Generative AI

Course Name: Generative AI	**CourseCode**	**L-T-P**	**Credits**
	ETMCGI274	3-0-2	4
Type of Course:	Major		
Pre-requisite(s), if any:	Prior knowledge of Python programming, linear algebra, probability, machine learning fundamentals, and deep learning basics.		

Course Perspective. This course introduces the foundations and applications of Generative Artificial Intelligence (GenAI), with a focus on deep generative models such as Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), and Transformers. It equips students with practical skills for building GenAI models, applying prompt engineering, fine-tuning language models, and evaluating outputs across modalities like text, images, and audio.

The Course Outcomes (COs). On completion of the course the participantswill be able to:

COs	Statements
CO 1	Understanding the theoretical foundations and architectures of generative models including GANs, VAEs, and Transformers.
CO 2	Designing and implementing GenAI applications for text, image, and code generation using industry frameworks.
CO 3	Fine-tuning and deploying foundation models using prompt engineering, transfer learning, and open-source APIs.

CO 4	Evaluating generative models using performance metrics, interpretability tools, and ethical guidelines for responsible AI.

CO = Course outcomes. A student is expected to have learnt conceptsand demonstrated/developed abilities or skills related to strategic management at the end of the course.

Course Outline:

Unit Number: 1	**Foundations of Generative Models**	**No. of hours: 12**
Topic Covered: • Generative vs Discriminative models • Latent variable models and sampling • Variational Autoencoders (VAE) – architecture, loss functions • Generative Adversarial Networks (GANs) – generator, discriminator, loss functions, stability • **Real-World Use Case:** Generating new product designs using GANs and VAEs		
Unit Number: 2	**Large Language Models and Transformers**	**No. of hours: 11**
Topic Covered: • Transformer architecture: self-attention, multi-head attention • BERT vs GPT models: masked vs autoregressive • Pretraining and fine-tuning workflows • Introduction to OpenAI, Hugging Face Transformers, and LLaMA • **Real-World Use Case:** Auto-generating technical documentation from code repositories using LLMs		

Unit Number:3	Prompt Engineering and Foundation Model Tuning	No. of hours: 11

Topic Covered:

- Prompt types: zero-shot, few-shot, chain-of-thought
- Instruction tuning and reinforcement learning from human feedback (RLHF)
- Fine-tuning LLMs using parameter-efficient methods (LoRA, adapters)
- Embeddings and retrieval-augmented generation (RAG)
- **Real-World Use Case:** Chatbots for customer service fine-tuned on organizational data

Unit Number:4	Evaluation, Ethics, and Multimodal GenAI	No. of hours: 11

Topics Covered:

- Evaluation metrics: BLEU, FID, perplexity, accuracy
- Hallucination, bias, and interpretability in GenAI
- Copyright, fairness, and ethical AI practices
- Introduction to text-to-image (DALL·E, Stable Diffusion) and audio generation (Voice Cloning, MusicLM)
- **Real-World Use Case:** Evaluating the factuality and safety of AI-generated news summaries

Text & Reference Books

- Ian Goodfellow et al. – *Deep Learning*
- Sebastian Raschka – *Machine Learning with PyTorch and Scikit-Learn*
- Hugging Face Course – *https://huggingface.co/learn*
- Jason Brownlee – *Generative Adversarial Networks with Python*

Learning Outcomes

Inside the Classroom

1. **Understanding Core Concepts:**
    - Explain the fundamentals of Generative AI, including GANs, VAEs, Diffusion Models, and Transformer-based architectures.
    - Differentiate between discriminative and generative models.
2. **Mathematical and Algorithmic Foundations:**
    - Apply probability, statistics, and linear algebra in the context of generative models.
    - Implement and analyze loss functions like adversarial loss, reconstruction loss, and KL divergence.
3. **Model Development Skills:**
    - Design, train, and evaluate generative models using frameworks like TensorFlow or PyTorch.
    - Tune hyperparameters for optimal generation performance.
4. **Ethical and Responsible AI Practices:**
    - Discuss bias, misinformation, and copyright concerns associated with generated content.
    - Evaluate the ethical use of generative models in different domains.
5. **Hands-on Experiments:**
    - Generate images, text, or music using trained models in lab sessions.
    - Conduct comparative performance analysis between different generative techniques.

Outside the Classroom

1. **Practical Applications:**
    - Apply generative AI in real-world scenarios such as content creation, code generation, image synthesis, and medical imaging.
    - Use tools like ChatGPT, DALL·E, Midjourney, or RunwayML to create innovative content.
2. **Collaborative Projects:**
    - Work in teams to develop generative AI-based applications for hackathons or academic projects.
    - Solve open-ended problems using generative techniques with minimal supervision.
3. **Research and Innovation:**
    - Explore recent academic papers, blogs, and case studies to stay current with breakthroughs like Sora, GPT-4o, and diffusion-based systems.
    - Contribute to open-source projects or Kaggle competitions involving generative tasks.
4. **Ethical Reflection and Community Impact:**
    - Reflect on the societal impact of AI-generated content and propose responsible use guidelines.
    - Participate in seminars/webinars or discussions on the future of generative AI.
5. **Portfolio and Career Development:**
    - Build a personal portfolio with generative AI projects (e.g., AI-generated art, AI-written poetry, or synthetic data generation).
    - Gain exposure to industry use cases and prepare for AI/ML roles in tech, media, gaming, healthcare, or marketing.

K.R. MANGALAM UNIVERSITY

Lab Assignment

S.No	Lab Tasks
1	**LAB TASK 1: Generating Digits with VAE and GAN (MNIST)** **Objective:** To train a VAE and GAN on image datasets and visualize generated outputs. **Activities:** • Implement a VAE and GAN using PyTorch/TensorFlow • Train on MNIST or Fashion-MNIST dataset • Visualize latent space and generated samples **Learning Focus:** Deep generative modeling, latent vector manipulation **Tools:** Python, PyTorch, Matplotlib, TensorBoard
2	**LAB TASK 2: Building a Text Generator with Transformers** **Objective:** To generate coherent sentences using transformer-based language models. **Activities:** • Load pretrained GPT-2 using Hugging Face • Generate text using top-k sampling and temperature control • Experiment with prompt templates **Learning Focus:** Transformer inference, text generation, sampling strategies **Tools:** Hugging Face Transformers, Python, Google Colab
3	**LAB TASK 3: Prompt Engineering and LLM Fine-tuning**

	Objective: To apply prompt engineering strategies and fine-tune a small language model. Activities: • Test different prompt templates (zero-shot, CoT, few-shot) • Fine-tune a T5 model using LoRA on a custom dataset • Compare performance of tuned vs untuned models **Learning Focus:** Prompt crafting, few-shot learning, tuning best practices **Tools:** PEFT (Parameter Efficient Fine-Tuning), Hugging Face, Google Colab
4	**LAB TASK 4: Text-to-Image Generation and Evaluation** **Objective:** To create visual content from natural language prompts using diffusion models. **Activities:** • Generate images using Stable Diffusion / DALL·E • Evaluate image quality using FID or human feedback • Compare outputs from different models **Learning Focus:** Multimodal generation, prompt engineering, model comparison **Tools:** Stable Diffusion API, OpenAI DALL·E, Python
5	**Capstone Project: Domain-Specific Generative Assistant** **Objective:** To design, train, and deploy a domain-specific assistant (e.g., legal, healthcare, academic) using LLMs with prompt optimization and retrieval-augmented generation. **Project Tasks:**

	• Integrate open-source LLM (e.g., LLaMA or GPT-J) with embeddings and RAG • Collect domain-specific documents for knowledge grounding • Design advanced prompts and test different prompting strategies • Evaluate output for accuracy, bias, and factual grounding **Learning Focus:** LLM orchestration, RAG, prompt tuning, GenAI app delivery **Tools:** LangChain, OpenAI API / LLaMA.cpp, ChromaDB / FAISS, Streamlit / Flask