



## Enhancing Generative Model through the Integration of Large Language Models and MLOps

Supritha R<sup>1\*</sup>, Tejaswini M<sup>2</sup>

<sup>1</sup>Department of Information Science and Engineering, JSS Science and Technology  
University Mysuru, India

<sup>2</sup>Department of Information Science and Engineering, JSS Science and Technology  
University Mysuru, India

### ABSTRACT

The Generative model is a powerful tool in generating data resembling training sets, often struggling with producing coherent and contextually relevant outputs in domains like internal communication. Traditional HR processes involve manual resume analysis and are inconsistent due to subjective judgment. Due to this a novel approach developed in human resource (HR) chatbots. It leverages the capabilities of Large Language Models (LLMs) like ChatGPT to enhance the coherence and contextual understanding of generated responses in HR interactions, fine-tuning an LLM on a dataset of HR-specific queries and responses, and integrating with existing generative models. The outcomes of the experiment show notable gains in the relevance and quality of generated answers for HR queries. Analyze the effectiveness of our integrated approach and discuss its potential for broader applications in advancing internal communication and conversational AI systems within organizations.

**Keywords:** Generative models, Large Language Models (LLMs), Human Resource chatbots, MLOps, Internal communication, Conversational AI, Ensemble models.

### 1 Introduction

The Human Resource (HR) function plays a crucial role in bridging the technology and HR gap in the context of Industry 4.0 [1]. Due to continuous technological improvements The field of Human Resources (HR) is currently experiencing a significant transformation, driven by ongoing technological advancements. Among these innovations, the integration of Artificial Intelligence (AI) stands out as particularly promising. One of the key applications of AI in HR is the use of chatbots-automated conversational systems designed to streamline workflows and enhance employee experiences.

HR chatbots struggle to handle employee inquiries due to their extensive dataset training and lack of sophistication in handling natural language and unique situations. AI can improve recruitment by analyzing job descriptions, resumes, and applications, identifying candidates without bias, and aiding in candidate screening and interview scheduling, but it cannot provide complete descriptions.

The rapid advancement of technology, especially the implementation of the Large Language Model with MLOps in HR, has brought about significant changes in HR processes and practices.



As corporations increasingly approach digitizing their HR operations, it is crucial to understand the effects of the Large Language Model with MLOps on different aspects of HR such as employee productivity, health and safety, payroll processing, employee comfort, handling multiple resume selection process, real-time feedback and ticket handler.

Moreover, understanding how these HR tasks impact the creation and analysis of organizational networks can provide insights into how businesses can use generative chatbots to improve productivity and effectiveness across the board. This study looks at the relationship between chatbots and HR digitisation, as well as how HR digitisation affects organizational network analysis and design. Finding the benefits and drawbacks of HR digitisation in the context of organizational network analysis and design, as well as assessing the effects of HR digitisation on real-time feedback, improving employee comfort, boosting health and safety, measuring employee productivity, and automating payroll processing are some of the research objectives. Another is to examine the relationship between AI and HR digitisation in terms of measuring employee productivity, health and safety, payroll processing, employee comfort, handling multiple resume selection processes, real-time feedback, and ticket handler.

## 2 Recent Works

AI Chatbot has the potential to be extremely important in the areas of employee retention and engagement. Artificial intelligence (AI) algorithms are able to recognise patterns and trends in employee data that could point to excessive employee turnover or low engagement. HR professionals can use this knowledge to take proactive steps to solve these problems, like putting in place training programs or enhancing workplace culture [2].

AI can also be utilized to improve staff learning and development initiatives. AI systems are able to detect knowledge gaps in employees and suggest training programs to correct these gaps by examining employee data [3].

AI-powered learning platforms can improve learning results by customizing learning experiences to each employee's needs [4].

AI can also have a significant impact on performance management [5].

The Artificial Intelligence systems are capable of analysing employee performance data to pinpoint areas in need of improvement. Personalized performance improvement programs for each employee can be created using this information, which will raise productivity and performance levels overall [6].

Personalized Marketing Intelligence (PMI) is a system that uses Large Language Models and prompt engineering techniques to create personalized marketing messages. Validated through surveys, it uses marketing persuasion theories and research. However, it has limitations and recommends further exploration [7].

Chat2VIS is a solution that enhances language comprehension and visualizations using natural language queries, automating chart type selection and reducing development costs [8].

In this paper a comprehensive Machine Learning Operation's architecture that automates machine learning processes, highlighting the need for scalable implementation, ongoing



monitoring, and improvement of ML systems, while suggesting proven methods and resources for successful implementation [9].

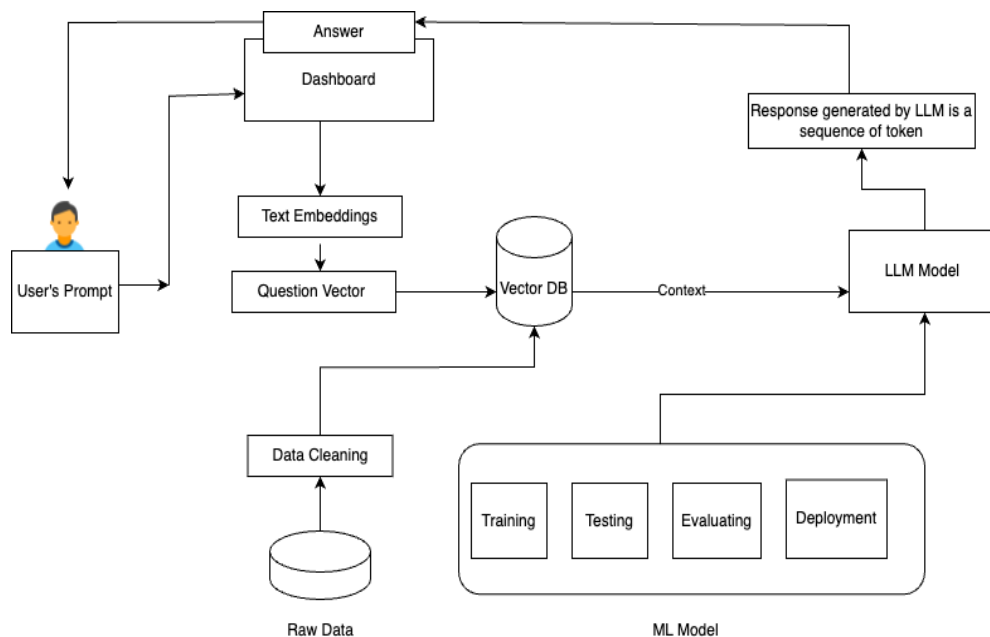
This paper model examines the ML lifecycle, focusing on integrating CI/CD pipelines into MLOps workflows, highlighting differences between DevOps and MLOps, and suggests a three-tiered efficiency framework for optimizing MLOps phases [10].

The Retrieval-Augmented Generation (RAG) model, a technique that utilizes LLMs to create generative AI services in businesses, addresses insufficient data and improves content generation, with real-world examples provided [11].

A multi-modal chatbot combining textual and visual data in manufacturing is being developed to enhance intent classification accuracy. This system uses an auto encoder and the YOLOv4 object detection technique, with experiments confirming its improved accuracy and speed. [12].

Prog Prompt is a robot task planning tool that structures LLM prompts with programming details, achieving high success rates in both physical and simulated contexts. It specifies actions and objects and offers sample programs, but further research is needed for specific settings and challenging jobs [13].

### 3 Proposed Work Explanation



**Figure 1:** Integration of Human Resource Chatbot on MLOps Framework

From the above figure the system introduces an advanced HR Q&A bot that leverages cutting-edge technologies, including generative AI, Lang Chain, and PDF processing tools. This system automates the extraction of relevant information from PDF resumes, identifying key details such as names, contact numbers, emails, skills, and educational qualifications. By utilizing Google Generative AI for embedding and FAISS for vector storage, the bot can



efficiently process large text chunks and perform accurate similarity searches to address user queries. The integration of logistic regression and ML flow facilitates the evaluation and logging of model performance, thereby enhancing the system's reliability and effectiveness. This automated approach reduces the time and effort required for screening resumes, while simultaneously increasing precision and consistency in candidate assessment, ultimately transforming HR workflows and improving operational efficiency. The system matches the resume against the required skills for a position and provides a score, indicating how well the candidate's qualifications align with the job requirements.

### User

The user initiates the conversation by entering text through a chat interface.

- Let  $U$  represent the user input.

### Text Embeddings

The system prepares the user's query by converting it into a numerical representation suitable for the LLM. This process, called text embedding, captures the semantic meaning of the words in the query.

- Let  $E$  be the embedding function that converts text into numerical representation.
- $V=E(U)$ , where  $V$  is the vector representation of the user input.

### User's Prompt & Question Vector

The user's query is then transformed into two parts:

**User's prompt:** This provides context for the LLM, potentially including previous conversation history.

**Question Vector:** This is the core question or intent derived from the user's query.

- Let  $P(V)$  represent the function that extracts the prompt from the vector  $V$ .
- Let  $Q(V)$  represent the function that extracts the question vector from  $V$ .
- $P=P(V)$
- $Q=Q(V)$

### Data Pre-processing

The system prepares the user's query for further processing. This involves:

**Cleaning:** Removing typos, special characters, or extraneous spaces from the text.

**Tokenization:** Breaking the query down into individual words or meaningful units.

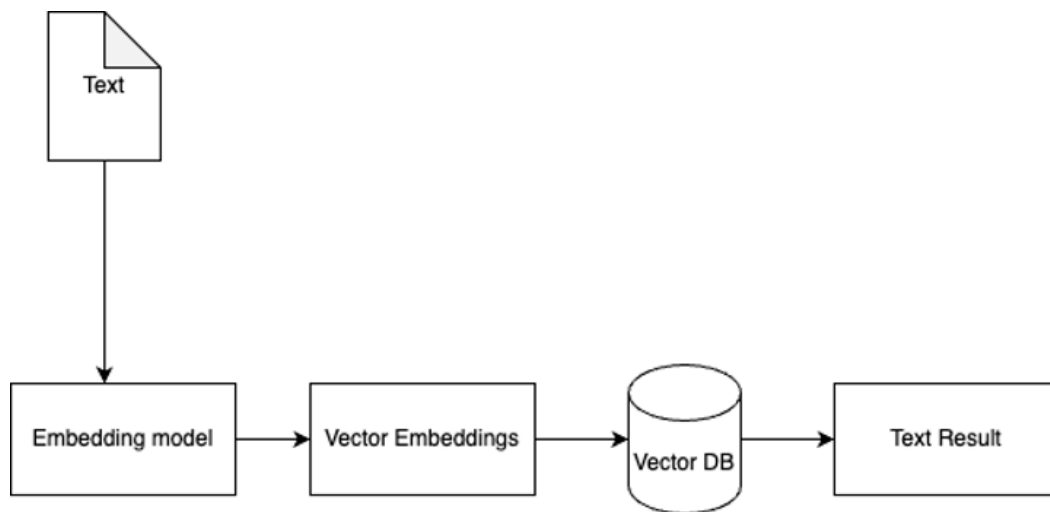
- Let  $C(Q)$  represent the cleaning function for the query vector  $Q$ .
- Let  $T(Q)$  represent the tokenization function for the query vector  $Q$ .
- $Q_{clean}=C(Q)$
- $Q_{tokens}=T(Q_{clean})$



## Vector DB

The pre-processed query is then forwarded to a Vector Database (Vector DB). Vector databases store and retrieve information based on semantic similarity.

- Let  $F(Q_{tokens})$  represent the function that forwards the query to the Vector DB and retrieves relevant vectors based on semantic similarity.
- $V_{similar} = F(Q_{tokens})$



**Figure 2: Workflow of Vector DB**

To power fast and accurate searches, user queries can encompass the different data formats like text, images, documents, videos, and audio. This data, structured or unstructured, is processed through an embedding model, often a complex neural network. This model transforms the data into high-dimensional vectors, essentially capturing its key characteristics. These vector embeddings are then stored in a specialized database like Single Store DB.

When a user seeks information, the vector database utilizes similarity searches to identify and retrieve vectors most closely resembling the query. This approach efficiently handles complex queries and delivers relevant results, regardless of data type. This entire process empowers applications requiring high-speed search and retrieval to manage vast and diverse dataset with speed and accuracy.

## LLM Model

The Vector DB sends the user query to the Large Language Model (LLM) which is the core component of the system. LLMs are trained on massive amounts of text data and are adept at generating human-like text in response to a given prompt or query.

- Let  $L(V_{similar})$  represent the Large Language Model function that generates a response based on the input vectors.
- $R = L(V_{similar})$ , where RR is the generated response.



### Response Generation

The LLM model leverages its knowledge and understanding of language to generate a response that aligns with the user's query and the conversation history. The response is a sequence of tokens that the LLM determines is the most likely and relevant continuation of the dialogue.

- Let  $G(P, R)$  represent the function that combines the prompt and the generated response to create the final response.

### ML Model (MLOps)

Deploying the chatbot model to a production environment using MLOps tools for version control, infrastructure management, and scalable operation. And continuously monitor the chatbot's performance in real-world scenarios. By collecting user feedback, analyse interaction logs, and track key metrics.

- Let  $M(R_{final})$  represent the MLOps function that handles deployment, monitoring, and updating of the model.

### Generated Response

The final response is delivered back to the user through the chatbot interface, completing the conversation cycle. Combining all steps, the complete mathematical formula for the process is: Where:

- $U$  is the user input.
- $E$  is the embedding function.
- $V$  is the vector representation of the user input.
- $P$  is the prompt extraction function.
- $Q$  is the question vector extraction function.
- $C$  is the cleaning function.
- $T$  is the tokenization function.
- $F$  is the function that interacts with the Vector DB.
- $L$  is the Large Language Model function.
- $G$  is the response generation function.
- $M$  is the MLOps function.
- $R_{user}$  is the final response delivered to the user.

### Pseudo code - Algorithm

**Data Preparation:** Let  $D = \{d_1, d_2, d_n\}$  be the set of uploaded documents (resumes, job descriptions, etc.)  $T: D \rightarrow D'$  is the preprocessing function  $D' = \{T(d_1), T(d_2), \dots, T(d_n)\}$  is the set of preprocessed documents

**Feature Extraction:**  $X = [x_1, x_2, x_m] \in \mathbb{R}^{m \times n}$ , where each  $x_i$  is a feature vector extracted from a preprocessed document

**Logistic Regression Model:** Let  $\theta = [\theta_0, \theta_1, \theta_n]^T$  be the model parameters The logistic function:



$\Sigma(z) = 1 / (1 + e^{-z})$  for a single input  $x$ :  $P(y = 1|x; \theta) = \sigma(\theta^T x) = 1 / (1 + e^{-\theta^T x})$

**Model Training:** Objective function (log-likelihood):

$$L(\theta) = \sum_i [y_i \log(\sigma(\theta^T x_i)) + (1 - y_i) \log(1 - \sigma(\theta^T x_i))]$$

Optimize  $\theta$  to maximize  $L(\theta)$  using gradient descent or other optimization methods

**Prediction:** For a new input  $x_{\text{new}}$ :

$$\hat{y} = \sigma(\theta^T x_{\text{new}})$$

**Model Evaluation:** Model evaluation is a crucial part of the MLOps lifecycle, ensuring model accuracy, reliability, and value in production environments, providing insights for continuous improvement and maintenance.

**Accuracy:** This statistic computes the percentage of true positive and true negative predictions among all made predictions, indicating the model's overall soundness.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

**Precision:** Positive predictive value, or precision, is a metric that expresses the percentage of actual positive predictions among all positive predictions generated by the model. Given by the following formula:

$$\text{Precision} = \frac{\text{True Positives}(TP)}{\text{True Positives}(TP) + \text{False Positives}(FP)}$$

High precision indicates a low false positive rate, which is important in applications where the cost of false positives is high.

**Recall:** The percentage of true positive predictions among all real positive occurrences is called recall, which is sometimes referred to as sensitivity or true positive rate. Given by the following formula

$$\text{Recall} = \frac{\text{True Positives}(TP)}{\text{True Positives}(TP) + \text{False Negatives}(FN)}$$

High recall is important in these applications where missing positive studies (false negatives) are costly.

**F1-Score:** The F1-Score is the harmonic mean of precision and recall, providing a single metric that balances both. It is particularly useful when you need to balance precision and recall and when the class distribution is uneven. The F1-Score is given by the formula:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1-Score ranges from 0 to 1, with 1 being the best possible score

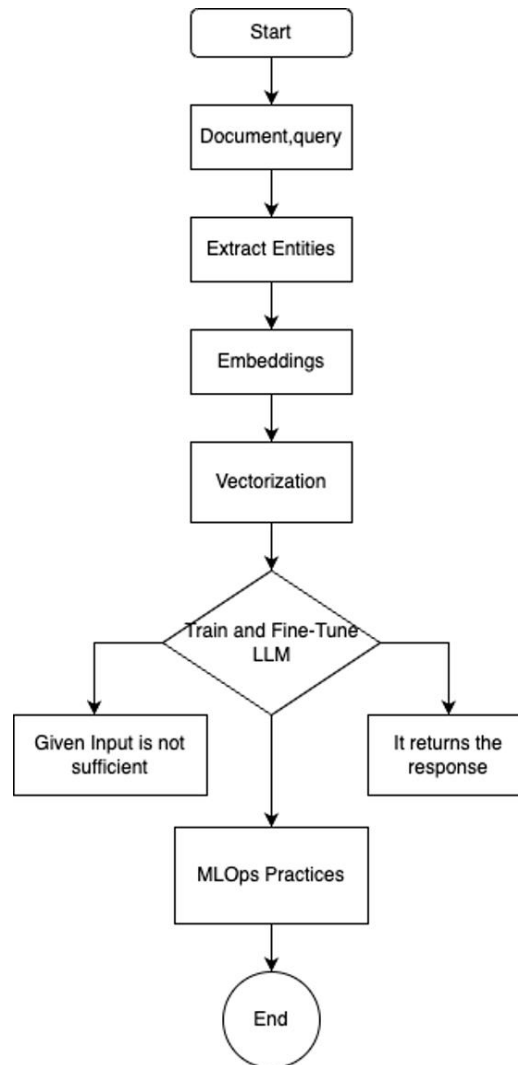


### MLOps Integration

- Version control: Track model versions  $v_1, v_2, \dots, v_k$
- Model registry: Store models  $M_1, M_2, \dots, M_k$  corresponding to versions
- Continuous monitoring: Track performance metrics  $p_1, p_2, \dots, p_n$  over time
- Retraining trigger: If any  $p_i < \text{threshold}_i$ , initiate retraining

### Flow Paths

In this research, our chatbot system follows a structured interaction flow designed to enhance user engagement and the process includes conditional checks for query and answer validation. These checks either allow the flow to proceed to the next step or redirect to a default message (validation or error) if the conditions are not met.



**Figure 3: Process Flowchart**



The process initiates when the user accesses the HR chatbot system. Depending on the user input, the system branches into two main paths: uploading PDF files or inputting a query. When PDFs are uploaded, the system first extracts text from these documents using libraries such as pdfminer, high\_level and PyPDF2. The extracted text is then analyzed to identify and extract relevant entities including names, contact information, skills, and educational qualifications. This text is subsequently divided into manageable chunks using langchain.text\_splitter for efficient processing.

The text chunks are converted into vector representations and stored in a vector database, facilitating efficient retrieval. A Large Language Model (LLM) is trained and fine-tuned using the text data from the PDFs. This model, alongside MLOps practices for model management, ensures efficient development, deployment, and continuous improvement of the system.

When a user submits a query related to the resumes, the system processes the query using the trained LLM and the vector store to locate relevant information. The results, including the score and message related to the query, are then displayed to the user. The process concludes once the information is presented.

The system utilizes various libraries and tools for Natural Language Processing (NLP) and machine learning. Streamlet is used to build a user interface for interaction, while spaCy aids in named entity recognition. Scikit-learn supports model training, and mlflow manages the machine learning lifecycle. For text embeddings and retrieval, google.generativeai and langchain\_google\_genai integrate Google's generative AI services, and langchain.vectorstores with FAISS is employed for efficient vector storage and similarity search. Regular expressions from the re module are utilized for extracting specific information such as contact details and skills from the resume text.

#### **4 Results and Discussion**

The HR Bot for internal communication showed significant improvements in metrics by extracting categorized skills from resumes, as seen in the "Resume Information Extraction" interface. The system configuration included MLOps, Lang Chain, NLP, and other machine learning tools. Compared to other models, the HR Bot's performance was superior due to its precise skill extraction and classification capabilities, making it the best model for resume analysis in this context.

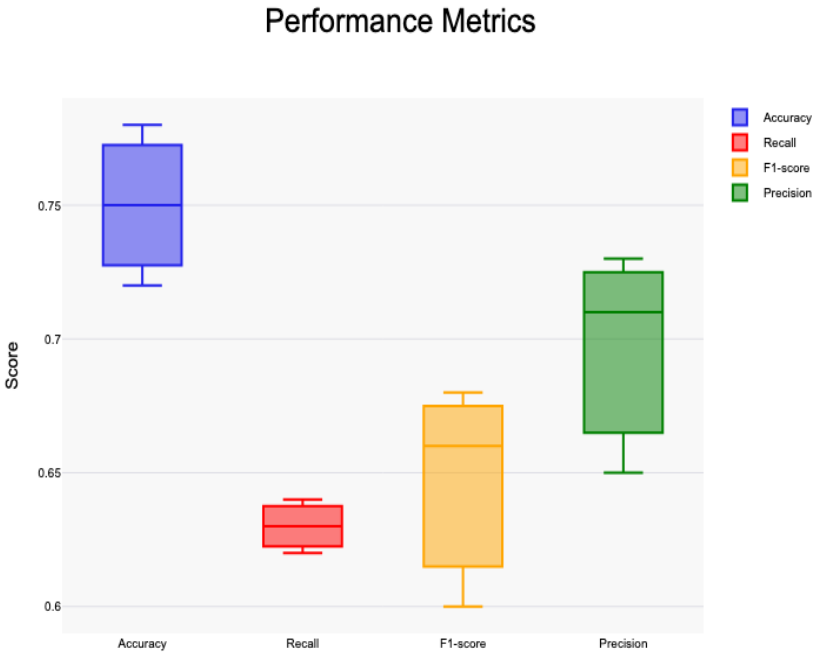


Figure 4 : Performance Metrics

Table 1: Metrics

| Algorithm: Random Forest | App1 | App2 | App3 |
|--------------------------|------|------|------|
| Accuracy                 | 0.78 | 0.75 | 0.72 |
| Recall                   | 0.64 | 0.62 | 0.63 |
| F1-score                 | 0.68 | 0.66 | 0.60 |
| Precision                | 0.73 | 0.71 | 0.65 |

5 Conclusions

Implementing an AI-driven HR chatbot that leverages LLMs and MLOps practices offers significant advantages in terms of efficiency, accuracy, and scalability. By automating resume analysis and query responses, HR departments can streamline the operations, reduce manual workload, and enhance the overall recruitment process. This use case highlights then potential of integrating advanced AI technologies to revolutionize traditional HR functions. It provides 24/7 support, ensuring that HR services are available at all times. The chatbot's continuous learning capabilities allow it to adapt to evolving HR needs, maintaining its relevance and effectiveness. It empowers HR teams to focus on strategic initiatives that drive organizational growth.



## 6 Future Work

To enhance the HR chatbot's capabilities, several key improvements can be made. First, implementing advanced query handling techniques will enable the chatbot to manage complex, multi-turn conversations, improving user experience by allowing for a more natural and continuous dialogue. Additionally, integrating the chatbot with existing HR management systems will facilitate seamless data exchange and automate various HR processes, enhancing operational efficiency. Finally, security enhancements are essential to protect sensitive user information, with robust data privacy measures in place to ensure that all user interactions remain secure and confidential.

## References

1. Murugesan; Subramanian; Srivastava; Dwivedi, Year: 2023, "A study of Artificial Intelligence impacts on Human Resource Digitalization in Industry 4.0", IEEE Transactions on Industry Human Resource, Vol: 7, pp. 100249.
2. Kimseng; Javed; Jeenanunta; Kohda, Year: 2020, "Applications of fuzzy logic to reconfigure human resource management practices for promoting product innovation in formal and non-formal RandD firms", J. Open Innov.: Technology Market and Complexity, Vol: 6, no: 38.
3. Masum; Beh; Azad; Hoque, Year: 2018, "Intelligent human resource information system (i-HRIS): a holistic decision support framework for HR excellence", Int. Arab J. Inf. Technol, Vol: 15, no: 1, pp. 121–130.
4. Reddy; Kumar; Rangaiah, Year: 2019, "Artificial Intelligence (AI) in learning and development: A conceptual paper", IEEE Transactions on Industry Human Resource, Vol: 38, no: 1, pp. 34–49.
5. Bhardwaj; Singh; Kumar, Year: 2020, "An empirical study of artificial intelligence and its impact on human resource functions", IEEE Conference on Computation, Automation and Knowledge Management, pp. 47–51.
6. Oswald; Behrend; Putka; Sinar, Year: 2020, "Big data in industrial organizational psychology and human resource management: Forward progress for organizational research and practice", Vol: 7, pp. 505–533.
7. Gun Ho Lee; Koyung Jun Le; et al, Year: 2023, "Developing Personalized Marketing Service Using Generative AI", IEEE Explore, Vol: 12, pp. 22394-3536.
8. Maddigan; Susnjak, Year: 2022, "Chat2VIS: Generating Data Visualizations via Natural Language Using ChatGPT, Codex and GPT-3 Large Language Models", IEEE Explore, Vol: 11.
9. Kreuzberger; Kuhl; N, Year: 2021, "Machine Learning Operations (MLOps): Overview, Definition, and Architecture", IEEE Explore, Vol: 11, pp. 31866-31879.



10. Garg; et al, Year: 2021, “On Continuous Integration / Continuous Delivery for Automated Deployment of Machine Learning Models using MLOps”, IEEE Explore, pp. 25-28.
11. Jeong; Cheonus, Year: 2023, “A Study on the Implementation of Generative AI Services Using an Enterprise Data-Based LLM Application Architecture”, Research Gate, pp. 2309-01105.
12. Chen; Tzu-Yu; Yu-Ching Chiu; Nanyi Bi; Richard Tzong-Han Tsai, Year: 2021, “Multi-Modal Chatbot in Intelligent Manufacturing”, IEEE Explore, Vol: 9, pp. 82118 – 82129.
13. Singh; Ishika; Valts Blukis; Arsalan Mousavian; Ankit Goyal; Danfei Xu; Jonathan Tremblay; Dieter Fox; Jesse Thomason; Animesh Garg, Year: 2023, “ProgPrompt: Generating Situated Robot Task Plans using Large Language Models”, IEEE Explore, pp. 11523-11530