International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V72I7P112

Volume 72 Issue 7, 93-100, July 2024
© 2024 Seventh Sense Research Group®

Original Article

Strategies for Integrating Generative AI in Industrial Settings

Pan Singh Dhoni[1,] Saurabh Shukla[2], Jagjot Bhardwaj[3]

[1]Independent Researcher, Engineering Technical Manager at a prestigious Company.
[2]Independent Researcher, Specialist Solution Architect at a prestigious Company.
[3]Independent Researcher, Lead, Minneapolis, MN, USA at a prestigious Company.

[1]Corresponding Author : ps.dhoni@gmail.com

Received: 18 May 2024 Revised: 28 June 2024 Accepted: 18 July 2024 Published: 31 July 2024

Abstract - *The evolution of ChatGPT has triggered anxiety across the globe, from industries to governments. There has been considerable discussion about this, from scholarly communities to social media and organizational forums. Leading technology companies have started investing heavily in artificial intelligence, which will broadly impact industry and society. The question now arises: how can industry leverage this capability for our betterment and the growth of industries? In this interpretative paper, the authors aim to highlight the best approach to implementing generative Artificial Intelligence in organizations, from small to large. The approach discussed advocates for a step-by-step, or ladder, method, ensuring that the models used yield better outcomes and reduce instances of hallucination. The results suggest that lowering hallucinations and being cost-effective can lead to better outcomes that accurately meet organizational needs. Additionally, the paper highlights the importance of generative AI model security and budget allocation to POCs, with a strong emphasis on the feedback loop to the production of the product, which provides reassurance and confidence in the system's reliability.*

Keywords - *AI, ChatGPT, Generative AI, LLM, Rag.*

1. Introduction

Research and innovation continually propel technological advancements, emphasizing novel concepts and papers. However, a significant gap often exists between theoretical knowledge and industry implementation. As the author reviews increasing articles across various platforms, a pertinent question arises: How does the industry practically implement these ideas?

The referenced research papers [1-7] provide insights into selecting Large Language Models (LLMs) and building AI applications using enterprise datasets. However, they lack a comprehensive approach to effective application development. This paper identifies critical areas for improvement, such as model selection approaches, generative AI model security, and product development lifecycle across different sectors.

This research begins with an introduction to Generative AI (GenAI), outlining its journey and evolution within industrial applications. The methodology section explores various methods, compares them, and highlights the importance of prompt engineering patterns and fine-tuning techniques. It also discusses model security, the Proof of Concept (POC) approach for initial production versions and the feedback loop for improving first project implementations. Subsequently, real-world use cases are illustrated, providing insights into their implementation and impact. The results section details the outcomes derived from these implementations.

Additionally, a specific use case is presented to demonstrate GenAI's efficacy in industrial scenarios. Finally, the author synthesizes the findings into a conclusive discussion, outlining potential avenues for future research and development in this field.

2. AI Terminology and Journey

2.1. Terminology Used for AI

Figure 1 has highlighted different terminology used for Artificial Intelligence (AI); below is a more detailed explanation:

2.1.1. AI (Artificial Intelligence)

Systems that emulate human intelligence, performing tasks like speech recognition and decision-making.

2.1.2. ML (Machine Learning)

An AI subset where computers learn from data to make predictions or decisions, using algorithms to learn and make informed decisions.

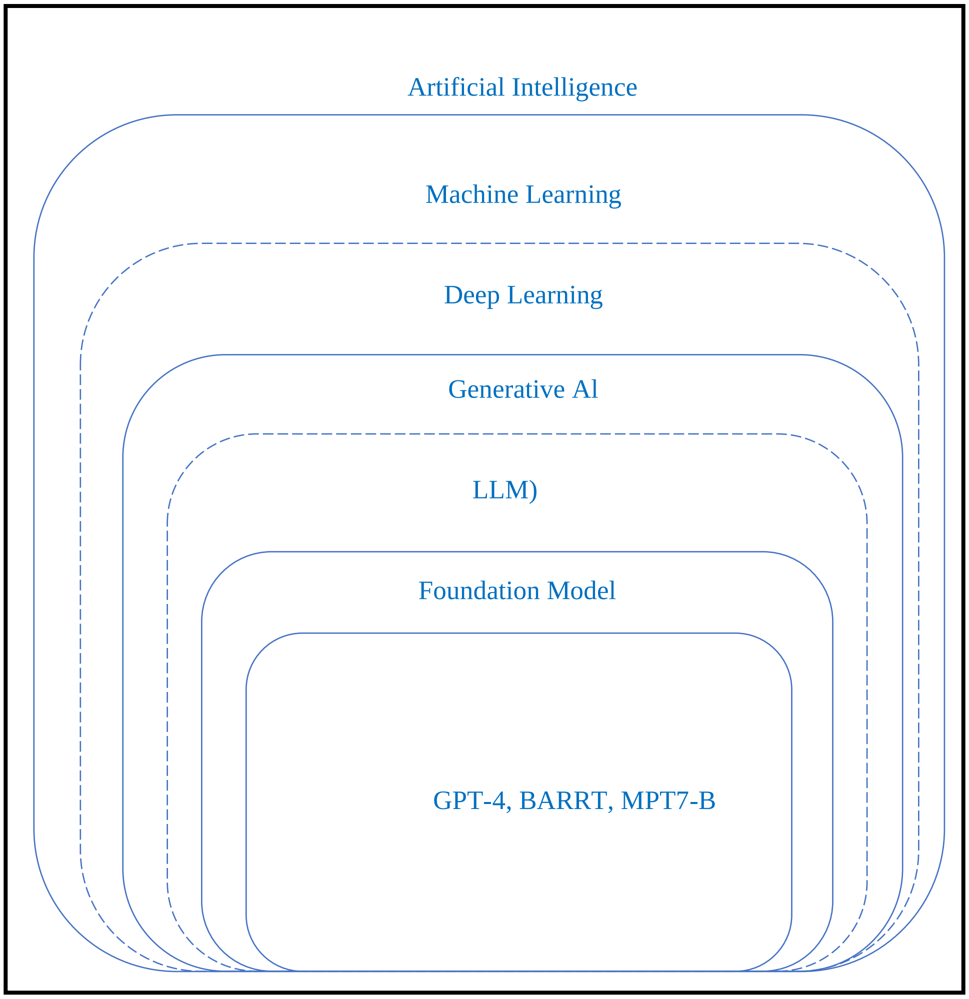

Fig. 1 Artificial intelligence

2.1.3. DL (Deep Learning)

An ML subset using neural networks inspired by the human brain to recognize patterns in data.

2.1.4. Generative AI

AI that generates new content, ranging from images and text to code, creating data previously nonexistent.

2.1.5. LM (Large Language Models)

Models trained on extensive datasets for advanced text processing, including translation and content generation.

2.1.6. Foundation Models

Large-scale models like GPT-4 and BART, trained on vast data, fine-tunable for specific tasks, providing a broad base of understanding for customization.

2.2. Generative AI Journey

In the rapidly evolving landscape of Generative Artificial Intelligence (GenAI), organizations are increasingly focusing on the strategic integration of AI technologies to stay ahead of the curve. This drive towards innovation raises crucial questions about the underlying needs, key questions, and pivotal advice for harnessing the full potential of GenAI. At the heart of this endeavor lies the pursuit of Complete Control over one's models and data, emphasizing data sovereignty as a critical competitive advantage. Ensuring robust Production Quality as solutions transition from research and development to full-scale production encapsulates another vital aspect, highlighting the seamless integration of GenAI into existing Machine Learning Operations (MLOps) frameworks. Moreover, the imperative to Lower Costs without compromising on efficiency or effectiveness calls for a strategic approach to model serving and the fine-tuning process. Balancing these priorities requires a thoughtful consideration of ownership, competitive differentiation, and the strategic deployment of technology within the organization's broader ecosystem. This introduction sets the stage for a deeper exploration of how businesses can navigate the complexities of GenAI adoption, ensuring control, quality, and cost-efficiency in their AI-driven initiatives. Table 1 reference taken from [8][9][10].

Table 1. Considerations for deploying generative AI applications

Underlying Needs	Key Questions	Key advice
Complete Control	Do you own and control your models and data? What gives you an edge over competitors? What are your security and privacy risks?	Your data is your competitive edge. Maintain control of it and use it for GenAI applications and custom models.
Production Quality	How do you move from R&D to production? How does GenAI fit into your MLOps platform and processes?	GenAI is best deployed on a data-centric platform. Most of the traditional MLOps work for GenAI but pay attention to a few new requirements from GenAI.
Lower Costs	How do you serve models and pipelines efficiently? How do you fine-tune and pre-train models efficiently?	Leverage optimized serving systems. Plan for building customized models on your primary ML/AI platform.

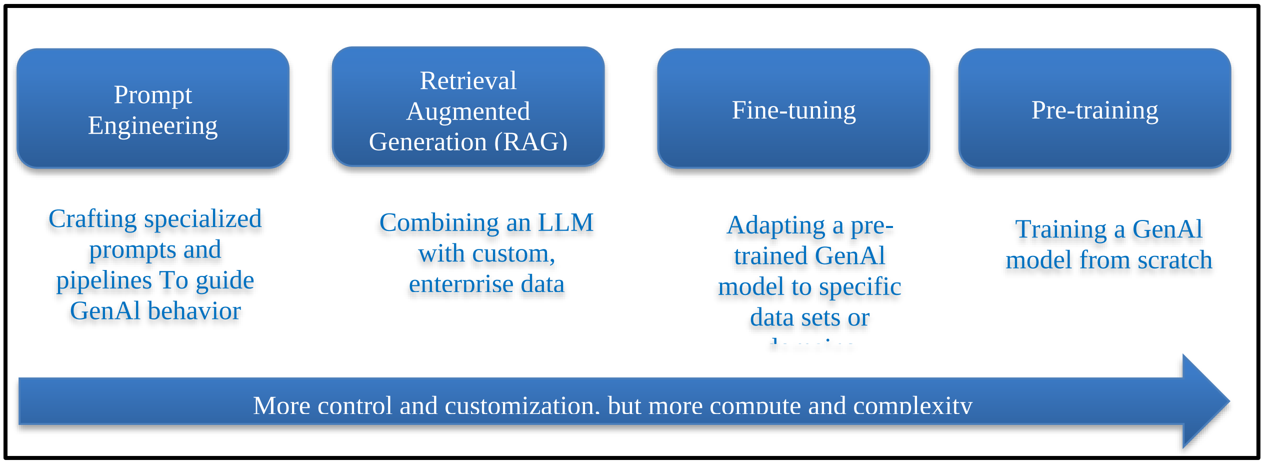
Prompt Engineering
Retrieval Augmented Generation (RAG)
Fine-tuning
Pre-training
Crafting specialized prompts and pipelines To guide GenAI behavior
Combining an LLM with custom, enterprise data
Adapting a pre-trained GenAI model to specific data sets or
Training a GenAI model from scratch
More control and customization, but more compute and complexity

Fig. 2 Gen AI journey

GenAI's Journey towards sophisticated generative AI systems unfolds in four critical stages, emphasizing the leveraging of proprietary data. Figure 2 details this incremental and strategic progression.

2.2.1. Prompt Engineering

This initial step involves the meticulous crafting of specialized prompts, fine-tuning the AI's interaction to guide its behavior effectively within specific contexts.

2.2.2. Retrieval Augmented Generation (RAG)

The second phase enhances AI's capabilities by integrating it with external knowledge sources, thus enriching its responses with broader, more accurate context drawn from enterprise data.

2.2.3. Fine-Tuning

The subsequent stage focuses on customizing a pre-trained language model to suit particular datasets or domains, refining its applicability and precision for specialized tasks.

2.2.4. Pre-Training

The final and most complex phase involves training a large language model from scratch building a foundational AI tool uniquely tailored to specific needs.

As one moves from prompt engineering to pre-training, control and customization of the AI system increase, but so do the computational resources and complexity required. This iterative path underscores a strategic approach to developing generative AI, balancing the demand for precision and specificity with the need for comprehensive data analysis and model training.

3. Methodology

This methodology section aims to elucidate the practical approaches and architectural patterns for integrating GenAI capabilities within industrial settings. The following subsections will delve into a multifaceted framework encompassing various scenarios, ranging from utilizing off-the-shelf LLMs to fine-tuning and pretraining custom models tailored to specific domains. Furthermore, we will explore architectural patterns that facilitate the seamless integration of GenAI into existing workflows and processes, enabling enterprises to reap the benefits of this transformative technology while maintaining control over their data and models.

3.1. High-Level Scenarios For GEN AI / LLMS

In general, there are two ways of improving the responses of LLMs:

- Add context to the query (and use the LLM as is)
- Change the behavior of models, e.g. by fine-tuning.

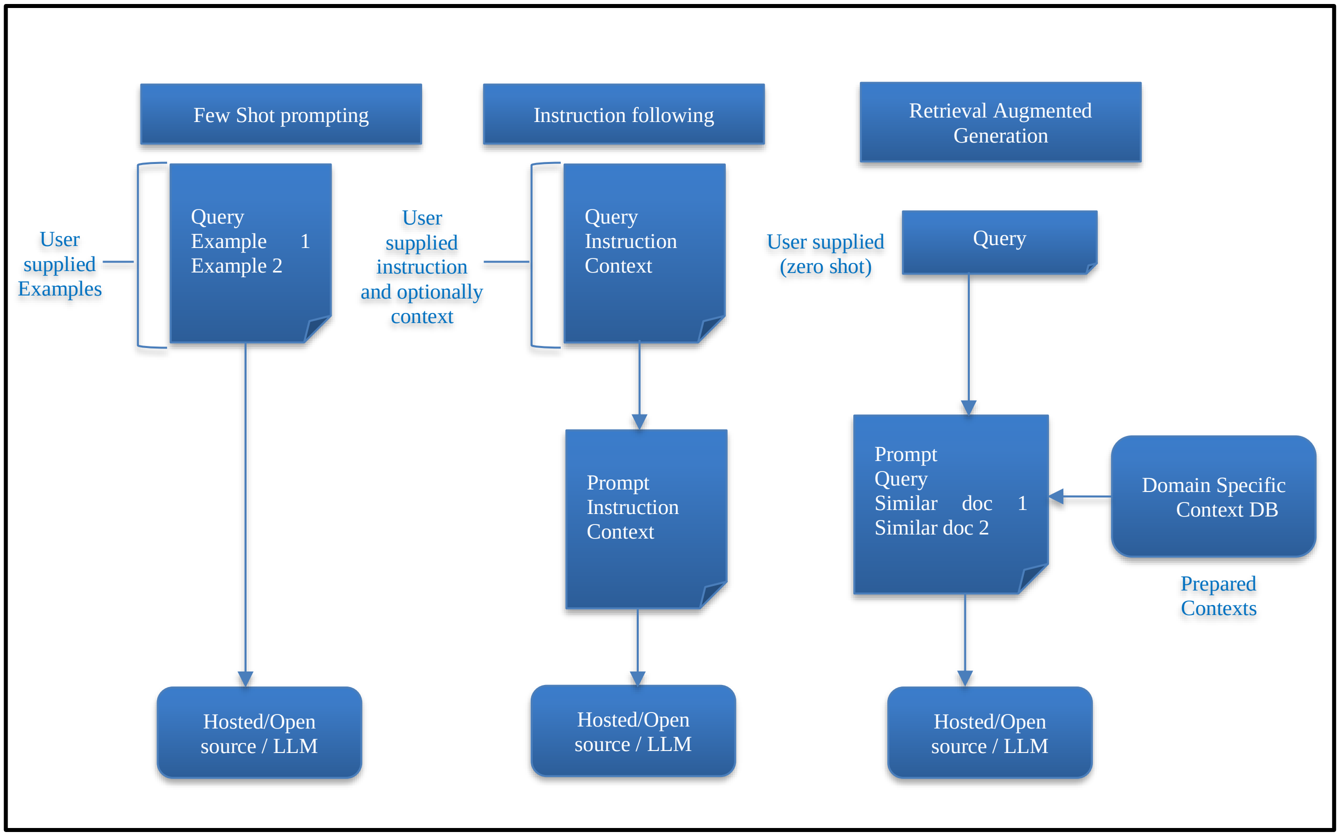

Fig. 3 Add context to the query

Let's start with Adding context to the query (See Figure 3). The easiest is "few shots" and "instruction following", where the user simply provides context either in the form of examples (few shots) or in the form of clear instruction (optionally adding some context) for the model.

In order to control the context and help the user to only need simple queries, "Retrieval Augmented Generation" can be set up. This requires a high-quality input set.

In this case, the system will retrieve similar data from the "domain-specific context DB" and augment the query automatically and independent of the user.

When it comes to altering the behavior of a model (See Figure 4), fine-tuning existing LLMs is typically preferred over constructing a new foundational model. The rationale behind this is that building your own foundational model is a highly challenging and costly endeavor.

A direct method of fine-tuning involves retraining the model with a smaller corpus (As per Figure 4) specific to the domain compared to the corpus used for building the foundational model. To optimize for time and cost, performance-efficient fine-tuning techniques (e.g., LoRA, QLoRA, etc.) can be utilized during unsupervised training.

Another approach, particularly suited for prompts involving "instruction following," is "instruction tuning." This method requires the creation of a high-quality instruction/response dataset, optionally with context, for supervised learning. Once again, performance-efficient fine-tuning methods may be considered.

The most favorable outcomes in instruction following prompts are expected when both fine-tuning methods are employed: first, domain-specific fine-tuning, followed by instruction tuning.

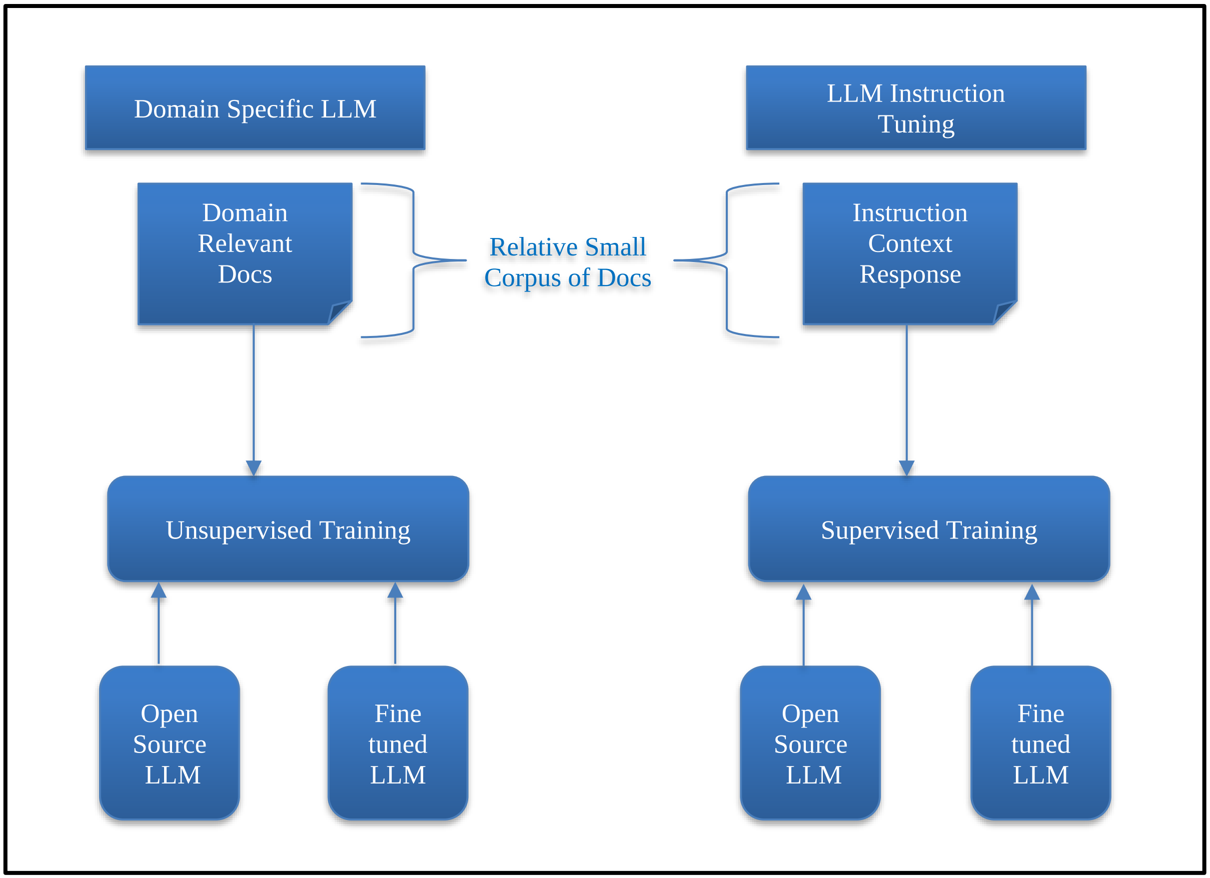

Fig. 4 Change the model behavior

3.2. Architecture Patterns for GenAI-Powered Application

The idea for the patterns is for Enterprises to build an AI App.

3.2.1. Pattern 1: Querying 3rd Party AI Services

The process is straightforward, involving the use of a third-party Large Language Model (LLM) behind your AI application.

In this scenario, user input is structured as a prompt. Developers have the option to supplement the user-provided input with additional instructions, effectively 'guiding' the model in performing the correct action.

This prompt is reliably sent to the third-party service, and the response is relayed to the customer. Logic to evaluate the response against predefined rules can also be implemented, ensuring a secure process.

3.2.2. Pattern 2: Querying Open-Source Models

In instances where using external third-party services is not viable (e.g., to avoid the leakage of intellectual property), the same objectives can be achieved with one of the open-source LLMs. This requires a preparatory step to select and serve the LLM in a "model-serving" capacity. The rest is identical to pattern 1.

3.2.3. Pattern 3a: Batch Inference (via Model Serving)

This approach caters to scenarios where inferences are generated in batches and stored for future use. Option (a) involves making the model available through 'Model Serving.' [11] Documents for inference, with or without contextual examples and instructions, are prepared, and prompts are generated. An API call is then made for these prompts. The received responses are verified and stored in an inference table for later reference. In this setup, the AI orchestrator—which handles embedding construction, prompt engineering, and sending requests to LLMs—may run on a standard CPU cluster since the actual inference is performed on the served model. While this method is efficient, it is crucial to account for the rate limits of the serving API, mainly when dealing with large data volumes [4].

3.2.4. Pattern 3b

Batch Inference Option (b) entails performing inference computations directly on a machine. To begin, the LLM is loaded onto the cluster. Next, documents for inference are prepared—with or without contextual examples and instructions—and prompts are generated. The cluster is then utilized to process the queries using the LLM. The responses may be verified and stored in an inference table for future use. This method involves more effort in setting up the environment but is accessible from the rate limits associated with a hosted model [12][13].

3.2.5. Pattern 4: Retrieval-Augmented Generation (RAG)

The ingested documents, which are of paramount importance in the RAG process, need to be both sufficient in quantity and high in quality. Their quality is crucial as it enhances the later responses. These documents will be cleansed and, if too lengthy, summarized using another AI model.

The orchestrating code plays a pivotal role in the RAG process. It loads an embedding model to generate embeddings for the ingested documents, which are then stored alongside the papers in a vector database for subsequent similarity. This reassures you of the reliability and efficiency of the process.

It's important to note that the RAG process offers flexibility. For instance, you could choose to store the documents in the database and compute similar documents in real-time based on a given query, giving you the power to tailor the process to your specific needs.

The context database, a crucial element in the RAG process, could take various forms, such as Vector Search, temporary vector libraries, or third-party services like Pinecone [4]. It serves as a repository for the embeddings and other relevant data, facilitating efficient document retrieval and generation.

3.2.6. Pattern 5: Prompt Engineering for RAG

This pattern will utilize the context stored in the context database.

The AI application loads the same embedding model used to build the context database and loads the corresponding LLM into Model Serving. It then creates an embedding for the query and retrieves similar documents from the context database, showcasing the system's adaptability. The system could also tap into additional third-party services to gather contextual information. This context is then integrated into the prompt and sent to the served LLM [14].

3.2.7. Pattern 6: Fine-Tuning of Open Source LLMs

The final approach, fine-tuning, is quite straightforward. You can select from various tools and techniques to perform the training step on a GPU cluster. One might start with an open-source model obtained from a model repository like Hugging Face. After fine-tuning, the model is then registered in a model registry for future use [15].

3.3. Generative AI Model Security

Ensuring the security of generative AI models is a crucial aspect of their deployment and use. It is essential to guarantee that these models are free from vulnerabilities that could be exploited. Utilize either an open-source model scanner or an approved licensing code scanner within the organization to achieve this. Regularly updating and patching the models can help in mitigating potential threats. Implementing robust security protocols and conducting thorough security assessments can further enhance the protection of AI models. Ensuring compliance with industry standards and regulations is also vital in maintaining the integrity and trustworthiness of generative AI systems.

3.4. Data Quality and Data Governance in Place for Data-Driven Apps

To derive meaningful value from generative AI, it is essential to clean, validate, and ensure the quality and trustworthiness of data. During data transfer for processing, it is vital to track and audit these movements to maintain security and enable effective data governance.

3.5. POC's and productionizing Application

According to the provided Figure 5, the industrial approach spans from budget allocation to production movements, incorporating a feedback loop for continuous improvement. Sample LLM application located at GitHub (https://github.com/padhoni/LLMApps developed using https://huggingface.co/ and https://ollama.com/).

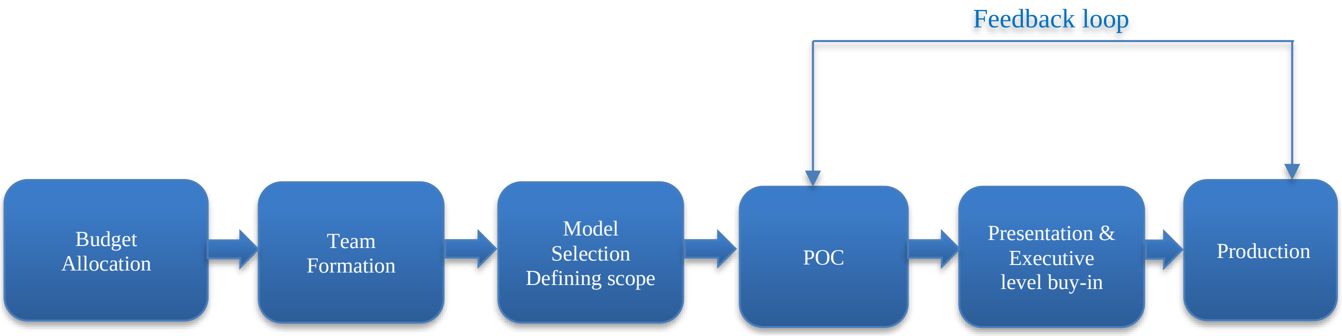
Feedback loop
Budget Allocation
Team Formation
Model Selection Defining scope
POC
Presentation & Executive level buy-in
Production

Fig. 5 Application development approach

4. Conclusion

Incorporating generative AI into the industrial arena offers a better opportunity to boost productivity, enhance competitive strength, and drive innovation. This innovation potential is particularly exciting as it can lead to new ways of working and breakthrough solutions across various sectors.

This paper strongly advocates for a strategic, step-by-step approach to integrating AI models into existing workflows. This method, meticulously designed to optimize outcomes and minimize instances of hallucination, provides a clear and compelling path for industrial integration, instilling confidence in the process.

This paper underscores the significance of techniques such as prompt engineering, retrieval-augmented generation (RAG), and fine-tuning. As emphasized in the abstract, these techniques are instrumental in achieving superior outcomes, cost-effectiveness, and reduced hallucinations. The architectural patterns presented offer a clear roadmap for enterprises to seamlessly integrate generative AI capabilities into their processes, enabling them to reap the benefits of this transformative technology while maintaining control over their data and models.

Furthermore, this paper emphasizes the need for generative AI model scanning for security measures, such as data encryption, access control, and regular security audits. This paper highlights the approach for developing generative AI applications from budget to production.

Additionally, Table 2 provides valuable insights into different methods for building GenAI-powered applications and guides selecting appropriate processes based on specific needs.

Table 2. Different methods

Method	Definition	Primary use case	Data requirements	Training time	Advantages	Considerations
Prompt engineering	Crafting specialized prompts to guide model behavior	Quick, on-the-fly model guidance	None	None	Fast, cost-effective, no training required	Less control than fine-tuning
Retrieval augmented generation (RAG)	Combining an LLM with external knowledge retrieval	Dynamic datasets and external knowledge	External knowledge base or vector database	Moderate (e.g. computing embeddings)	Dynamically updated context, enhanced accuracy	Increases prompt length and inference computation
Fine-tuning	Adapting a pre-trained model to specific datasets or domains	Domain or task specialization	Thousands of domain-specific or instruction examples	Moderate - long (depending on data size)	Granular control, high specialization	Requires labeled data, the computational cost
Pre-training	Training a GenAI model from scratch	Unique tasks or domain-specific corpora	Large datasets (billions to trillions of tokens)	Long (days to many weeks)	Maximum control, tailored for specific needs	Extremely resource-intensive

Future Research Directions

While this paper lays a solid foundation for integrating generative AI in the industrial sector, there remain ample opportunities for further exploration and advancement:

1. Investigating hybrid approaches that combine the methodologies discussed, such as integrating RAG with fine-tuning or exploring multi-stage fine-tuning techniques, could unlock new avenues for optimizing performance and accuracy.
2. Research into efficient and scalable deployment strategies for generative AI models in industrial settings is crucial for practical implementation and widespread adoption.
3. Exploring performance optimization and resource management techniques is essential, considering the computational resources required for methodologies like pre-training.
4. Advancements in continuous learning mechanisms could enable GenAI systems to adapt and evolve dynamically, enhancing their capabilities and relevance in rapidly changing industrial environments.
5. Governance and ethical considerations surrounding model deployment, as well as techniques for enhancing explainability and trust in generative AI systems, warrant further investigation to ensure responsible and transparent adoption.

6. Optimization and customization of GenAI applications for industry-specific use cases could unlock novel solutions tailored to the unique challenges and requirements of various sectors.

Prioritizing these research directions will not only enhance the capabilities of generative AI but also drive innovation, operational efficiency, and sustained competitive advantages in the rapidly advancing technological landscape of industrial sectors.

References

[1] Christof Ebert, and Panos Louridas, "Generative AI for Software Practitioners," *IEEE Software*, vol. 40, no. 4, pp. 30-38, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Kalyan Prasad Agrawal, "Towards Adoption of Generative AI in Organizational Settings," *Journal of Computer Information Systems*, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] David Sweenor, and Kalyan Ramanathan, *The CIO's Guide to Adopting Generative AI: Five Keys to Success*, TinyTechMedia LLC, 2023. [Google Scholar]

[4] Cheonsu Jeong, "Generative AI Service Implementation Using LLM Application Architecture: Based on RAG Model and LangChain Framework," *Journal of Intelligence and Information Systems*, vol. 29, no. 4, pp. 129-164, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Shreekant Mandvikar, "Factors to Consider When Selecting a Large Language Model: A Comparative Analysis," *International Journal of Intelligent Automation and Computing*, vol. 6, no. 3, pp. 37-40, 2023. [Google Scholar] [Publisher Link]

[6] Cheonsu Jeong, "A Study on the Implementation of Generative AI Services Using an Enterprise Data-Based LLM Application Architecture," *arXiv*, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Andrew Burgess, "Starting an AI Journey," *The Executive Guide to Artificial Intelligence*, pp. 91-116, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[8] Patrick Wendell et al., Lakehouse AI: A Data-Centric Approach to Building Generative AI Applications, 2023. [Online]. Available: https://www.databricks.com/blog/lakehouse-ai

[9] Sandra Durth et al., McKinsey and Company, The Organization of the Future: Enabled by Gen AI, Driven by People, 2023. [Online]. Available: https://www.mckinsey.com/capabilities/people-and-organizational-performance/our-insights/the-organization-of-the-future-enabled-by-gen-ai-driven-by-people

[10] François Candelon et al., BCG, The CEO's Guide to the Generative AI Revolution, 2023. [Online]. Available: https://www.bcg.com/publications/2023/ceo-guide-to-ai-revolution

[11] Eric Breck et al., "Data Infrastructure for Machine Learning," *SysML Conference*, 2018. [Google Scholar]

[12] Wenqi Jiang et al., "FleetRec: Large-Scale Recommendation Inference on Hybrid GPU-FPGA Clusters," *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3097-3105, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] Alexander Borzunov et al., "Distributed Inference and Fine-tuning of Large Language Models Over the Internet," *Advances in Neural Information Processing Systems*, 2024. [Google Scholar] [Publisher Link]

[14] Bongsu Kang et al., "Prompt-RAG: Pioneering Vector Embedding-Free Retrieval-Augmented Generation in Niche Domains, Exemplified by Korean Medicine," *arXiv*, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Teven Le Scao, and Alexander M. Rush, "How Many Data Points is a Prompt Worth?," *arXiv*, 2021. [CrossRef] [Google Scholar] [Publisher Link]