LLMOps: Definitions, Framework and Best Practices

Megha Sinha
Vice President
Data Science & Insights
Genpact
Bengaluru, India
megha.sinha@genpact.com

Sreekanth Menon
Vice President
Data Science & Insights
Genpact
Bengaluru, India
Sreekanth.Menon@genpact.com

Ram Sagar
Manager
Data Science & Insights
Genpact
Bengaluru, India
ram.sagar@genpact.com

2024 International Conference on Electrical, Computer and Energy Technologies (ICECET) | 979-8-3503-9591-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICECET61485.2024.10698359

***Abstract*—Operationalizing large language models is unlike traditional AI solutioning. As the vestiges of the MLOps paradigm serve as a reminder of the sophistication that surfaces at scale, Generative AI, while mitigating a few of those challenges, brings a few of its own. This is where LLMOps (Large Language Model Operations) comes into the picture. LLMOps is a subset of FMOps (Foundation Model Operations) that builds on the principles of MLOps (Machine Learning Operations) and helps enterprises deploy, monitor, and retrain their LLMs seamlessly. This paper provides a comprehensive definition of LLMOps and a robust framework and highlights best practices accrued through our experience building solutions in different domains. Finally, this work attempts to provide guidance for AI practitioners who want to operationalize their GenAI applications seamlessly.**

Keywords— LLMOps, MLOps, FMOps, operations, workflow orchestration, GenAI, foundation models (FMs)

I. INTRODUCTION

Generative AI is poised to disrupt mundane tasks. However, setting up architectures to support GenAI would require additional efforts. Employees should be equipped with knowledge of prompt engineering, prompt evaluation, prompt drift, prompt injections, and guard railing mechanisms. A whole new gamut of roles and responsibilities must be integrated with existing teams. Building next-generation enterprise-grade AI solutions would require cross-functional teams embedded with new roles. For instance, a domain expert, like a distinguished medical practitioner, must now embrace a foundational grasp of the technical bedrock that underpins these foundation models [1]. This is no mere supplementary skill; the linchpin unveils the potential of the large language models (LLMs). Large language models (LLMs) like OpenAI's GPT-4 [2], which power conversational interfaces like ChatGPT, are a type of foundation model. These interfaces powered by LLMs can generate new content in text, image, video and audio format by taking instructions in natural language. The multi-modal foundation models (FMs) are collectively called as generative AI (GenAI), which refers to the generative capabilities of the FMs that can be applied to narrow tasks. Furthermore, as we navigate the expansive terrain of GenAI, a new vista of possibilities unfurls before us. Consider this akin to the emergence of novel disciplines within the organizational framework, embodied by linguistics experts, AI quality controllers, AI editors, and prompt engineers. In domains where GenAI illuminates the path ahead, it becomes imperative to dissect existing roles into their elemental constituents methodically. GenAI invigorates each task – orchestrating the symphony of full automation, harmonizing in augmentation, or preserving the sanctity of untouched tasks, all with the precision of a seasoned consultant.

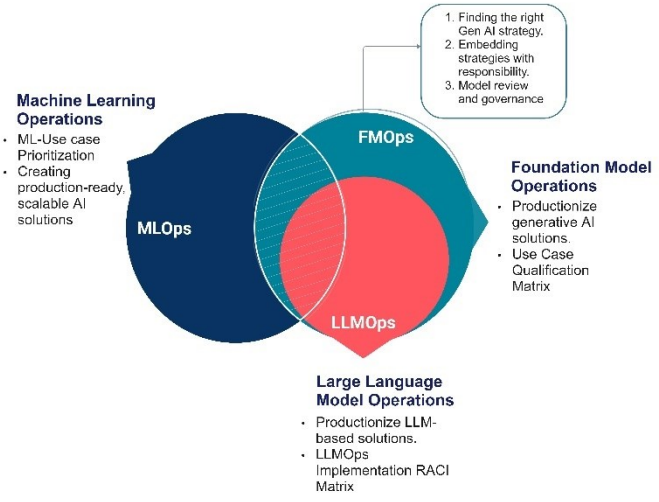


Fig.1. How LLMOps differs from its predecessors

979-8-3503-9591-4/24/$31.00 ©2024 IEEE

A foundation model (FM), by definition, is considered to be good for diverse downstream tasks. Since most enterprises deal with niche use cases from multiple industries, a domain expert and prompt engineer would create prompts and additional data that would fine-tune and help the FM to deliver the desired output. Operationalizing large language models is unlike traditional AI solutioning. As the vestiges of the MLOps [3] paradigm serve as a reminder of the sophistication that surfaces at scale, GenAI, while mitigating a few of those challenges, brings a few of its own. This is where LLMOps (Large Language Model Operations) comes into the picture. LLMOps(Large Language Model Operations), a subset of FMOps(Foundation Model Operations) built on the principles of MLOps(Machine Learning Operations) and helps enterprises deploy, monitor and retrain their LLMs seamlessly.

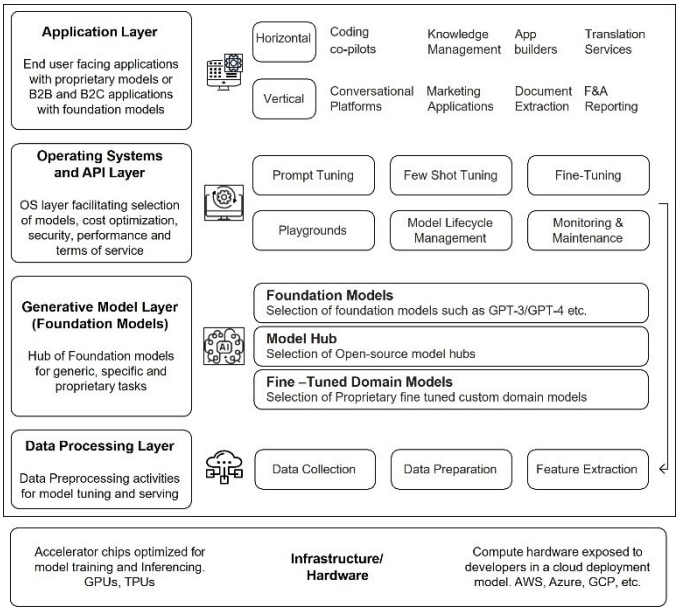


Fig.2. GenAI Enterprise Stack

Like any typical AI tech stack, GenAI applications, too, have an application layer, as shown in Fig.2, that communicates with the model hub, which in turn communicates with the data layer. Shown above is a brief overview of a typical GenAI tech stack.

Unlike classic ML and MLOps, FMOps and LLMOps [4] differ in four main areas: personnel and processes, foundation model (FM) selection and adaptation, FM evaluation and monitoring, data privacy, model deployment, and technology requirements. Generative AI developers integrate the chosen FM into solutions, collaborating with prompt engineers to automate converting end-user input into appropriate FM prompts. Prompt testers contribute entries to the prompt catalog for automatic or manual testing. GenAI developers establish prompt chaining and application mechanisms for the final output, where prompt chaining breaks complex tasks into smaller, more manageable sub-tasks, enhancing context awareness. They also implement input and output monitoring, such as toxicity detection, and introduce a rating mechanism to enrich the evaluation prompt catalog with examples. To provide functionality to end-users, a frontend website is developed to interact with the backend.

DevOps and cloud application developers follow best practices to implement input/output and rating features. Beyond basic functionality, the front and back end must support personal user accounts, data uploading, black-box fine-tuning initiation, and personalized models instead of the base FM. The LLMOps pipelines have a new application layer that serves as the workspace where GenAI developers, prompt engineers, testers, and application developers collaboratively build generative AI applications' backend and frontend components. While GenAI end-users access the front end of these applications, typically through a web-based user interface, data annotators must be able to perform data preprocessing tasks without directly accessing the backend data storage infrastructure. Consequently, a secure web-based user interface (website) with integrated editing capabilities becomes essential for their interactions with the data. In the next section, we shall discuss in detail about how all these processes can seamlessly fit into a framework--- an LLMOps framework.

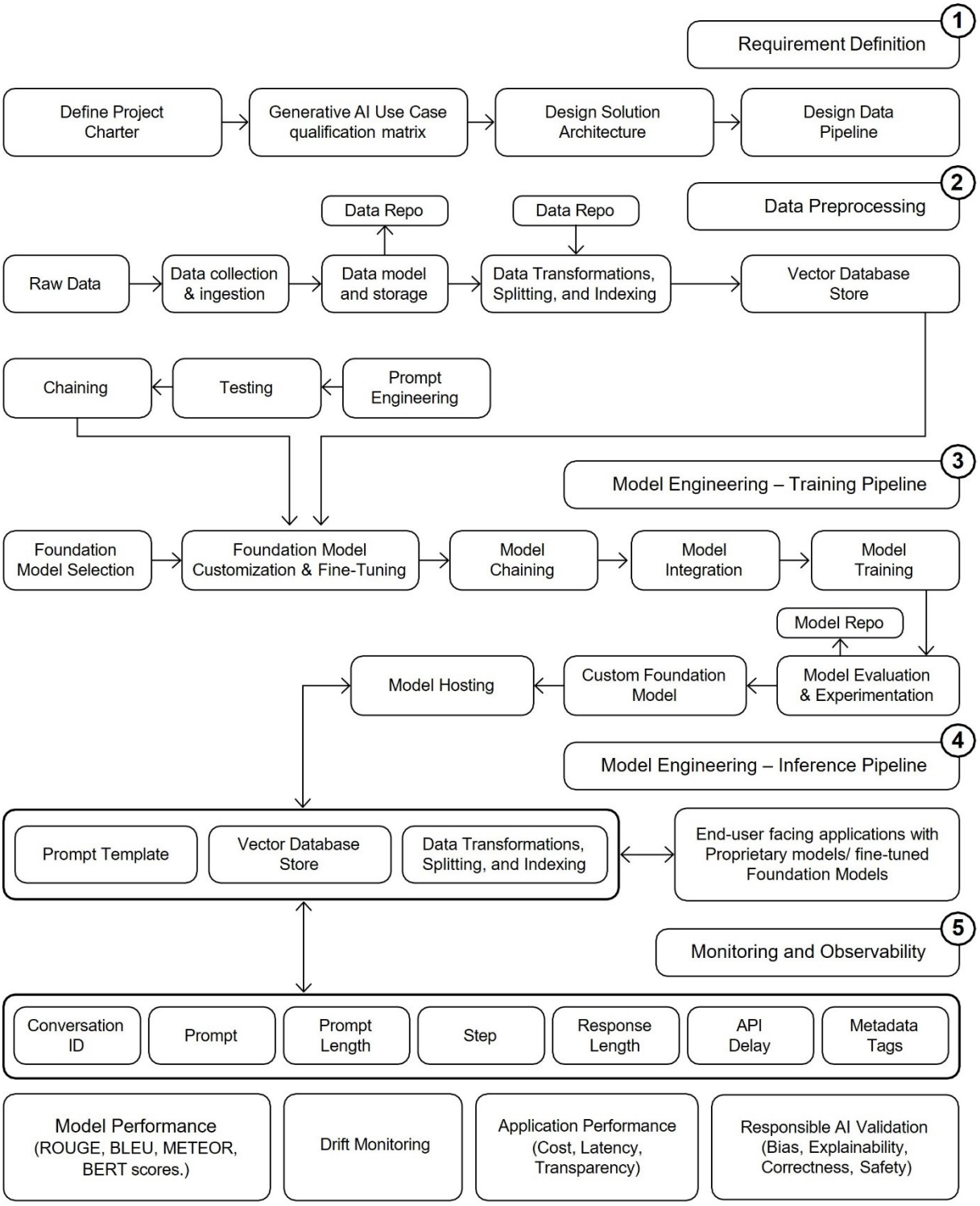


Fig.3. LLMOps framework

II. LLMOps FRAMEWORK

Fig.3 depicts the foundational LLMOps framework that acts as a scaffold for the enterprise-grade GenAI solutions that we have built over the past few months. The LLMOps framework consists of the following main steps:

A. Defining a Project Charter

This stage involves defining project goals, scope, and success metrics. It is important to involve key stakeholders and subject matter experts to create a well-defined project charter with clear objectives and expectations.

B. Generative AI Use Case Selection

This stage involves identifying GenAI use case value generation and implementation complexity. Typically, assessments of use case viability are done using a use case qualification matrix that considers all the salient points indicating risk and ROI.

C. Solution Architecture

Designing a scalable and cost-effective architecture is critical to the project's overall outcome. This requires collaboration with SMEs and ML engineers to develop robust architectures.

D. Design Data Pipeline

This stage involves determining data sources, integration methods, and data flow. This further helps in developing a comprehensive data strategy and identifying data sources.

E. Data Collection

Contextual Data to train the LLMs needs to be made available to reduce hallucinations and contextualize the outcomes to the business at hand. Key steps involve implementing automated data collection scripts from enterprise systems and databases to retrieve contextualized data and normalized data formats and use data connectors that can handle various sources seamlessly.

F. Data Transformation

This step deals with cleaning noisy text data and handling missing data. This also includes typical correcting inconsistencies, and removing duplicate data. One best practice is to leverage imputation techniques to clean data and utilize a combination of human-annotated and synthetic data to compensate for the missing data.

G. Data Storage

Conventional database storage systems are not adequate, optimized and fast to retrieve unstructured data stored in the form of embeddings. This is why it is recommended to use vector databases to store such data. Today, there are many open-source and closed-source vector databases, which we discuss in the next section.

H. Data Handling

In addition to traditional AI data pipeline building, GenAI introduces techniques/steps such as prompt engineering, which must be carefully crafted to ensure that the LLM generates the desired output. The data for training a LLM is required to be tokenized and normalized. The following two steps are important in data preparation:

- Data Versioning helps monitor development.
- Data Encryption and Access Controls help safeguard data with encryption and enforce access controls, such as role-based access, to ensure secure data handling.

I. Model Training

Once the data is prepared using the steps mentioned above, the LLM can be considered trained. The data scientist has to choose between prompt tuning, few shot tuning, and fine-tuning an LLM to create a custom-trained model. It is important to optimize model performance using established libraries and techniques for fine-tuning and enhancing the model's capabilities in specific domains with contextualized data specific to the business function and industry. This requires careful assessment between performance and cost to identify which stage of tuning is best suitable for the problem at hand.

J. Model Evaluation

In assessing trained or fine-tuned Large Language Models (LLM), predefined evaluation metrics are employed on benchmark tasks to gauge their performance. This involves measuring their capacity to produce accurate, coherent, and contextual. In addition to this, factuality [5], robustness, and trustworthiness of the model outputs are also key metrics for evaluation, amongst others.

K. Model Governance

A LLM must be tracked for performance [6], making changes to it as needed and retiring it when it is no longer needed. This requires defining and identifying metrics that must be captured at each run of the model in production to capture the prompt and response along with the latency and metadata. Setting thresholds and trigger alerts to re-train the LLM for any deviations is also considered a good practice.

L. Model Inference

The best practice for model inferencing [7] is to choose the appropriate deployment strategy based on budget, security, and infrastructure requirements. Defined architectural patterns must be identified to deploy the fine-tuned models in production. Alternatively, issues such as concurrent users, scalability, and performance considerations must also be managed for off-the-shelf LLM usage.

M. Model Serving

Model serving identifies the design pattern to make the LLM available to users. LLM serving decision pivots around the costs that are incurred. It takes into account the optimal way of using the compute resources. While many cloud providers offer these services, there are toolkits like Ray Serve [8] and others with custom response streaming and dynamic request batching for LLMs. We shall discuss more about the tool selection in the next section.

N. Model Monitoring

The accuracy demands of the LLM outputs vary with use cases even within the same domain; a clinician's treatment decision requires more accuracy than classifying log records of the patients. And, given the fact that LLMs tend to hallucinate, model monitoring is a very critical stage in the LLMOps framework. It depends on extrinsic factors like AI acts across geographies [9] and intrinsic factors like model-level metrics. This requires creating a playbook involving IT, Infosec, Legal, Business Owners, and ML Practitioners to arrive at a consensus of parameters that need to be monitored. A general thumb rule is to create a system that can capture machine and human feedback to improve the performance of the LLM or trigger alerts as required. Metrics selection depends on the type of use case at hand.

In the next section, we shall briefly discuss the current landscape of LLMOps tools and how they fit into the LLMOps framework.

III. METHODOLOGY

GenAI's operational logistics and facilitation are provided by various tool vendors, offering essential support in configuring, executing, and monitoring the end-to-end workflow. This underlines the importance of having the right tech stack that connects various layers, layers that are specific to GenAI pipelines. At the enterprise level, identifying and finalizing individual components of the LLMOps framework comes down to diligent decision-making; identifying the right tool, platform, and metric requires an assessment of the use case at hand and the maturity of the organization. As an organization, our approach to building AI solutions has always been tool-agnostic and platform-agnostic.

As discussed in the previous section, defining the project charter and drawing a conclusion on use case viability are critical first steps in the LLMOps workflow. These decisions are made with the help of in-house proprietary assessment frameworks and SMEs. For instance, the results of maturity assessments eventually help decide the kind of tools that the infrastructure would support. For instance, GenAI popularity has resulted in a tremendous demand for vector databases, which were traditionally restricted to the search domain. The emergence of ChatGPT has underscored the potential of vector databases to amplify the capabilities of large language models (LLMs). Vector Databases are used to store and manage vector representations of data, which may include embeddings of text,

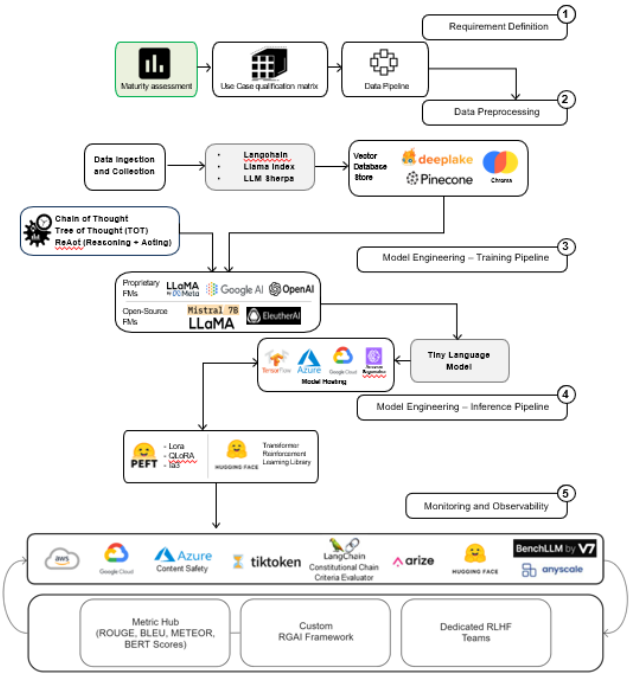


Fig.4. An Overview of LLMOps framework in production

images, or other features. Today, there are a handful of vector databases like DeepLake, Pinecone [10], Chroma dB, Azure cognitive search, and AWS Open Search. Picking the vector database depends on performance factors [11] like updatability, speed, sharding, replication, and execution. It also depends on an enterprise maturity that decides whether to go for open-source or closed-source. Similarly, identifying tools, libraries, and platforms for every stage of the LLMOps framework depends on a combination of people, process, technology, and their respective positions in the enterprise growth trajectory.

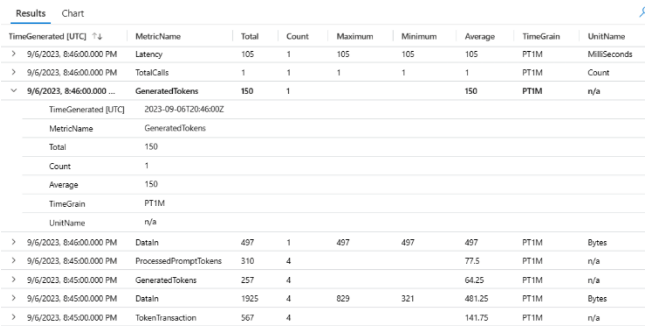
The next critical decision is the selection of the LLM. Once the cleaned data is stored in the vector database, the next step is to select the appropriate model. Depending on the cost-accuracy trade-offs, a suitable model is selected. For instance, an enterprise trying to build a tiny language model would prefer to fine-tune an open-source model. Whereas use cases that do not restrict data transfer can leverage the specialized cloud AI services. Using the right technique, as discussed above, the instructions that dictate model behavior are decided and integrated with the backend. The engineered instructions and the data stored in the vector database, once deployed, can be leveraged for querying, summarization, and other tasks. The generated outputs are then evaluated for their accuracy and usability. This layer undergoes an iterative process of re-training to maintain usability. This is done through RLHF or reinforcement learning through human feedback in combination with automated prompt tuning techniques.

Evaluation of Generative AI applications require monitoring and observability of the overall application performance. Existing platforms in the market provide out-of-the-box capabilities for LLM monitoring. However, our comprehensive approach includes a combination of tools and

platforms to track and report the LLM metrics and overall application metrics. This ensures an accurate, relevant, and high-performance system capable of scaling in production systems.

Fig.5 and Fig.6 depict one of the popular LLM evaluation tools offered by Microsoft Azure [12] that were used as part of the LLMOps framework which was implemented for one of the use cases. These tools measure the application health by tracking user activity metrics, system performance, and underlying knowledge base or data health.

- Application Metrics: The Azure Monitor provides end-to-end observability for applications. Kusto query language is used to monitor the data of the Azure OpenAI resource. To examine the Azure metrics, the following query displays a subset of the available columns of Azure Metrics data:



Results | Chart

TimeGenerated [UTC]	MetricName	Total	Count	Maximum	Minimum	Average	TimeGrain	UnitName
9/6/2023, 8:46:00.000 PM	Latency	105	1	105	105	105	PT1M	MilliSeconds
9/6/2023, 8:46:00.000 PM	TotalCalls	1	1	1	1	1	PT1M	Count
9/6/2023, 8:46:00.000 ...	**GeneratedTokens**	**150**	**1**			**150**	**PT1M**	n/a
TimeGenerated [UTC]	2023-09-06T20:46:00Z							
MetricName	GeneratedTokens							
Total	150							
Count	1							
Average	150							
TimeGrain	PT1M							
UnitName	n/a							
9/6/2023, 8:46:00.000 PM	DataIn	497	1	497	497	497	PT1M	Bytes
9/6/2023, 8:45:00.000 PM	ProcessedPromptTokens	310	4			77.5	PT1M	n/a
9/6/2023, 8:45:00.000 PM	GeneratedTokens	257	4			64.25	PT1M	n/a
9/6/2023, 8:45:00.000 PM	DataIn	1925	4	829	321	481.25	PT1M	Bytes
9/6/2023, 8:45:00.000 PM	TokenTransaction	567	4			141.75	PT1M	n/a

Fig.5. Snapshot of GenAI application health monitoring

- LLM Metrics: Azure Machine Learning provides a Data Collector feature that logs inference data in Azure Blob Storage, allowing data collection for new or existing online endpoint deployments. By using the provided Python SDK, the collected data is automatically registered as a data asset in the Azure machine learning workspace, which can be utilized for model monitoring purposes.

Combining the two categories, a custom dashboard was created to track and monitor overall application health and alert stakeholders at pre-defined thresholds.

These assessments assist in monitoring the overall health of the LLM application. In addition to this, traditional metrics like precision, recall along with metrics like relevance, completeness, brevity, and toxicity are also used for testing LLM reliability and robustness.

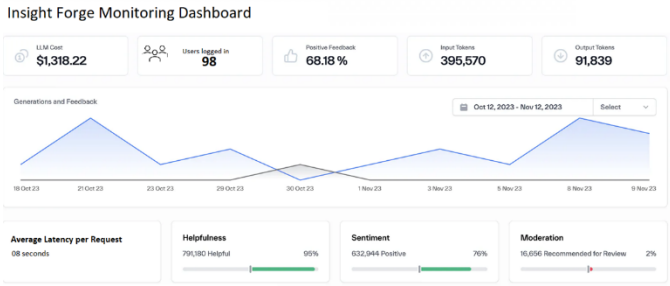


Fig.6. Comprehensive LLM application monitoring dashboard

IV. CONCLUSION

This paper attempts to standardize the LLMOps terminology and provide a realistic perspective on implementing GenAI applications. The frameworks and best practices discussed in this paper are derived from the accumulated experience of developing solutions in various domains, and hence, they are applicable to any AI team trying to operationalize in different capacities. That said, we also believe that the GenAI frameworks are a work in progress and should evolve to accommodate the ethical and social implications of operationalizing LLMs. A robust LLMOps framework will help developers to adapt to these changes and provide ease of integration to the AI practitioners to experiment and productionize new tools and techniques.

REFERENCES

[1] R. Bommasani et al., “On the Opportunities and Risks of Foundation Models”, July 2022.
[2] Open AI, “GPT-4 Technical Report”, published.
[3] D. Kreuzberger et al.,“Machine Learning Operations (MLOps): Overview, Definition, and Architecture”, May 2022.
[4] S. Kartakis and H. Hotz, “FMOps/LLMOps: Operationalize generative AI and differences with MLOps”, September 2023.
[5] Y. Chang et al., “A Survey on Evaluation of Large Language Models”, December 2023.
[6] Microsoft, “Evaluation and monitoring metrics for generative AI”, November 2023.
[7] H. Naveed et al., “A Comprehensive Overview of Large Language Models”, December 2023.
[8] Anyscale Ray Team, “Building Production AI Applications with Ray Serve”, October 2023.
[9] Council of the EU, “Artificial intelligence act: Council and Parliament strike a deal on the first rules for AI in the world”, December 2023.
[10] R. Schwaber-Cohen, “What is a Vector Database & How Does it Work? Use Cases + Examples”, May 2023.
[11] J.J. Pan, J. Wang and G. Li, “Survey of Vector Database Management Systems”, October 2023.
[12] Microsoft, “Model monitoring for generative AI applications (preview)”, November 2023.