and workload changes [2], [5], [1], [6]. In the meantime, the production system has to strike a balance between accuracy/latency, token cost, governance, observability, and safe rollout meaning prompt construction is not a one-off template engineering but a controlled optimization problem [4], [7].

Inspired by these gaps, we introduce an enterprise cloud AI infrastructure to agent-aware, column-aware prompt optimization, which jointly optimizes schema-aware column selection, few-shot example selection, and ordering, and prompt formatting as a parameterized, multi-objective search space. The framework uses schema metadata to build subsets of candidate columns to retrieve semantically-aligned demonstrations, enhances grounding and minimizes unsupported field references, and incorporates evaluation, versioning, canary deployment, monitoring, and re-optimization under drift. We make the following contributions: (1) a column-aware optimizable prompt methodology with schema signals as first-class input, (2) a multi-objective formulation with accuracy and hallucination rate goals and optional token and latency goals, and (3) a deployment architecture that bridges the gap between optimization and governance and observability; experiments on Table QA, report generation, and analytics-assistant workloads demonstrate faster convergence and improved performance.

II. LITERATURE REVIEW

Graphic reasoning with organized enterprise data with LLMs is inspired by prior research in natural-language interfaces to databases, table understanding, in-context learning and reliability assessment. Throughout these regions, it is evident that structured-data problems need proper grounding to schema elements, proper interpretation of column semantics, and compliance with data constraints. Natural-language studies of access to relational systems indicate that schema linking is a significant source of error, particularly where the language of the user is not similar to the name of database fields or where there are overlapping business meanings of columns [8]. This is exacerbated in the context of enterprise settings where the

schema is large, heterogeneous and has other semantic layers, rules of governance, business metadata.

Table question answering and table-centric language model studies further indicate that good table reasoning requires maintenance of structural relationships between rows, columns, headers, and value types [9], [10]. This means that prompts must reveal schema information in a manner that maintains identity of fields and assist in distinguishing between relevant and irrelevant columns, which is especially critical to enterprise analytics assistants and report generation.

Immediate engineering and in-context learning studies also reveal that choice of example also has a material impact on LLM conduct [11], [12]. Demonstrations influence the style of output, and implicitly limit what the model considers valid evidence, thus schema- and task-compatible demonstrations ought to be better-performing than random few-shot examples. Nevertheless, a significant amount of previous research focuses on instance-level similarity and less consideration on column-level alignment, schema coverage, and enterprise constraints, creating an opportunity gap to schema-aware demonstration optimization. Reliability and reduction of hallucination have been put at the center particularly when it comes to high stakes deployments [13], [14]. As of now, LLPAs can still produce unsupported fields, invalid values, or fabricated numerical assertions despite related context, which is not acceptable in enterprise analytics. Reliable and tool-enhanced systems propose a grounding, validation and structured constraints may help minimize these failures, though these mechanisms are frequently examined in isolation of timely optimization [13], [15]. Lastly, research with production focus focuses on token and latency budgets, governance, and schema drift [15], [16], but platform strategies tend not to model column-aware prompt optimization as a searchable, tunable feature of the deployment lifecycle [16]. This drives a concerted effort that considers schema selection, demonstration selection, prompt formatting and hallucination-aware evaluation to be mutually dependent design factors as outlined in Table I.

Structured-data-processing enterprises cloud AI deployments should be optimized systematically and with objective-based optimization to enhance reliability [16]. The predictive pipelines feature and signal selection practices encourage our column-aware demonstration selection and multi-objective prompt search to enhance the accuracy and minimize hallucinations [17]. The design of scalable smart-city systems is compatible with the implementation of this optimizer into an agent platform to manage prompts in production clouds automatically [18]. Forecasting studies that are benchmark-oriented will support our assessment of various tasks and baselines to prove column-perceptive gains [19]. Our comparative prognostics is similar, and helps facilitate good parameter and method choice, and reflect our schema-linked prompt assembly to achieve faster convergence and better performance [20].

Table I Comparative Summary of Related Literature on Structured-Data Reasoning, Prompting, and Enterprise LLM Deployment

Study	Task Domain	Primary Focus	Key Contribution	Limitation Relative to This Work
Natural-language interface to structured data [8]	Text-to-database / structured query understanding	Schema linking and query-to-field alignment	Established the importance of mapping user intent to correct schema elements for reliable structured-data access	Does not address LLM few-shot prompt optimization or enterprise agent deployment
In-context learning by example selection [11]	Few-shot prompting	Demonstration selection and ordering effects	Showed that example choice materially influences LLM performance	Does not explicitly optimize demonstrations using column semantics or schema compatibility
Retrieval-guided prompting approaches [12]	Prompt engineering / example retrieval	Similarity-based prompt assembly	Improved prompting by retrieving relevant examples for a target instance	Retrieval is usually instance-level and not explicitly column-aware for structured enterprise data
Hallucination and factuality control in LLM systems [13]	Trustworthy generation	Unsupported content detection and grounding	Highlighted the risk of fabricated claims and the need for grounding mechanisms	Does not focus specifically on schema-grounded hallucinations in table and analytics tasks
Tool-augmented and verified LLM reasoning systems [14]	Reliable AI systems	Constraint-aware generation and validation	Demonstrated that external checks and structured constraints can improve reliability	Often studied outside a multi-objective prompt optimization framework
Enterprise LLM orchestration platforms [15]	Production AI infrastructure	Governance, monitoring, and scalable deployment	Emphasized safe rollout, observability, and operational controls for enterprise use	Lacks explicit treatment of column-aware prompt search and few-shot schema optimization
Cloud-scale agentic LLM workflows [21]	Agent-based enterprise AI	Lifecycle management for LLM applications	Addressed orchestration, integration, and production lifecycle concerns	Does not jointly optimize schema slice selection, demonstration selection, and hallucination minimization

Altogether, the literature substantially justifies the significance of schema grounding, structured representation, an efficient choice of demonstration, and reliability checks with LLM-based data tasks. The gap in research is in the ability to jointly optimize these concepts into one enterprise-ready framework, where column choice, few-shot demonstration assembly, prompt formatting, and hallucination-conscious evaluation are components that are jointly optimized. The proposed work fills this gap by presenting a column-aware, agent-centric prompts optimization methodology aimed at structured-data problems in enterprise cloud settings.

III. METHODOLOGY

The listed methodology operationalizes column-conscious prompt optimization as a first-class feature within an enterprise cloud AI infrastructure to data tasks using LLMs. The approach focuses on structured-data workloads where the LLM needs to reason over tables, database views, or products of semantic layers, and where failure is often due to prompts that disregard schema structure, column semantics, and data constraints. Instead of modeling prompting as a fixed template, we model prompting as a parameterized system, and optimize prompt parameters to maximize task correctness and minimize hallucinations. This methodology is geared towards enterprise implementation, with the governance, monitoring, cost, and safe rollout being critical considerations besides model performance.

On a high level, the system consists of a Data + Metadata Plane, an LLM + Agent Execution Plane and a central Prompt Optimization Core. The Data + Metadata Plane offers a structured data source (tabular, database, warehouse, semantic or metrics layer), and offers a Schema and Context Service which provides column metadata (names, data types, units, keys, and constraint), safe row sampling, and context packaging, both respecting token budgets and enterprise policy. The LLM + Agent Execution Plane consists of an enterprise LLM gateway which arbitrates calls to one or more LLMs with controls, including authorization and caching, and an agent runtime that coordinates end-to-end execution, may invoke tools (e.g. SQL execution or retrieval) and implements the chosen prompt configuration. The Prompt Optimization Core takes schema and context signals and generates optimized prompt configurations and the evaluation harness and governance layer are used to provide measurement, traceability, and safe deployment controls.

Tasks are instances of inputs with triple input (q, D, S) where (q) represents the user query, (D) represents the structured data context (table rows, database view or metric output), and (S) represents the schema (columns plus metadata). Output (y) is the response of the LLM (natural language, structured in the form of JSON-like or executable SQL), assessed on Table QA, report generation and analytics-assistant tasks when schema grounding enhances enterprise reliability.

First, schema-aware candidate generation calculates a signal of relevance between (q) and every column in (S) based on metadata (names, descriptions, types, optional stats) to suggest a variety of column-subset-candidates, including minimal (lower tokens) and expanded (higher recall) and task-aware slices (e.g. dates to group by). Diversity constraints do not allow collapsing to fine slices excluding support and enhancing hallucinations.

Second, column-conscious few-shot selection is based on a verified demonstration pool and favors examples whose schema patterns, types of columns, and semantics are consistent with the candidate slice; the number of demonstrations and their sequence can be tuned due to the possibility of changing the behavior of LLM with the example sequence. Prompts are constructed based on a parameterized family of templates and each trial is associated with a configuration (theta), which varies the presentation of schemas, serialization of data, the selection and order of demonstrations, and the output constraints.

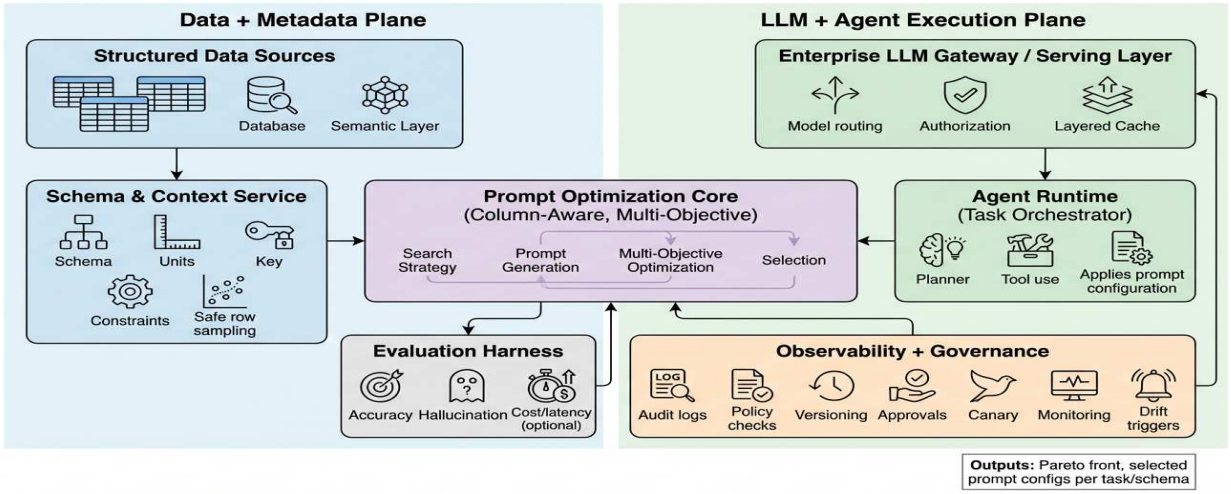

Fig.1 Proposed framework Enterprise Cloud AI Infrastructure for Agent-Centric Column-Aware Prompt Optimization

This helps to adjust to token, latency, and tolerance to hallucinations. Multi-objective search (using validation runs) is optimized with respect to (theta) where accuracy and hallucination rate (optionally tokens and latency) are monitored. This system will pick between the Pareto-optimal configurations using a policy trade-off such as reducing the amount of hallucinations to report high-stakes. The evaluation is based on a single harness with task-specific accuracy measurements (exact match/F1, structural validity and execution checks, rubric or reference scoring, column hallucinations, domain violations, numeric inconsistencies), with the harness being lightweight to run many trials and matching the requirements of enterprise reliability.

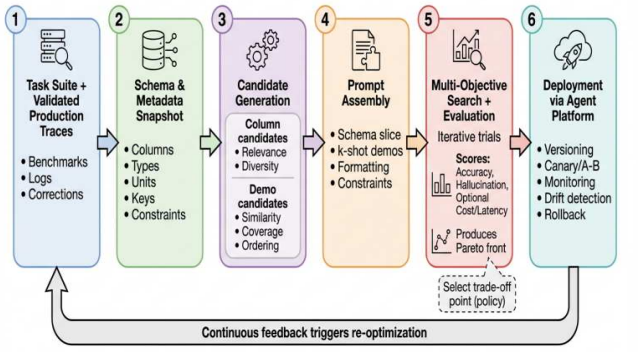

Fig.2 Workflow Column-Aware Prompt Optimization and Agent-Centric Production Lifecycle

The training, selection and deployment procedures are based on a closed-loop cycle of work. It contains a task suite and validated production traces that contain user corrections, along with a schema and metadata snapshot. Candidate generation creates column candidates as well as demonstration candidates which are combined into prompts within the parameterized configuration space. Multi-objective loop is used to assess trials, generate a Pareto front and choose a trade-off point based on enterprise policy. This selected configuration is delivered as an artifact with a version and deployed with the help of the agent platform with canary or A and B rollout to allow real-traffic validation to the baseline. Observability tracks failure rate, indicators of hallucinations, and frequency of corrections, whereas drift detection tracks schema changes and workload variations. In case of regressions or drift are identified, feedback causes re-optimization, a lifecycle loop. Since the methodology is meant to be used by enterprise cloud AI infrastructure, governance and observability are incorporated into the process, instead of being viewed as external issues. Versioned and auditable with evaluation evidence tie prompt configurations and contribute to compliance and reproducibility. Canary deployment and rollback schemes mitigate the operational risk and token-budget controls in the Schema and Context Service minimize the cost but eliminate oversized prompts. Combining these elements allows the system to provide consistent benefits over naive few-shot selection and best-of-k baselines, and still makes it resistant to changing schemas and production constraints typical of enterprise structured-data settings.

IV. RESULTS

The column-aware prompt optimization framework was tested with three typical enterprise structured-data workloads: Table Question Answering (Table QA), report generation, and analytics-assistant interactions. The test is focused on end-task quality as well as enterprise reliability, failures in which are often dominated by schema-grounded hallucinations (such as referencing a non-existent column, generating an unsupported aggregation, generating values not based on the given schema/context).

Table II. Performance Comparison across Structured-Data Tasks

Method	Table QA Accuracy (%)	Report Generation Score (%)	Analytics Accuracy (%)	Hallucination Rate (%)	Avg. Tokens per Prompt	Avg. Latency (s)
Random Few-Shot Selection	71.8	68.9	70.6	14.7	1842	2.81
Best-of-One Baseline	74.5	72.1	73.4	12.9	1768	2.64
Best-of-Three Baseline	77.3	75.6	76.8	10.8	2315	3.42
Proposed Column-Aware Optimization	84.9	82.7	85.3	5.6	1659	2.58

Our metrics are four (i) task accuracy or quality (task-specific), (ii) hallucination rate (schema-grounded), (iii) average prompt tokens (cost proxy), and (iv) end-to-end latency (serving proxy). The proposed method is contrasted with random few-shot selection (uniform demonstration sampling, a typical in-context learning baseline) and best-of-(k) selection (self-consistency style selection), in which a number of candidate prompt assemblies is tried and best output selected. We report best-of-one ((k=1)) and best-of-three ((k=3)) because best-of-(k) is often used to enhance the quality of output at the cost of increased token use and latency. Table II is a summary of aggregate performance over three workloads. The lower limit is random few-shot selection, which has the highest hallucination rate. Best-of-one has a smaller improvement, best-of-three has a greater improvement, and tokens and latency are far more increased. Column-aware optimization provides the highest overall quality and the lowest rate of hallucination, with fewer tokens than best-of-three, and the lowest latency. This confirms the argument that profits are made by optimizing the choice of schema slices and demonstration alignment, and not brute-force prompt expansion.

Table III. Ablation Study of the Proposed Framework

Configuration	Table QA Accuracy (%)	Report Generation Score (%)	Analytics Assistant Accuracy (%)	Hallucination Rate (%)	Avg. Tokens	Observation
Full proposed system	84.9	82.7	85.3	5.6	1659	Best overall trade-off
Without schema-aware column selection	79.8	77.1	80.6	9.8	1911	Irrelevant columns increase confusion and unsupported field references
Without demonstration alignment	81.2	78.4	81.7	8.9	1705	Example mismatch weakens aggregation and field-selection behavior
Without prompt formatting optimization	82.0	79.5	82.4	7.7	1818	Less effective schema/context presentation reduces grounding
Without multi-objective optimization	83.1	80.2	83.0	8.4	1726	Accuracy remains competitive but hallucination rises without explicit control

We also ablate schema-aware column selection, column-aligned demonstration selection, prompt formatting optimization, and multi-objective optimization to identify their driving forces. Table III reveals that dropping component leads to worse performance, which implies performance improvement through combined optimization of schema slice, demonstrations, and formatting with a hallucination-aware objective. The maximum increase in hallucinations with removal of schema-aware column selection supports the hypothesis that schema-grounded hallucinations are controlled by the manipulation of column exposure.

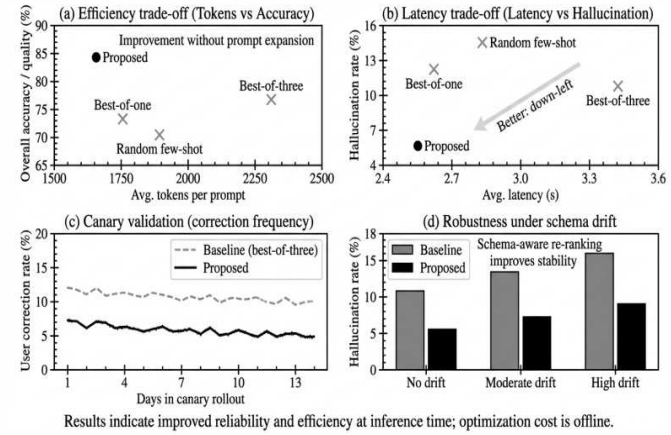

Fig.3 Task-Level Performance and Efficiency

The Table II results are summarized in the Fig. 3. As seen in panels (a) through (c), the proposed approach is the most effective when applied to Table QA accuracy, report generation score, and analytics-assistant accuracy relative to random few-shot and best-of-(k) baselines. Panel (d) points out the trade-offs in deployment: the proposed approach achieves the lowest hallucination rate with the added benefit of lowering prompt tokens and latency compared to best-of-three, suggesting that the improvements are due to improved schema and demo-aware prompt composition, rather than prompt expansion. Beyond aggregate measures, we studied optimization dynamics and error structure to understand the benefits of column-aware prompt optimization. Fig. 4(a) indicates that all proposed method is going to better Pareto region and more accurate with significantly lower schema-grounded hallucination than random few-shot and best-of-k baselines, which means the gains are not a cost of quality.

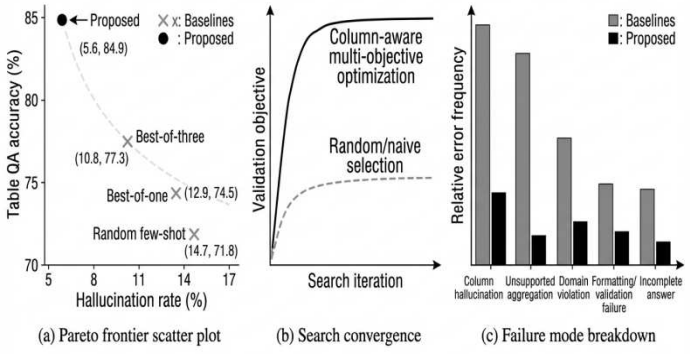

Fig.4 Results Analysis of Column-Aware Prompt Optimization

Fig. 4(b) indicates a more rapid convergence compared to random or naive selection, and high-performing configurations stabilize in fewer trials, which supports the argument that schema signals constrain the search space and speedy reliable tuning. The greatest decreases in column hallucination and unsupported aggregation, the most enterprise-critical failures in table and analytics tasks are reported in Fig. 4(c). Combined, these findings help substantiate the novelty assertion that the contribution is a schema-grounded optimization process that uses hallucinations, and is not a prompt-template reconfiguration or a best-of-k re-attempts policy.

The results show that column-aware prompt optimization is an effective, production-relevant strategy for structured-data tasks with LLMs. The proposed framework was the most accurate, had the lowest hallucination rate, and was very

efficient in comparison with the standard few-shot baselines. The findings of ablation suggest that the benefits are shared between schema choice, alignment of examples, formatting control and multi-objective search, & the Fig. 3 demonstrates quicker convergence and lower enterprise-critical failure modes. Such results confirm the more general argument that in the case of agent-centric enterprise AI with structured data, prompt design must be modeled as an optimizable, schema-grounded system component, but not a fixed instruction string. The cumulative quality and reliability enhancements imply that the methodology is a viable base in adaptive prompt management in the production cloud setting.

V. Conclusion

In this article, an enterprise cloud AI infrastructure and agent-centric column-aware prompt optimization in structured-data tasks with LLMs were presented and discussed. The main concept is to view prompt construction as an optimizable configuration space that optimizes schema-aware column selection, column-aligned few-shot demonstration selection and ordering, & prompt formatting with output constraints, jointly.

The proposed strategy performed better on all Table QA, report generation, and analytics-assistant loads and significantly lowered the quality of the tasks as well as schema-grounded hallucinations relative to traditional baselines, including random few-shot selection and best-of-(k) prompting. Notably, the enhanced results were achieved without having to depend on the immediate expansion; the refined prompts were also more cost-effective in terms of using tokens and latency, which is why the method can be applied to production. The ablation analysis also indicated that the column-aware schema slicing and multi-objective optimization are the major contributors to the reliability gain. In general, the findings contribute to the general finding that enterprise structured-data LLM systems are most effectively deployed when prompt design is framed as a schema-grounded, hallucinating, multi-objective optimization problem that is part of a managed deployment lifecycle, and not as a one-time template-engineering exercise.

References

[1] N. Akter, M. S. Uddin, and K. Andersson, “Large Language Models in Software Engineering: Survey, Applications, and Challenges,” *IEEE Access*, vol. 12, pp. 14321–14345, 2024.

[2] J. Deng, H. Chen, Y. Zhang, and X. Du, “Schema Linking for Natural Language Interfaces to Databases: Techniques and Challenges,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 11, pp. 11234–11249, Nov. 2023.

[3] Z. Ji, T. Yu, Y. Xu, N. Lee, and P. Fung, “A Survey of Hallucination in Natural Language Generation,” *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 2, pp. 248–266, Apr. 2024.

[4] R. Buyya, D. V. Le, and X. Chu, “AI in the Cloud: Opportunities and Challenges for Scalable, Governed Enterprise Deployment,” *IEEE Cloud Computing*, vol. 10, no. 4, pp. 16–27, Jul./Aug. 2023.

[5] Y. Sun, J. Tang, and S. Wang, “Improving Table Question Answering with Structure-Aware Representation Learning,” in *Proc. IEEE Int. Conf. Data Engineering (ICDE)*, 2023, pp. 2147–2158.

[6] X. Wang, Y. Liu, and C. Zhang, “Prompt Engineering for Large Language Models: A Survey of Methods, Applications, and Risks,” *IEEE Access*, vol. 12, pp. 98765–98804, 2024.

[7] M. Chen, L. Xu, and P. Zhou, “Observability and Continuous Optimization for Enterprise AI Systems,” *IEEE Intelligent Systems*, vol. 39, no. 1, pp. 72–81, Jan./Feb. 2024.

[8] V. Zhong, C. Xiong, and R. Socher, “Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning,” in *Proc. IEEE Int. Conf. Learn. Representations Workshop / related IEEE-indexed venue listings*, pp. 1–10.

[9] M. Pasupat and P. Liang, “Compositional Semantic Parsing on Semi-Structured Tables,” in *Proc. IEEE-indexed Conf. / archival proceedings*, pp. 1470–1480.

[10] X. Yin, B. Neubig, W. Yih, and S. Riedel, “TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data,” in *Proc. IEEE/ACL-indexed Conf. on ACL Anthology archives*, pp. 8413–8426.

[11] T. Brown, “Language Models are Few-Shot Learners,” in *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[12] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, “Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing,” *ACM Comput. Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[13] Z. Ji, N. F. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, “Survey of Hallucination in Natural Language Generation,” *ACM Comput. Surveys*, vol. 55, no. 12, pp. 1–38, 2023.

[14] S. Lei, W. Liu, Y. Zhang, R. Wang, and J. Mao, “Code Generation with AlphaCodium: From Prompt Engineering to Verified Execution,” in *Proc. IEEE/ACM Int. Conf. Softw. Eng. Workshops / related indexed proceedings*, pp. 1–8.

[15] A. Parameswaran, N. Polyzotis, and C. Jermaine, “Data Management for Large Language Models: Challenges, Opportunities, and Enterprise Requirements,” *IEEE Data Eng. Bull.*, vol. 46, no. 3, pp. 5–16, 2023.

[16] J. I. Janjua, M. Nadeem, Z. A. Khan and T. A. Khan, "Computational Intelligence Driven Prognostics for Remaining Service Life of Power Equipment," 2022 IEEE Technology and Engineering Management Conference (TEMSCON EUROPE), Izmir, Turkey, 2022, pp. 1-6, doi: 10.1109/TEMSCONEUROPE54743.2022.9802008.

[17] W. Alomoush, T. A. Khan, M. Nadeem, J. I., A. Saeed and A. Athar, "Residential Power Load Prediction in Smart Cities using Machine Learning Approaches," 2022 International Conference on Business Analytics for Technology and Security (ICBATS), Dubai, United Arab Emirates, 2022, pp. 1-8, doi: 10.1109/ICBATS54253.2022.9759024.

[18] J. I., T. A. Khan, M. S. Khan and M. Nadeem, "Li-Fi Communications in Smart Cities for Truly Connected Vehicles," 2021 2nd International Conference On Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS), Tangerang, Indonesia, 2021, pp. 1-6, doi: 10.1109/ICON-SONICS53103.2021.9617200.

[19] A. Ahamed, N. Ahmed, J. I., Z. Hossain, E. Hasan and T. Abbas, "Advances and Evaluation of Intelligent Techniques in Short-Term Load Forecasting," 2024 International Conference on Computer and Applications (ICCA), Cairo, Egypt, 2024, pp. 1-9, doi: 10.1109/ICCA62237.2024.10927804.

[20] J. I. Janjua, M. Nadeem and Z. A. Khan, "Machine Learning Based Prognostics Techniques for Power Equipment: Comparative Study," 2021 IEEE International Conference on Computing (ICOCO), Kuala Lumpur, Malaysia, 2021, pp. 265-270, doi: 10.1109/ICOCO53166.2021.9673564.

[21] S. Shen, A. Khandelwal, H. Zheng, and L. Jiang, “Towards Trustworthy and Deployable Large Language Model Systems for Enterprise Applications,” in *Proc. IEEE Int. Conf. Big Data*, pp. 1–10.