Shri Vile Parle Kelavani Mandal's

Dwarkadas J. Sanghvi College of Engineering

(Autonomous College Affiliated to the University of Mumbai)

Scheme and detailed syllabus

Final Year B.Tech

in

Artificial Intelligence and Machine Learning

(Semester VII/VIII)

Academic Year 2026-27

Continuous Assessment (A):

Course	Assessment Tools	Marks	Time (mins)
Theory	a. Term Test 1 (based on 40 % syllabus)	15	45
	b. Term Test 2 (on next 40 % syllabus)	15	45
	c. Assignment / course project / group discussion /presentation / quiz/ any other.	10	--
	Total marks (a + b + c)	40	
Audit course	Performance in the assignments / quiz / power point presentation / poster presentation / group project / any other tool.	--	As applicable
Laboratory	Performance in the laboratory and documentation.	25	
Tutorial	Performance in each tutorial & / assignment.	25	
Laboratory & Tutorial	Performance in the laboratory and tutorial.	50	

Continuous Assessment (B):

Course	Assessment Tools	Marks	Time (hrs.)
Theory / * Computer based	Written paper based on the entire syllabus.	60	2
	* Computer-based assessment in the college premises.		
Oral	Questions based on the entire syllabus.	25	As applicable
Practical	Performance of the practical assigned during the Examination and the output / results obtained.	25	2
Oral & Practical	Project based courses - Performance of the practical assigned during the examination and the output / results obtained. Based on the practical performed during the Examination and on the entire syllabus.	As per the scheme	2

Prepared by | Checked by | HoD | Vice Principal | Principal

Program: Artificial Intelligence & Machine Learning	B. Tech.	Semester: VII/VIII
Course: Reinforcement and Federated Learning (DJS23ACPC701)		
Course: Reinforcement and Federated Learning Laboratory (DJS23ALPC701)		

Prerequisite: Machine Learning, Knowledge of Probability, Linear Algebra, and Calculus, Python programming and deep learning frameworks (TensorFlow / PyTorch).

Course Objectives:
To provide a comprehensive understanding of Reinforcement Learning (RL) principles, algorithms, and applications, including multi-armed bandits, dynamic programming, temporal difference learning, and deep RL. Students will acquire practical skills in model-free and model-based RL, policy evaluation, and function approximation using neural networks. The course covers real-world applications in robotics, gaming, recommender systems, and industry. It also introduces Federated Learning (FL) fundamentals for privacy-preserving distributed learning. Learners will explore the integration of RL and FL in practical scenarios.

Course Outcomes: Students will be able to:

1. Explain the fundamental concepts of reinforcement learning, including agents, environments, rewards, policies, and the mathematical foundations such as value functions and Bellman equations.
2. Understand Markov Decision Processes (MDPs) and multi-armed bandit problems using dynamic programming, Monte Carlo methods, and temporal difference learning.
3. Evaluate deep reinforcement learning algorithms, including Deep Q-Networks (DQN) and Policy Gradient methods, for practical problems in games, robotics, and industrial applications.
4. Understand the principles of federated learning, including FedAvg, privacy mechanisms, and communication challenges, and analyze potential applications in healthcare, finance, and personalized AI systems.

Reinforcement and Federated Learning (DJS23ACPC701)		
Unit	**Description**	**Duration**
1	**Introduction to Reinforcement Learning**: Introduction to reinforcement learning. RL Elements: Agent, Environment, Rewards, States, Actions, Policy, History and applications (Games, Robotics, Recommenders, Industry), Mathematical foundations: Rewards, returns, discounting, Value functions (V and Q), Bellman equations (include derivation), Markov Decision Processes (MDPs): States, actions, transitions, rewards, Episodic vs continuing tasks, Difference between Immediate and Full Reinforcement Learning.	7
2	**Bandit Problems**: k-armed bandit, Stochastic rewards, Action-value methods, Exploration vs Exploitation, Bandit Algorithms: ε-greedy method, Optimistic initial values, Softmax / Boltzmann exploration, Upper Confidence Bound (UCB),Thompson Sampling Relationship between MAB and RL Non-stationary bandits, Real-world applications: Online advertising: A/B testing, Recommendation systems, Clinical trials.	7

3	**Dynamic Programming & Planning**: Model-based RL, Policy evaluation, Policy improvement, Policy iteration, Value iteration, Generalized Policy Iteration (GPI), Convergence issues, Applications: Gridworld, path planning.	7
4	**Model-Free Prediction & Control**: Monte Carlo Methods, Monte Carlo prediction, First-visit vs every-visit MC, Monte Carlo control, Exploring Starts, Off-policy MC & importance sampling, Temporal Difference Methods: TD(0) prediction, SARSA (on-policy control), Q-Learning (off-policy control), Expected SARSA, Advanced TD Methods, n-step bootstrapping, Eligibility traces, TD(λ).	7
5	**Deep Reinforcement Learning**: Function approximation, Deep neural networks for RL, Deep Q-Network (DQN), Experience replay, Target networks, Variants: Double DQN, Dueling DQ, Policy Gradient Methods: REINFORCE algorithm, Proximal Policy Optimization (PPO), Actor-Critic Methods A2C and A3C, Application of Deep Reinforcement Learning in Samuel's Checkers Player, Dynamic Channel Allocation, Job-Shop Scheduling.	7
6	**Federated Learning Fundamentals, Privacy & Applications:** Federated Learning, motivation for decentralized machine learning, centralized learning vs federated learning, advantages and limitations. Federated Learning architecture: clients, servers, communication cycles, local model training, and global model aggregation. Federated Averaging (FedAvg) algorithm: basic working process and distributed learning workflow. Challenges in Federated Learning: data heterogeneity, communication overhead, device constraints, and scalability issues. Privacy and security in Federated Learning: Differential Privacy, Secure Aggregation, and privacy-preserving AI techniques. Introduction to adversarial attacks and model poisoning attacks, and a basic overview of defense mechanisms.	7
	TOTAL	**42**

Books Recommended:

Text books:

1. Soumyadip Sarkar" An Introduction to Reinforcement Learning" Notion Press, 2025.
2. Saravanan Krishnan, A. Jose Anand, R. Srinivasan, R. Kavitha, S. Suresh, "Handbook on Federated Learning: Advances, Applications and Opportunities", CRC Pr I Llc, 2025.
3. Milad Farsi, Jun Liu, Maria Domenica Di Benedetto, " Model-Based Reinforcement Learning: From Data to Continuous Actions with a Python-based Toolbox (IEEE Press Series on Control Systems Theory and Applications)", Wiley-IEEE Press, 2022
4. Richard S. Sutton and Andrew G. Barto, "Reinforcement Learning: An Introduction", MIT Press, 2nd Edition, 2020.
5. Laura Graesser Wah Loon Keng, "Foundations of Deep Reinforcement Learning," Pearson Education, 1st, Edition, 2020.

Reference Books:

1. Marcus C Lauritsen "Reinforcement Learning for Beginners: A Step-by-Step Guide to Training Smart AI Agents", Independently Published, 2025.
2. Qiang Yang , Yang Liu, "Federated Learning", Springer / BSP Books, 1st Edition, 2025.
3. George Jeno, "Federated Learning with Python: Design and implement a federated learning system and develop applications using existing frameworks" Packt Publishing, 1st Edition, 2022.
4. Phil Winder, "Reinforcement Learning Industrial Applications of Intelligent Agents", O'Reilly, 1st Edition, 2020.
5. Csaba Szepesvari, "Algorithms for Reinforcement Learning," Morgan & Claypool Publishers, 1st Edition, 2019.
6. Enes Bilgin, "Mastering Reinforcement Learning with Python", Packt publication, 1st Edition, 2020.
7. Brandon Brown, Alexander Zai, "Deep Reinforcement Learning in Action", Manning Publications, 1st, Edition, 2020.
8. Micheal Lanham, "Hands-On Reinforcement Learning for Games," Packt Publishing, 1st Edition, 2020.
9. Abhishek Nandy, Manisha Biswas, "Reinforcement Learning: With Open AI, TensorFlow and Keras using Python," Apress, 1st Edition, 2018.

Web Links:

1. NPTEL Course in Reinforcement Learning: https://onlinecourses.nptel.ac.in/noc22_cs75/preview
2. Udemy Course in Reinforcement Learning Top Reinforcement Learning Courses Online - Updated [May 2026]
3. Coursera Course in Reinforcement Learning Reinforcement Learning | Coursera
4. Reinforcement Learning Course (Stanford University): https://www.youtube.com/watch?v=FgzM3zpZ55o
5. AI Games with Deep Reinforcement Learning: https://towardsdatascience.com/how-to-teach-an-ai- to-play-games-deep-reinforcement-learning-28f9b920440a
6. Deep Reinforcement Learning: https://www.v7labs.com/blog/deep-reinforcement-learning-guide
7. TensorFlow Federated documentation: https://www.tensorflow.org/federated
8. OpenAI Gym documentation: https://gym.openai.com
9. Federated Learning What Is Federated Learning? | IBM
10. Federated Learning https://blogs.nvidia.com/blog/what-is-federated-learning/
11. Federated Learning 8 Innovative applications of federated learning across the globe - OpenSistemas

Reinforcement and Federated Learning Laboratory (DJS23ALPC701)	
Sr. No	**List of Suggested Experiments**
1	Interacting with an OpenAI Gym Environment (CartPole-v1).
2	Simulate and Model a simple environment (e.g., Gridworld or FrozenLake) as a Markov Decision Process (MDP) by Defining States, Actions, Transitions, and

	Rewards.
3	Implement a simple MDP and visualize transitions. (Python, Matplotlib)
4	Implement iterative policy evaluation using Bellman expectation equations. (Python, Numpy)
5	Implement Monte Carlo Prediction and Control for a small environment.
6	Implement TD Learning (SARSA and Q-Learning) on a GridWorld environment.
7	Comparison of On-policy and Off-policy learning methods: CliffWalking-v0.
8	Develop a Deep Q-Network (DQN) using PyTorch/TensorFlow CartPole-v1.
9	Implement Policy Gradient and REINFORCE algorithm.
10	Actor–Critic model implementation on a continuous control problem.
11	Introduction to Federated Learning framework (TensorFlow Federated / PySyft).
12	Use federated transfer learning when clients have different feature spaces or label spaces using PyTorch/TensorFlow, heterogeneous datasets (e.g., MNIST & SVHN or image + tabular splits)
13	Implement Federated Averaging (FedAvg) algorithm for MNIST dataset.
14	Evaluate privacy-preserving mechanisms (Differential Privacy in FL).

Minimum ten experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Prepared by Checked by HOD Principal

Program: Artificial Intelligence and Machine Learning	B.Tech.	Semester: VII/VIII
Course: AI in Healthcare (DJS23ACPE711)		
Course: AI in Healthcare Laboratory (DJS23ALPE711)		

Pre-requisite:

- Knowledge of Machine Learning (Supervised/Unsupervised Learning).
- Understanding of Deep Learning and Neural Networks.
- Familiarity with Data Science tools (Python, TensorFlow/PyTorch).

Course Objectives:

1. To understand the role of AI in transforming healthcare.
2. To learn how to apply AI techniques such as machine learning, deep learning, and natural language processing (NLP) to healthcare data.
3. To gain hands-on experience in working with real healthcare datasets.
4. To discuss ethical, privacy, and regulatory concerns related to AI in healthcare.

Course Outcomes: Upon completion of this course, students will be able to:

1. Identify and evaluate various types of healthcare data (e.g., EHR, medical imaging, genomic data) and assess challenges like data privacy, bias, and regulatory constraints.
2. Implement machine learning algorithms (classification, clustering, time-series analysis) for predictive analytics, diagnostics, and clinical decision support systems using healthcare datasets.
3. Design and implement deep learning architectures (CNNs, RNNs) for medical image classification, segmentation, and time-series health data analysis, enhancing diagnostic accuracy.
4. Use AI tools to aid in drug discovery and optimize personalized medicine approaches by analyzing genomic data and predicting patient responses to treatments.
5. Discuss and address ethical challenges, privacy issues, and the legal framework surrounding the deployment of AI systems in clinical settings.

AI in Healthcare (DJS23ACPE711)		
Unit	**Description**	**Duration**
1	**Introduction to AI in Healthcare:** Overview of AI in Healthcare: Historical perspective, Types of AI systems in healthcare, Role of AI in healthcare transformation. Key Healthcare Domains for AI Applications: Medical imaging, Diagnostics, Drug discovery, Personalized medicine. Challenges in Healthcare Data: Data	06

	variability, noise, and incompleteness, Data privacy and security concerns, Regulatory constraints.	
2	**Machine Learning in Healthcare:** Introduction to Healthcare Data: Electronic Health Records (EHR), Medical imaging data (X-rays, MRIs, CT scans), Genomics and clinical trial data. Machine Learning Techniques: Classification and regression for disease prediction, Clustering for patient stratification, Time series analysis for health monitoring. **Use cases**: Predictive analytics, clinical decision support systems	**08**
3	**Deep Learning in Healthcare:** Introduction to Deep Learning Architectures: Convolutional Neural Networks (CNNs) for medical imaging, Recurrent Neural Networks (RNNs) for time-series health data, Auto encoders for anomaly detection. Applications in Medical Imaging: Image classification, segmentation, and detection in radiology. Natural Language Processing (NLP) in Healthcare: Information extraction from medical literature and EHRs, NLP applications: Chatbots, virtual assistants, voice recognitions. **Introduction to Federated Learning in Healthcare data**	**08**
4	**AI in Drug Discovery and Personalized Medicine:** AI-Driven Drug Discovery: Introduction to computational drug discovery and AI Case studies: AI in accelerating drug development. AI for Personalized Medicine: AI-based genomic analysis, Predicting patient responses to therapies, Tailoring treatment plans using AI models.	**08**
5	**AI Ethics, Privacy, and Regulations in Healthcare:** Ethical Considerations in AI for Healthcare: Bias in AI models and its implications, AI decision-making in healthcare. Privacy and Security in Healthcare AI: Handling sensitive healthcare data, Regulations: HIPAA, GDPR, and their implications for AI. Regulatory Aspects and AI Approval: AI in clinical trials and FDA approvals.	**07**
6	**Emerging Trends of AI in Healthcare:** Wearable AI and Remote Monitoring: AI for patient monitoring using IoT and wearables. AI in Robotics and Surgery: AI-assisted surgeries, robotic platforms, and automation. Telemedicine and AI Integration: AI-driven remote consultations and diagnostics.	**05**
	TOTAL	**42**

Books Recommended:

Text Books:

1. Bohr, A., & Memarzadeh, K. (2020). *Artificial intelligence in healthcare.* ScienceDirect / Academic Press.
2. Yang, H., & Lee, E. K. (2016). *Healthcare analytics: From data to knowledge to healthcare improvement.* Wiley.

3. Zhou, S. K., Greenspan, H., & Shen, D. (2017). *Deep learning for medical image analysis*. Academic Press.

***Reference Books*:**

1. Floridi, L., & Taddeo, M. (Eds.). (2021). Ethics of artificial intelligence in healthcare. Springer / ACM.
2. (Note: The exact editor names may vary slightly depending on the specific volume; Springer publishes multiple editions on AI ethics.)
3. Topol, E. (2019). Deep medicine: How artificial intelligence can make healthcare human again. Basic Books.
4. Frangi, A. F., et al. (2023). Medical image analysis. Elsevier.

Research Papers

Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H. R., Albarqouni, S., … & Cardoso, M.J. (2020). *The future of digital health with federated learning*. npj Digital Medicine, 3(119).
Springer. (2020). *Internet of Medical Things (IoMT) in healthcare*. Springer Publishing.

Online Resources:

1. https://ai.stanford.edu/
2. https://drerictopol.com/tag/medical-ai/
3. https://medicalfuturist.com/
4. https://blogs.nvidia.com/blog/tag/healthcare-life-sciences/
5. https://www.coursera.org/specializations/ai-for-medicine

Suggested List of Experiments:

AI in Healthcare Laboratory (DJS23ALPE711)	
Sr. No.	**Title of the Experiment**
1	Perform data preprocessing (cleaning, normalization) on an EHR dataset
2	Implement classification models (e.g., Decision Trees, Random Forest) to predict disease outcomes using structured healthcare data.
3	Apply clustering techniques (e.g., K-Means, DBSCAN) for patient stratification.
4	Use time-series analysis (e.g., LSTM) for health monitoring, predicting patient vitals over time.
5	Develop a Convolutional Neural Network (CNN) for medical image classification.
6	Implement U-Net or similar architecture for segmentation of medical images (e.g., tumor localization).
7	Use NLP techniques to extract insights from clinical notes or research papers.
8	Perform AI-assisted drug discovery using molecular structure datasets.
9	Use machine learning to analyze genomic data for personalized treatment recommendations.

10	Evaluate bias in AI models trained on healthcare data, and propose solutions to mitigate it.
11	Implement machine learning models to analyze sensor data (e.g., Fitbit) for real-time health monitoring.
12	Use social media, clinical reports, and mobility data to predict disease outbreaks.
13	Review of Research Articles
14	Case Study: Medical Regulations and Compliances

Minimum ten experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt

Prepared by Checked by HoD Principal

Program: Artificial Intelligence & Machine Learning	B.Tech.	Semester: VII/VIII
Course: Multimodal Generative AI(DJS23ACPE712)		
Course: Multimodal Generative AI Laboratory (DJS23ALPE712)		

Pre-requisite: Deep Learning, Large Language Models

Course Objectives:

1. To provide students with a thorough grasp of image generative AI, including its historical context, key technologies, and applications across various industries.
2. To Equip students with hands-on experience in image generation techniques, manipulation, and evaluation methods.
3. Enable students to explore and analyze the diverse applications of image generative AI in fields such as art, healthcare, and entertainment.
4. Foster an understanding of the ethical considerations and societal impacts of image generation technologies, preparing students to address potential challenges and biases in the field.

Course Outcomes: Students will be able to

1. Understand the principles of Image Generation and applications of Image Generation
2. Compare and contrast various generative models, including GANs and VAEs, and apply their architectures to real-world problems.
3. Demonstrate proficiency in using diffusion models and vision transformers in generating images and utilize vision language models.
4. Understand and apply techniques for video generation

Multimodal Generative AI (DJS23ACPE712)		
Unit	**Description**	**Duration**
1	**Introduction and Applications of Image Generation:** Overview of Image Generation, Historical Context and Evolution, Challenges, Types of Image Generation Techniques, Applications in Various Industries: Art and Design, Image-to-Image Translation, Super-Resolution, Face Generation, Deepfakes, 3D Image Generation, Overview of Generative Model Families: Likelihood-based, Implicit, Diffusion, and Energy-based models, Evaluation Metrics for Generated Images : FID, PSNR, SSIM	7
2	**Generative Adversarial Networks (GANs) and Variants:** Architecture of GANs: Generator and Discriminator, Adversarial Training, Variants of GANs (e.g., DCGAN, StyleGAN), BigGAN, Pix2Pix, GauGAN	7
3	**Generative and Representation Learning Architectures for Images** Introduction to Variational Autoencoders (VAEs), -Variational inference and reparameterization trick. Conditional VAEs (CVAE), Vector Quantized VAE	7

	(VQ-VAE, VQ-VAE-2). **Representation Learning Architectures :**Neural Style Transfer (NST), Siamese Networks and UNET Architecture	
4	**Diffusion Models and Modern Image Generation:** Introduction to Diffusion Models, Concepts and architecture overview , De-noising Diffusion Probabilistic Model, Stochastic Differential Equation (SDE), Diffusion Probabilistic Models: Concept & Architecture (DALL-E, Stable Diffusion,Imagen), Comparison: GANs vs VAEs vs Diffusion Models, Hybrid Diffusion Techniques: Latent Diffusion, VQ-Diffusion. Flow based models, ControlNet.	**8**
5	**Vision Language Models and Image Generation:** Introduction to Vision Language Models (VLMs), Transformer-Based Captioning Models :Contrastive Language Image Pairs (CLIP), Vision-and-Language Transformer(ViLT), DeepSeek-VL2, Gemma 3.	**8**
6	**Video Generation Models:** Video GAN (VGAN) , MoCoGAN: Decomposing Motion and Content for Video Generation, Video Diffusion Models : TATS (Text-to-Video via Diffusion) model, Video LDM (High-Resolution Video Synthesis with Latent Diffusion Models)	**6**
	TOTAL	**42**

Books Recommended:

Text Books:

1. Denis Rothman," Transformers for Natural Language Processing- Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more" , Second Edition, Packt Publishing, 2023.
2. Zonunfeli Ralte, Indrajit Kar, "Learn Python Generative AI: Journey from autoencoders to transformers to large language models", First Edition 2024, ISBN: 978-93-55518-972.
3. Soon Yau Cheong, "Hands-On Image Generation with TensorFlow: A practical guide to generating images and videos using deep learning", Packt Publishing, 2020.
4. David Foster, "Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play", O'Reilly Media, Inc. June 2019, ISBN: 9781492041948.

Reference Books:

1. Martin Yanev, "Building AI Applications with OpenAI APIs: Leverage ChatGPT, Whisper, and DALL-E APIs to build 10 innovative AI projects, Second Edition" Packt Publishing, 2024.
2. Amita Kapoor, Antonio Gulli, Sujit Pal, "Deep Learning with TensorFlow and Keras – 3rd edition: Build and deploy supervised, unsupervised, deep, and reinforcement learning models" , Third Edition, Packt Publishing, 2022.
3. V Kishore Ayyadevara, Yeshwanth Reddy, "Modern Computer Vision with PyTorch: Explore deep learning concepts and implement over 50 real-world image applications" Packt Publishing, 2020.

4. Martinez, "TensorFlow 2.0 Computer Vision Cookbook: Implement machine learning solutions to overcome various computer vision challenges", 1st Edition, " Packt Publishing, 2021.

Web Links:

1. https://www.tensorflow.org/tutorials/generative/style_transferhttps://openai.com/index/clip/
2. https://www.coursera.org/specializations/deep-learning
3. https://medium.com/@zhonghong9998/neural-style-transfer-creating-artistic-images-with-deep-learning-803409fc64c0
4. https://medium.com/@outerrencedl/a-simple-autoencoder-and-latent-space-visualization-with-pytorch-568e4cd2112a
5. https://pyimagesearch.com/2020/03/30/autoencoders-for-content-based-image-retrieval-with-keras-and-tensorflow/
6. https://realpython.com/generative-adversarial-networks/
7. https://towardsdatascience.com/reparameterization-trick-126062cfd3c3
8. https://shashank7-iitd.medium.com/understanding-vector-quantized-variational-autoencoders-vq-vae-323d710a888a

Suggested List of Experiments:

Multimodal Generative AI Laboratory (DJS23ALPE712)	
Sr. No.	**Title of the Experiment**
1	Implement Neural Style Transfer on a set of images and compare results based on different styles.
2	Encode a set of images using autoencoders and visualize the latent space representation.
3	Evaluate the quality of generated images using different metrics (e.g., FID, PSNR).
4	Build and train a basic GAN, observing the training process and quality of generated images.
5	Explore different variants of GANs (e.g., DCGAN, StyleGAN) and report on their performance on a common dataset.
6	Implement and compare a Variational Autoencoder with and without the reparameterization trick.
7	Create and evaluate a hybrid model using VQ-VAE-2 and diffusion techniques.
8	Analyze the architecture of a diffusion model (e.g., DALL-E, stable diffusion) and generate images using available pre-trained models.
9	Explore the application of transformers in image generation and implement a basic transformer model for image tasks.
10	Utilize CLIP to generate text-image pairs and evaluate their relevance and accuracy.
11	Train a U-Net for semantic segmentation on a medical or natural image dataset. Also show how U-Net encoder features can be used for representation learning (e.g., feature extraction for classification).
12	Compare Transformer captioners (ViLT) vs. CLIP-guided prefix tuning vs. Gemma-style multimodal decoder. Evaluate how (a) direct vision-language transformers

	(ViLT), (b) contrastive CLIP features + small prefix-tuned decoder, and (c) a modern multimodal decoder (Gemma-style) perform on image captioning.
13	Classic Video GANs: Compare VGAN vs MoCoGAN for short action clips (decomposing motion & content) Implement and compare VideoGAN (VGAN) and MoCoGAN on generating short (1–3s) synthetic videos conditioned on action labels; emphasize decomposition into content and motion latent factors.
14	Create a project that showcases the interaction between text and images, evaluating the generated outcomes.

Minimum ten experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Prepared by Checked by HoD Principal

Program: Artificial Intelligence and Machine Learning	BTech	Semester: VII/VIII
Course: Capital Budgeting and Decision Strategies (DJS23ACPE713)		
Course :Capital Budgeting and Decision Strategies Laboratory (DJS23ALPE713)		

Prerequisite: - Foundation of Finance, Quantitative Finance, Financial Market and Risk Analysis, Pricing Models.

Course Objectives:

To develop advance statistics skills for financial data analysis.

Course Outcomes:

1. Apply simple and multiple regression models for data analysis, while addressing violations of classical regression assumptions.
2. Analyse advanced econometric techniques for time series analysis to extract insights from economic data.
3. Evaluate practical econometric skills to solve real-world economic problems and apply theoretical knowledge to data analysis. Analyse risk and return metrics, and evaluate asset pricing using CAPM, APT, and multi-factor models.

Capital Budgeting and Decision Strategies (DJS23ACPE713)		
Unit	**Description**	**Duration**
1.	**Foundations of Financial and Insurance Decision Frameworks:** Overview of financial decision-making in data-driven contexts; Role of insurance as a risk transfer mechanism; Expected utility theory, risk aversion; Financial statements (Balance Sheet, Income statements, Cash flow) and key ratios including profitability, liquidity, and solvency ratios; Introduction to forecasting and handling uncertainty in financial planning.	7
2.	**Investment Decisions and Capital Budgeting under Risk:** Concept of time value of money; Techniques for investment evaluation including Net Present Value (NPV), Annual Rate of Return (ARR), Internal Rate of Return (IRR), Profitability Index (PI), and Payback Period; Risk-adjusted discount rate; Sensitivity analysis and scenario analysis for risk evaluation; Integration of insurance-based risk mitigation.	8
3.	**Financing and Capital Structure Decisions:** Sources of finance including equity, debt, and hybrid instruments; Estimation of cost of capital—Cost of Equity (Ke), Cost of Debt (Kd), and Weighted Average Cost of Capital (WACC); Theories of capital structure including Net Income (NI) approach, Net Operating Income (NOI) approach, Modigliani–Miller (MM) proposition, and Traditional theory; Leverage impact and determination of optimal capital structure considering risk and return.	8

4.	**Dividend, Payout and Working Capital Management:** Theories of dividend policy including Walter, Gordon models; Dividend policy models such as stable dividend, residual dividend, and constant payout ratio models; Share buyback and payout optimization strategies; Overview of working capital management including management of cash, receivables, inventory, and payables; Concepts of liquidity risk and short-term financing options.	7
5.	**Financial and Insurance Risk Analytics:** Types of financial and insurable risks; Risk–return trade-offs and portfolio diversification principles; Basics of Value at Risk (VaR); Use of predictive analytics and data-driven forecasting techniques for financial and insurance risk assessment.	6
6.	**Simulation-Based Decision Modeling:** Monte Carlo simulation for investment and portfolio decision-making; Scenario-based decision analysis for uncertainty modeling; Conceptual introduction to Markov Chain Monte Carlo (MCMC) methods; Integrative case on combined financial and insurance risk simulation for portfolio or project decisions.	7
	TOTAL	**42**

Books Recommended:

Textbooks:

1. Brigham, E. F., & Houston, J. F. (2022). Fundamentals of Financial Management (13th Edition). Cengage Learning.

2. Benninga, S., & Mofkadi, T. (2022). Financial Modeling (5th Edition). MIT Press, Cambridge, MA.

3. Brealey, R. A., Myers, S. C., Allen, F., & Edmans, A. (2022). Principles of Corporate Finance (14th Edition). McGraw-Hill Education.

Reference Books:

1. Bodie, Z., Merton, R. C., & Cleeton, D. L. (2025). Principles of Finance. Cambridge University Press
2. Ross, S. A., Westerfield, R. W., & Jaffe, J. (2024). Corporate Finance (13th Edition). McGrawHill Education.
3. Bigel, K. S. (2023). Corporate Finance. Open Touro University Press (Open Access).

Web Links:

1. Aswath Damodaran – Applied Corporate Finance: A User's Manual (4th Edition, NYU Stern) https://pages.stern.nyu.edu/~adamodar/pdfiles/acf4E/acf4Ebook.pdf
2. Aswath Damodaran – Chapter 10: Investment Return Models (Applied Corporate Finance, 3rd Edition) https://pages.stern.nyu.edu/~adamodar/pdfiles/acf3E/ch10.pdf
3. Quoc Trung Tran – Dividend Policy: A Business Perspective (Emerald Publishing, 2023) https://bookstore.emerald.com/media/preview/9781837979882-23-2.pdf
4. Quantitative Financial Analytics and Risk Modelling (Routledge, 2021) https://api.pageplace.de/preview/DT0400.9781000320497_A40695981/preview9781000320497_A40695981.pdf

List of Suggested Experiments:

Capital Budgeting and Decision Strategies Laboratory (DJS23ALPE713)	
1	To analyze Balance Sheet, Income Statement, and Cash Flow Statement of a company and interpret financial performance. To calculate profitability, liquidity, and solvency ratios and evaluate financial health.
2	To estimate future cash flows using basic forecasting techniques under uncertain conditions.
3	To study investment choices under risk using expected utility theory and risk aversion concepts.
4	To compute present value, future value, and annuities for investment decisions. To evaluate investment projects using Net Present Value technique.
5	To determine project viability using Internal Rate of Return.
6	To assess projects using payback period and profitability index methods.
7	To evaluate best-case, worst-case, and expected-case investment scenarios.
8	To analyze the impact of changes in project variables on investment decisions.
9	To calculate cost of capital components for financing decisions. To determine WACC and analyze its role in capital budgeting.
10	To compare capital structure theories and identify optimal financing mix. To study operating and financial leverage and their effect on returns.
11	To evaluate dividend decisions using classical dividend valuation models.
12	To analyze cash, receivables, inventory, and payables for liquidity management.
13	To estimate financial risk exposure using Value at Risk.
14	To simulate uncertain investment outcomes using Monte Carlo methods.
15	To evaluate combined financial and insurance risks in a project or portfolio using simulation-based decision modeling.

Batch wise laboratory work of minimum ten experiments from the above-suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Prepared by Checked by HoD Principal

Program: Artificial Intelligence & Machine Learning	B.Tech.	Semester: VII/VIII
Course: AI in Robotics (DJS23ACPE714)		
Course : AI in Robotics Laboratory (DJS23ALPE714)		

Prerequisite:

Python programming and familiarity with scientific libraries (NumPy, Matplotlib, OpenCV, PyTorch/TensorFlow).
Understanding of Machine Learning and Deep Learning fundamentals: model training, CNNs, backpropagation, and evaluation metrics.
Basic knowledge of linear algebra, calculus, and probability as applied in AI and engineering systems.
Familiarity with Linux/Ubuntu environment, terminal usage, and package management (apt, pip, conda) is recommended.

Course Objectives: (At the end of course, student will be able to):

1. Introduce the integration of AI techniques with robotic systems covering perception, planning, control, and learning.
2. Explore classical and learning-based methods for robot motion planning, navigation, manipulation, and autonomous decision-making.
3. Provide hands-on experience with ROS2, simulation environments (Gazebo, Isaac Sim), and deep learning frameworks for robotics tasks.
4. Introduce foundation models, Vision-Language-Action (VLA) models, and LLM-driven task planning for next-generation intelligent robots.

Course Outcomes:

1. Understand core AI and robotic system architectures including the perception-planning-action-learning loop, robot kinematics, and sensor modalities.
2. Apply deep learning, imitation learning, and reinforcement learning algorithms to solve robot navigation, manipulation, and control problems.
3. Design, simulate, and evaluate robot systems using ROS2 or Gazebo, implementing AI perception and planning pipelines.
4. Apply foundation model-driven robotic systems using VLMs, VLAs, and LLM-based planners for open-vocabulary task execution.

AI in Robotics (DJS23ACPE714)		
Unit	**Description**	**Duration**
1.	**Foundations of AI in Robotics** Evolution of robotics: from rule-based automation to AI-driven autonomy, Robot taxonomy: industrial robots, mobile robots, humanoid robots, collaborative robots (cobots), drones, soft robots. Robot architecture: sensors (RGB, depth, LiDAR, IMU, force-torque, tactile), actuators, controllers, and end-effectors. Coordinate frames and transformations: homogeneous matrices, DH parameters, forward and inverse kinematics.	08
2.	**AI-Driven Robot Perception** Sensor modalities and their fusion: cameras, stereo depth, LiDAR point clouds, motion sensors. Classical computer vision for robotics: edge detection, feature extraction, optical flow, stereo matching. Deep learning for perception: CNN-based object detection, instance segmentation (Mask R-CNN, SAM — Segment Anything Model). 3D perception: point cloud	08

	processing Semantic scene understanding: panoptic segmentation, 3D scene graphs, Tools: OpenCV, Open3D, PyTorch3D, ROS2 perception packages	
3.	**Robot Motion Planning and AI-Augmented Control** Configuration space and robot motion basics, Configuration space (C-space), workspace analysis, degrees of freedom, and motion constraints. Classical path planning algorithms: Dijkstra, A*. AI-augmented planning: neural motion planners, learned heuristics for search, neural C-space representations. Introduction to PID control theory for robots. Learning-based control: neural network controllers, residual learning, adaptive control. Trajectory optimization, Robot manipulation and pick-and-place concepts. Manipulation planning.	07
4.	**Machine Learning and Deep Reinforcement Learning for Robotics** Supervised learning in robotics: regression for dynamics models, classification for perception tasks, anomaly detection. Imitation learning: behavioral cloning, Inverse Reinforcement Learning (IRL). Deep Reinforcement Learning (DRL) fundamentals: Markov Decision Processes (MDPs), Q-learning. Policy gradient methods: REINFORCE, Proximal Policy Optimization (PPO), Sim-to-real transfer: domain randomization, domain adaptation, reality gap analysis, AugGAN.	07
5.	**Foundation Models and Language-Driven Robotics** Foundation models for robotics: pre-training paradigms, zero-shot and few-shot generalization. Large Language Models (LLMs) for task planning: SayCan, Inner Monologue, Vision-Language Models (VLMs) for robot perception: CLIP, DINO v2. Vision-Language-Action (VLA) models: RT-1, RT-2. Diffusion models for robot learning: Diffusion Policy, 3D Diffusion Actor, BESO. Video Prediction models, UniSim. On-device deployment: model quantization, NVIDIA Jetson Orin, TensorRT, ROS2 integration with LLM APIs.	07
6.	**Human-Robot Interaction, Safety, and Emerging Applications** Human-Robot Interaction (HRI): shared autonomy, teleoperation, adjustable autonomy, intent prediction. Natural language interfaces for robots: voice command systems, dialogue-based task specification, multimodal HRI. Robot safety: functional safety standards (ISO 10218, ISO/TS 15066), collision detection, force-limiting, safety-rated monitored stops, safe RL. Ethical and social dimensions: job displacement, autonomous weapons (LAWS), accountability, robot rights, bias in robotic systems. Applications of AI robots in healthcare, industry, and autonomous systems. Future trends in humanoid and embodied AI robots.	04
	TOTAL	42

Books Recommended:

Textbooks:

1. Peter Corke, "Robotics, Vision and Control: Fundamental Algorithms in Python", 3rd ed., Springer, 2023.

2. Bruno Siciliano, Lorenzo Sciavicco et al., "Robotics: Modelling, Planning and Control", Springer, 2010.

3. Pieter Abbeel, Chelsea Finn, Sergey Levine, "Robot Learning — Foundations and Modern Approaches", UC Berkeley Lecture Notes, 2024.

Reference Books:

1. Lentin Joseph, "Robot Operating System (ROS) for Absolute Beginners", 2nd ed., Apress, 2022.

2. Michael Lanham, "Hands-On Deep Learning for Games", Packt, 2019. (Chapters on RL for robotics and simulation)

NPTEL Courses:

1. Introduction to Robotics https://onlinecourses.nptel.ac.in/noc22_me97/preview
2. Robot Mechanics and Control https://onlinecourses.nptel.ac.in/noc23_me65/preview
3. Deep Learning for Computer Vision https://onlinecourses.nptel.ac.in/noc22_cs92/preview
4. Reinforcement Learning https://onlinecourses.nptel.ac.in/noc24_cs58/preview
5. Introduction to Machine Learning https://onlinecourses.nptel.ac.in/noc23_cs86/preview
6. Modern Robotics: Mechanics, Planning, and Control Specialization https://www.coursera.org/specializations/modernrobotics
7. AI for Robotics https://www.coursera.org/learn/packt-artificial-intelligence-for-robotics

Suggested Experiments:

AI in Robotics Laboratory (DJS23ALPE714)	
1.	Set up Stable Baselines3 OR ROS2 (Humble/Jazzy) environment; create publisher-subscriber nodes and simulate a mobile robot in Stable Baselines3 OR Gazebo
2.	Implement forward and inverse kinematics for a 3-DOF robotic arm using Python and visualize with RViz2
3.	Perform real-time object detection using YOLOv8 on a camera feed and publish detected object poses over Stable Baselines3 /ROS2 topics
4.	Build a 2D occupancy grid map using LiDAR data in Stable Baselines3 /Gazebo and implement A* path planning for navigation
5.	Implement feature-based visual odometry using ORB keypoint matching and estimate robot trajectory from image sequences
6.	Train a CNN to classify objects in a robot workspace and integrate the model with a ROS2 perception node
7.	Implement a PID and MPC controller for a differential-drive robot to follow a given trajectory in simulation
8.	Train a Deep Q-Network (DQN) agent for obstacle-avoidance navigation in a 2D grid environment using OpenAI Gymnasium
9.	Train a PPO agent for robotic arm reaching and manipulation using Stable-Baselines3 and PyBullet
10.	Apply Behavioral Cloning (imitation learning) from human demonstrations to teach a simulated robot a pick-and-place task
11.	Implement domain randomization (textures, lighting, dynamics) in Isaac Sim and evaluate sim-to-real transfer performance
12.	Use CLIP and Grounded-SAM for language-guided object picking: specify target object in natural language, robot picks it
13.	Integrate an LLM (via OpenAI API) for high-level natural language task decomposition and execute steps on a simulated robot
14.	Mini project: Build a complete AI robotic pipeline — perception (YOLOv8), path planning (RRT*), LLM task planning, and manipulation execution in a Stable Baselines3 /ROS2-Gazebo environment

Batch-wise laboratory work of a minimum of ten experiments from the above suggested list or any other experiment based on the syllabus will be included, which would help the learner to apply the concept learnt.

Prepared by Checked by HoD Principal

Program: Artificial Intelligence & Machine Learning	B.Tech.	Semester: VII/VIII
Course: Quantum Computing (DJS23ACPE715)		
Course: Quantum Computing Laboratory (DJS23ALPE715)		

Prerequisite: Statistics for Data Science, and Machine Learning – I.

Course Objectives:

1. To equip students with foundational and practical knowledge of quantum computing, enabling them to understand quantum principles, analyze quantum algorithms, and apply quantum tools for solving computational problems.

Course Outcomes: On completion of the course, learner will be able to:

1. Explain the basic principles and mathematical foundations of quantum computing including qubits, superposition, and entanglement.
2. Analyze and construct quantum circuits for various quantum operations.
3. Demonstrate applications of key quantum algorithms such as Deutsch-Jozsa, Grover's, and Shor's.
4. Apply quantum cryptographic protocols like BB84 and analyze their security features.
5. Evaluate the effects of quantum error correction and noise models in computations.
6. Use industry-standard quantum programming platforms (e.g., Qiskit) to design, simulate, and test quantum algorithms.

Quantum Computing (DJS23ACPE715)		
Unit	**Description**	**Duration**
1	**Fundamentals of Quantum Computing and Qubits:** Introduction to Quantum Computing, Limitations of Classical Computing, Classical Bits vs Quantum Bits (Qubits), Basic Concepts of Quantum Mechanics, Quantum States, Superposition, Measurement, Representation of Qubits, Bloch Sphere Representation, Basic Mathematical Concepts, Complex Numbers, Vectors and Matrices.	07
2	**Qubits and Multi-Qubit Systems:** Quantum Gates and Their Functions, Single Qubit Gates- Pauli-X Gate, Pauli-Y Gate, Pauli-Z Gate, Hadamard Gate, Phase Gate, Rotation Gate Multi-Qubit Systems - Tensor Product Concept, Controlled Gates, CNOT Gate, SWAP Gate, Toffoli Gate. Quantum Entanglement, Bell States, No-Cloning Theorem.	07
3	Quantum Programming and Simulation: Introduction to Quantum Programming, Quantum Programming Platforms: Qiskit, Cirq, PennyLane. , Installing and Setting Up Quantum Environment, Writing Basic Quantum Programs, Quantum Circuit Simulation, Running Quantum Programs on Simulators, Visualization of Quantum States, Introduction to Quantum Cloud	06

	Platforms	
4	**Quantum Algorithms:** Introduction to Quantum Algorithms, Concept of Quantum Speedup Quantum Oracles and Query Model, Deutsch Algorithm, Deutsch–Jozsa Algorithm, Bernstein–Vazirani Algorithm, Simon's Algorithm Quantum Fourier Transform (QFT), Phase Estimation Algorithm, Shor's Algorithm (Quantum Factoring), Grover's Search Algorithm, Applications of Quantum Algorithms.	**08**
5	**Quantum Communication and Quantum Cryptography:** Introduction to Quantum Communication, Principles of Quantum Cryptography, Quantum Key Distribution (QKD), BB84 Protocol, Quantum Teleportation, Superdense Coding, Quantum Communication Networks, Security of Quantum Communication, Applications of Quantum Cryptography.	**07**
6	**Quantum Error Correction and Advanced Applications:** Introduction to Quantum Errors and Noise, Sources of Errors in Quantum Systems, Quantum Error Correction Concepts, Bit-Flip and Phase-Flip Errors, Shor Code and Steane Code, Fault-Tolerant Quantum Computing, NISQ (Noisy Intermediate-Scale Quantum) Computing, Variational Quantum Algorithms (VQE, QAOA), Introduction to Quantum Machine Learning Applications of Quantum Computing: Optimization, Artificial Intelligence, Chemistry, Finance	**07**
	TOTAL	**42**

Books Recommended:

Text Books:

1. Michael A. Nielsen and Isaac L. Chuang, Quantum Computation and Quantum Information, 10th Anniversary Edition, Cambridge University Press, 2010.

2. Phillip Kaye, Raymond Laflamme, and Michele Mosca, An Introduction to Quantum Computing, 1st Edition, Oxford University Press, 2007.

3. Eleanor G. Rieffel and Wolfgang H. Polak, Quantum Computing: A Gentle Introduction, 1st Edition, MIT Press, 2011.

Reference Books:

1. Noson S. Yanofsky and Mirco A. Mannucci, Quantum Computing for Computer Scientists, 1st Edition, Cambridge University Press, 2008.

2. Robert S. Sutor, Dancing with Qubits: How Quantum Computing Works and How It Can Change the World, 1st Edition, Packt Publishing, 2019.

3. Ivan B. Djordjevic, Quantum Information Processing, Quantum Computing and Quantum Error Correction: An Engineering Approach, 2nd Edition, Elsevier (Academic Press), 2021.

Web Links:

1. Web resources:
IBM Quantum Learning: https://quantum.cloud.ibm.com/learning

MIT xPRO Quantum Computing Fundamentals: https://learn-xpro.mit.edu/quantum-computing

2. Online Courses: NPTEL / Swayam :
   1. Introduction to Quantum Computing course: https://onlinecourses.nptel.ac.in/noc25_cs95/preview .
   2. Coursera Quantum Computing course: https://www.coursera.org/learn/quantum-computing

List of Experiments:

Quantum Computing Laboratory (DJS23ALPE715)	
Sr. No.	**Title of the Experiment**
1	Prepare single and multi-qubit quantum states (superposition, entangled states, Bell states)
2	Visualizing qubit and multi-qubit states on the Bloch sphere Observing the effects of various quantum gates on qubits
3	Implementation and experimental testing of basic gates (Pauli-X, Y, Z, Hadamard, Phase, CNOT, Toffoli, SWAP)
4	Deutsch and Deutsch-Jozsa algorithms: distinguishing between constant and balanced functions
5	Grover's search algorithm: amplitude amplification and unstructured search
6	Simon's algorithm: finding a secret string with exponential speedup
7	Shor's algorithm (simulation): factoring small integers
8	Preparation, measurement, and verification of Bell states and entanglement
9	Assignments on BB84 quantum key distribution protocol Analysis of quantum eavesdropping and protocol robustness
10	Implementation of three-qubit bit-flip and phase-flip error correction codes
11	Implement and benchmark quantum key distribution protocols (BB84, E91) on quantum networks to advance secure access for underserved populations. (SDG-9)
12	Implement and compare quantum support vector machines (QSVM) or quantum k-nearest neighbors (QkNN)
13	Demonstrate quantum teleportation across chained entangled qubit pairs or quantum repeater simulation
14	Compare performance of different quantum compiler tools (Qiskit, Cirq, PennyLane) on non-trivial circuits.

Batch wise laboratory work of minimum ten experiments from the above-suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Prepared by Checked by HoD Principal

Program: Artificial Intelligence & Machine Learning	B.Tech.	Semester :VII/VIII
Course: High Performance Computing (DJS23ACPE716)		
Course: High Performance Computing Laboratory(DJS23ALPE716)		

Prerequisite: System Fundamentals, C/C++ programming, Data Structures.

Course Objectives (At the end of course, student will be able to):

1. To learn concepts of parallel processing as it pertains to high-performance computing.
2. To design, develop and analyze parallel programs on high performance computing resources using parallel programming paradigm.

Course Outcomes:

1. Apply GPU architecture concepts and the CUDA programming model to address parallel programming challenges in heterogeneous systems.
2. Implement CUDA C programs to solve data parallelism problems and use CUDA memory models effectively.
3. Analyze GPU performance metrics, including memory coalescing, thread scheduling, and control divergence impacts.
4. Assess different GPU computing frameworks and libraries (cuBLAS, cuDNN) based on application requirements and efficiency.

Detailed Syllabus: High Performance Computing (DJS23ACPE716)		
Unit	**Description**	**Duration**
1.	**Introduction to Parallel Processing:** Sequential vs parallel computing, Need and applications of parallel computing, parallel computing architectures – shared memory and distributed memory, Flynn's taxonomy, Limitations of single core CPU, Multi-core CPU architecture. Performance Metrics in parallel computing – Speedup, Efficiency, scalability, Amdahl's law.	08
2.	**CUDA Parallelism Model:** **Introduction**: Architecture of A Modern GPU, Challenges in Parallel Programming, Parallel Programming Languages and Models. **Data Parallelism:** Data Parallelism, CUDA C Program Structure, A Vector Addition Kernel, Device Global Memory and Data Transfer, Kernel Functions and Threading, Kernel Launch. **Scalable Parallel Execution:** CUDA Thread Organization, Mapping Threads To Multidimensional Data, Synchronization And Transparent Scalability, Resource Assignment, Thread Scheduling And Latency Tolerance.	08
3.	**Performance Considerations On GPUs:** **Memory Access Performance:** Global Memory Bandwidth, Memory Coalescing in CUDA, Channels and Banks in Dram Systems, Techniques For Reducing Memory Transfers Between CPU And GPU. **Memory And Data Locality:** CUDA Memories, Tiled Parallel	08

	Algorithms, Tiled Matrix Multiplication, Tiled Matrix Multiplication Kernel, Boundary Checks, Memory As A Limiting Factor To Parallelism. **Thread Execution Efficiency:** Warps And SIMD Hardware, Dynamic Partitioning Of Resources, Performance Impact Of Control Divergence.	
4.	**Parallel Computation Patterns:** **Parallel Patterns: Convolution:** 1d Parallel Convolution—A Basic Algorithm, Constant Memory and Caching, Tiled 1d Convolution with Halo Cells, A Simpler Tiled 1d Convolution—General Caching. **Parallel Computation Patterns (Histogram):** Atomic Operations in CUDA, Atomic Operation Performance-Block versus Interleaved Partitioning, Latency versus Throughput of Atomic Operations, Atomic Operation in Cache Memory, Privatization Technique For Improved Throughput.	08
5.	**Performance Optimization on GPUs:** **Efficient Host-Device Data Transfer:** Pinned Host Memory, Task Parallelism in CUDA, Overlapping Data Transfer with Computation, CUDA Unified Memory. **Techniques for optimizing GPU performance:** warp divergence, loop unrolling, vectorization, Memory bandwidth optimization techniques, Advanced GPU programming concepts (shared memory atomics, warp shuffling).	04
6.	**Advanced Topics & AI Integration:** Introduction to GPU accelerated libraries (cuBLAS, cuDNN, cuGraph). Machine Learning Implementation: Reduction of Convolutional Layers to Matrix Multiplication, and a basic CUDA implementation of Forward Propagation. Integration with TensorFlow/PyTorch. Introduction to GPU clusters and distributed GPU computing.	06
	TOTAL	42

Books Recommended:

Textbooks:

Core HPC & Cloud Computing

1. Edson Borin et al., "High Performance Computing in Clouds: Moving HPC Applications to a Scalable and Cost-Effective Environment", Springer 2023.
2. Robert Robey & Yuliana Zamora, "Parallel and High Performance Computing", Manning Publications 2021.
3. Georg Hager & Gerhard Wellein, "Introduction to High Performance Computing for Scientists and Engineers", CRC Press 2019.
4. Charles Severance & Kevin Dowd, "High Performance Computing", O'Reilly Media, 2nd Edition.
5. Christian Vecchiola et al., "High-Performance Cloud Computing: A View of Scientific Applications", IEEE 2009.

GPU & CUDA Programming

1. David B. Kirk & Wen-mei W. Hwu, "Programming Massively Parallel Processors: A Hands-on Approach", Morgan Kaufmann, 4th Edition 2022.
2. Richard Ansorge, "Programming in Parallel with CUDA", Cambridge University Press 2022.
3. Jason Sanders & Edward Kandrot, "CUDA by Example: An Introduction to General-Purpose GPU Programming", Addison-Wesley.

4. Duane Storti & Mete Yurtoglu, "CUDA for Engineers: An Introduction to High-Performance Parallel Computing", Addison-Wesley 2016.

AI & Specialized Applications

1. Alexander Heifetz, "High Performance Computing for Drug Discovery and Biomedicine", Springer Nature 2023.
2. Brian Tuomanen, "Hands-On GPU Programming with Python and CUDA", Packt Publishing 2018.
3. Brian Tuomanen, "Deep Learning with CUDA", O'Reilly Media.

Reference Books:

1. David B. Kirk and Wen-mei W. Hwu, "Programming Massively Parallel Processors: A Hands-on Approach", Morgan Kaufmann, 4th Edition, 2022.
2. Jason Sanders and Edward Kandrot, "CUDA by Example: An Introduction to General-Purpose GPU Programming", Addison-Wesley 2010.
3. Georg Hager and Gerhard Wellein, "Introduction to High Performance Computing for Scientists and Engineers", CRC Press 2010.
4. Douglas Eadline, "High Performance Computing for Dummies", Wiley Publishing, Inc. 2009.
5. Catherine Régis, Jean-Louis Denis, Maria Luciana Axente, and Atsuo Kishimoto, "Human-Centered AI: A Multidisciplinary Perspective for Policy-Makers, Auditors, and Users", CRC Press 2024.

NPTEL and Coursera Courses:

Course Code	Course Name	Institute / Instructor	Link
noc20_me65	High Performance Computing	Prof. Somnath Roy / IIT Kharagpur	nptel.ac.in/courses/112105293
noc21_cs82	Multi-Core Computer Architecture - Storage and Interconnects	Prof. John Jose / IIT Guwahati	Preview: Multi-Core Computer Architecture – Storage and Interconnects | SWAYAM
noc22_cs110	Multi-Core Computer Architecture - Storage and Interconnects	Prof. John Jose / IIT Guwahati	Preview: Multi-Core Computer Architecture – Storage and Interconnects | SWAYAM
Coursera	Introduction to High Performance and Parallel Computing	University of Colorado Boulder	Introduction to High-Performance and Parallel Computing | Coursera

Suggested Experiments:

High Performance Computing Laboratory (DJS23ALPE716)	
1.	Set up the CUDA environment and write a basic "Hello World" kernel.
2.	Implement Vector Addition to master thread management and GPU memory allocation.
3.	Develop an optimized Tiled Matrix Multiplication kernel using Shared Memory.
4.	Apply CUDA for Image Processing (Blurring/Edge Detection).
5.	Implement Parallel Reduction (Sum/Min/Max) to grasp efficient collective operations.
6.	Explore Parallel Sorting Algorithms and compare GPU vs. CPU performance.
7.	Employ CUDA for Monte Carlo Simulations to estimate mathematical constants.
8.	Implement Concurrent Data Structures using locks and atomic operations.
9.	Optimize reduction steps using Warp Shuffling and Privatization techniques.
10.	Implement a Convolutional Layer Forward Propagation kernel using CUDA/cuDNN.
11.	Perform Log Analysis-Based Resource and Execution Time Improvement.
12.	Analyze Memory Coalescing and Control Divergence using GPU Profiling tools.
13.	Implement 1D Parallel Convolution using Constant Memory and Caching.

14.	Accelerate Graph Analytics (PageRank/BFS) using the cuGraph library.
15.	Implement Task Parallelism and Data Overlap using CUDA Streams and Unified Memory.

Batch wise laboratory work of minimum ten experiments from the above suggested list or any other experiment based on syllabus will be include, which would help the learner to apply the concept learnt.

SVKM's DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING Estd. 1994 CREATING ENGINEERS

Prepared by | Checked by | HoD | Principal

Program: Artificial Intelligence & Machine Learning	B.Tech.	Semester: VII/VIII
Course: Cloud Computing (DJS23ACMD701)		
Course: Cloud Computing Laboratory (DJS23ALMD701)		

Prerequisite: Computer Networks, Operating Systems & Linux Basics, Database Management Systems

Course Objectives:

1. To familiarize with modern cloud computing deployment topologies and containerization frameworks.
2. To gain hands-on experience in programmatically provisioning cloud resources using Infrastructure as Code (IaC).
3. To introduce the best practices for securing cloud services, implementing FinOps cost controls, and managing access rules.
4. To introduce modern serverless application design patterns and distributed observability techniques.

Course Outcomes (At the end of course, student will be able to):

1. Formulate resilient cloud architecture patterns utilizing public, private, and hybrid cloud models.
2. Programmatically provision, version, and manage multi-tier cloud networks using declarative IaC tools.
3. Apply modern Identity and Access Management (IAM), FinOps, and DevSecOps security guardrails.
4. Architect, containerize, scale, and monitor distributed microservices and serverless workloads.

Cloud Computing (DJS23ACMD701)		
Unit	**Description**	**Duration**
1.	**Cloud Architecture Foundations & Evolution:** Overview of cloud computing evolution from distributed and grid paradigms to Cloud-Native architectures, Fundamentals of cloud computing ecosystem and core characteristics, Cloud service models: IaaS, PaaS, SaaS, Core cloud services: compute, storage, object/block/file storage, databases, load balancers, DNS, queues, and autoscaling, Hypervisors vs. Container Virtualization, Cloud Global Infrastructure (Regions, Availability Zones, and Edge Locations)	**04**
2.	**Infrastructure as Code (IaC) & Cloud Provisioning:** Declarative vs. Imperative cloud provisioning, Idempotency and configuration drift, Infrastructure state management, remote backends, and concurrency state locks, Modular infrastructure architecture, Cloud networking layouts (VPCs, Public/Private subnets, Route Tables, and NAT Gateways).	**05**
3.	**Containerization & Cloud-Native Orchestration:** Monolithic to Microservices application migration patterns, Open Container Initiative (OCI) standards, Container networking interfaces (CNI) and storage volumes (CSI), Declarative container orchestration systems, Pod lifecycle and resource scheduling, Cloud deployment strategies (Rolling updates, Blue-	**06**

	Green, and Canary deployments).	
4.	**Serverless Computing & Event-Driven Architectures:** Overview of serverless computing and execution patterns, Micro-VM architectures and execution lifecycle, Optimizing cold starts vs. warm starts, Managed cloud notification services and event queues, Decoupled event producers, routers, and event consumers.	**04**
5.	**Cloud Security, FinOps & Site Reliability:** Securing the Cloud, the shared responsibility model, Managing the cloud security service boundary, Identity and Access Management (IAM), Least Privilege Principle and Role-Based Access Control (RBAC), Automated DevSecOps compliance and vulnerability scanning, Introduction to FinOps frameworks (Cloud cost auditing, right-sizing resources, and cost prediction).	**05**
6.	**Edge Computing, CDN & Hybrid Cloud Deployment:** Evolution from centralized cloud computing to distributed, Edge and Fog topologies, Cloud-to-Edge architectural frameworks, Content Delivery Networks (CDNs), edge caching mechanics, and edge compute functions, Hybrid cloud infrastructure patterns, Introduction to Edge AI/ML deployment considerations (hardware optimization, latency constraints, and dynamic data synchronization).	**04**
	TOTAL	**28**

Books Recommended:

Textbooks:

1. Yevgeniy Brikman, "Terraform: Up & Running: Writing Infrastructure as Code", 3rd Edition, O'Reilly Media, 2023.
2. Brendan Burns, Joe Beda, and Kelsey Hightower, "Kubernetes: Up and Running", 3rd Edition, O'Reilly Media, 2022.
3. Arshdeep Bahga and Vijay Madisetti, "Cloud Computing: A Hands-On Approach", 1st Edition, VPT/Universities Press, 2014.

Reference Books:

1. Justin Garrison and Kris Nova, "Cloud Native Infrastructure: Patterns for Scalable Infrastructure and Applications", 1st Edition, O'Reilly Media, 2017.
2. Thomas Erl, Ricardo Puttini, and Zaigham Mahmood, "Cloud Computing: Concepts, Technology & Architecture", 1st Edition, Pearson Education, 2013.
3. Betsy Beyer, Chris Jones, Jennifer Petoff, and Niall Richard Murphy, "Site Reliability Engineering: How Google Runs Production Systems", 1st Edition, O'Reilly Media, 2016.

NPTEL and Coursera Courses:

Course Code	Course Name	Institute / Instructor	Link
noc23_cs42	Cloud Computing	Prof. Soumya Kanti Ghosh (IIT Kharagpur)	https://onlinecourses.nptel.ac.in/noc23_cs42/preview
noc24_cs115	Google Cloud Computing Foundations	Joint Course by NPTEL and Google Cloud Team	https://nptel.ac.in/courses/106105223
noc22_cs90	Multi-Core Computer Architecture: Storage and Interconnects	By Prof. John Jose (IIT Guwahati)	https://nptel.ac.in/courses/106103183
Coursera	Introduction to Cloud	IBM	https://www.coursera.org/lea

Suggested Experiments:

Cloud Computing Laboratory (DJS23ALMD701)	
1.	Local Cloud Simulation Environment: Configure a local, emulated AWS environment using LocalStack and MinIO to interact with mock cloud storage via the AWS-CLI.
2.	Declarative Cloud Network Design: Write and validate a declarative Terraform script to provision a secure VPC with segregated Public/Private subnets.
3.	Stateful Infrastructure Controls: Implement automated state locking mechanisms using local backend databases to safely manage team-based infrastructure modifications.
4.	Production-Ready Application Packaging: Develop optimized, multi-stage Docker configurations to package web microservices securely under non-root permissions.
5.	Microservices Deployment Orchestration: Write declarative Kubernetes manifests to deploy multi-tier microservice workloads featuring localized configuration variables.
6.	Cloud Auto-Scaling and Self-Healing Automation: Implement a Kubernetes Horizontal Pod Autoscaler (HPA) and run synthetic stress testing to watch automated scheduling response.
7.	Serverless Event Hooks: Code and run an asynchronous background compute function triggered autonomously by an object storage file upload event.
8.	DevSecOps Pipeline Integration: Build automated configuration linters using Trivy and Checkov to catch security holes and hidden hardcoded tokens before deployment.
9.	FinOps Financial Forecasting: Configure Infracost scripts within a local repository to output a estimated monetary variance report on infrastructure modifications.
10.	Edge CDN Caching & Header Optimization. Deploy an application backend and configure a Content Delivery Network (CDN) layer (using an emulator or Cloudflare/CloudFront free tier).
11.	Containerized Edge AI Inference Deployment: Package a lightweight, pre-trained AI/ML inference model (like a mobile computer vision or NLP model) into a highly optimized Docker container. Restrict the container's CPU/RAM programmatically to simulate a low-power edge gateway/device and test its latency limits.
12.	Live Multi-Tier Cloud Deployment: Launch a fully secure and functional runtime application stack on cloud free tiers.
13.	Continuous Integration & Automation Pipelines (CI/CD): Construct a GitHub Actions pipeline that triggers formatting checks, unit assessments, and automated updates to cloud environments.
14.	Resilience Verification via Chaos Testing: Inject artificial performance degradation and instance terminations to evaluate real-time self-healing capabilities.

Batch-wise laboratory work of a minimum of ten experiments from the above suggested list or any other experiment based on the syllabus will be included, which would help the learner to apply the concept learnt.

Continuous Assessment:

20 Marks- Journal completion
5 Marks- Attendance
25 Marks- Mini Project

Prepared by Checked by HoD Vice Principal Principal

Program: All programs	Final Year B. Tech.	Semester: VII/VIII
Course: Research Ecosystem (DJS23ITELX12)		

Pre-requisites:

1. Basic understanding of Engineering concepts and problem-solving techniques.
2. Fundamental knowledge of project-based learning, data analysis, and domain-specific tools.
3. Basic exposure to project development, technical report writing, and emerging technologies.

Objectives:

1. To understand and identify complex engineering, societal, and industrial problems suitable for research through structured analysis of literature, patents, and real-world needs.
2. To learn and apply advanced research methodologies—including experimental, simulation-based, and data-driven approaches—for designing and validating engineering solutions.
3. To gain proficiency in selecting and using appropriate tools, software, and platforms (AI/ML, simulation, CAD, IoT, etc.) for experiment design, data acquisition, and performance evaluation.
4. To transform research insights into innovative prototypes, products, or startup concepts by engaging with innovation ecosystems, TRLs, and funding mechanisms.
5. To analyze socio-technical systems, ethics, and sustainability implications of engineering solutions, ensuring responsible and impactful outcomes.
6. To integrate systems thinking, multidisciplinary perspectives, policy awareness, future technology trends, and technical documentation skills to develop comprehensive research proposals and outputs (papers, patents, reports).

Outcomes: On completion of the course, the learner will be able to:

1. Analyze complex engineering and societal problems to evaluate literature and patents and formulate clear research gaps.
2. Design and apply advanced research methods and tools to acquire data and evaluate solution performance.
3. Transform research insights into prototypes or startup concepts using innovation frameworks and Technology Readiness Levels.
4. Evaluate ethical, socio-technical, and sustainability issues and create responsible engineering solutions.
5. Synthesize multidisciplinary knowledge to create research proposals and scalable, future-ready solutions.

Research Ecosystem (DJS23ITELX12)		
Unit	**Description**	**Duration**
1	**Research Gap Identification & Problem Formulation** • Identifying high-impact problems from: ▪ Industry and Society ▪ Emerging technologies (AI, IoT, Blockchain, EV, etc.) • Research gap identification techniques: ▪ Systematic Literature Review (SLR) ▪ Patent analysis	5

	• Defining objectives, scope, constraints, and feasibility of the identified problem • Writing problem statement for: ▪ Research paper ▪ Product/Startup idea **Activities:** • Convert Industry/ Societal problem into research problem. • Analyze research articles and extract research gaps.	
2	**Experimental Design, Tools & Validation Techniques** • Advanced research methodologies: ▪ Experimental, simulation, data-driven, design thinking • Tool selection: ▪ Computational tools and software, simulation and analysis platforms, system prototyping and testing platforms • Data acquisition: ▪ Data collection interfaces, Principles of data cleaning and organization (primary / secondary data) • Validation techniques: ▪ Accuracy and performance metrics ▪ Benchmarking with existing solutions • Result interpretation & visualization **Activities:** • Selection of Research Design for final year project. • Apply evaluation metrics (e.g., ML model accuracy, system efficiency).	6
3	**Research to Innovation, Product & Startup Ecosystem** • Research → Prototype → Product → Startup lifecycle • Intellectual Property Rights (IPR), patents, copyrights • Technology readiness levels (TRL) • Case studies: ▪ AI startups, EV ecosystem, Smart Cities • Industry-academia collaboration models • Funding opportunities (grants, incubation, government schemes - Anusandhan National Research Foundation (ANRF)) **Activities:** • Convert project into startup idea or product pitch. • Analyze failure of a technology/product.	5
4	**Socio-Technical Systems, Ethics & Sustainability** • Socio-technical systems: Interaction of people, technology and environment • Ethical issues: ▪ Plagiarism, AI bias, data privacy • Sustainability in Engineering: ▪ Green practices, resource optimization and lifecycle considerations • Impact assessment: ▪ Social, economic, environmental • Policies and governance (data protection laws, standards and compliance requirements) **Activities:** • Case study: Facial recognition ethics, smart surveillance. • Impact analysis of student's own project.	6
5	**Integrated Research, Proposals and Future Engineering Challenges** • Systems thinking and holistic analysis • Multidisciplinary problem solving: ▪ Healthcare tech., climate tech., smart mobility	6

	• Writing research proposals: ▪ Problem → Methodology → Expected outcomes • Future trends: ▪ AI ethics, Quantum computing, Industry 5.0 / 6.0 • Scaling solutions from lab to society **Activities:** • *Capstone Task:* ▪ Identify real-world problem. ▪ Apply research methodology. ▪ Propose: o Technical solution o Societal impact o Possible commercialization	
	Total	**28**

Term Work Guidelines:

The Term Work (TW) shall consist of continuous assessment based on tutorials activities and a final group presentation.

a) **Part A (25 Marks):**
Each student shall complete a minimum of five tutorials based on the syllabus topics during the semester. The tutorials may include activities such as research gap identification, literature survey, case study analysis, tool-based experimentation, impact analysis, technical report writing, and innovation-oriented problem solving.

b) **Part B (25 Marks):**
Students shall work in groups and select a project/research topic based on emerging engineering, industrial, or societal challenges. The group presentation shall include:
i. Identification of a real-world problem
ii. Application of appropriate research methodology
iii. Proposed technical solution
iv. Analysis of societal impact
v. Possible commercialization/startup opportunities

Books Recommended:

Reference Books:

- Kothari, C. R., Research Methodology: Methods and Techniques, New Age International Publishers, New Delhi, 2023.
- Booth, W. C., Colomb, G. G., Williams, J. M., Bizup, J., & FitzGerald, W. T., The Craft of Research, University of Chicago Press, Chicago, 2024.
- Montgomery, D. C., Design and Analysis of Experiments, Wiley, New York, 2020.
- Singh, Pooja, Katiyar, Aman, Rawat, Shalini, Dhuri, Archana, Research Methodology and IPR, RK Publications, India, 2025.
- Blok, Vincent (Ed.), Putting Responsible Research and Innovation into Practice: A Multi-Stakeholder Approach, Springer Cham, Switzerland, 2022.

Web References:

- Research Methodology: Quantitative Methods and Computer Application (https://onlinecourses.swayam2.ac.in/e-learning/preview/ini26_ma02)

- Research Methodology and IPR
  (https://onlinecourses.swayam2.ac.in/e-learning/preview/ntr26_ed28)
- Research and Publication Ethics
  (https://onlinecourses.swayam2.ac.in/e-learning/preview/nou26_ge46)
- Research Ethics and Plagiarism
  (https://onlinecourses.swayam2.ac.in/e-learning/preview/nou26_ge26)
- Quantitative and Mixed Methods Research for Management
  (https://onlinecourses.swayam2.ac.in/e-learning/preview/imb26_mg21)
- Doing Qualitative Research
  (https://onlinecourses.swayam2.ac.in/e-learning/preview/nou26_ge61)

Prepared by Checked by Head of the Department Principal

Program: B. Tech. (Common to all Programmes)	B. Tech.	Semester: VII/VIII
Course: Major Project	Course Code: DJS23IPELX13	

Objectives

The primary purpose of the Major Project is to enable students to apply the knowledge and skills acquired across the programme to conceive, design, implement, and evaluate a comprehensive solution to a real-world problem. The project strengthens technical competence, communication, and teamwork by integrating design, development, technical writing, presentation, and collaborative work. It emphasises professional engineering practice and addresses a range of non-technical considerations such as economic factors, safety, reliability, ethics, sustainability, and environmental and societal impact.

Course Outcomes (CO)

On successful completion of the Major Project, the student will be able to:

CO1: Apply technical knowledge gained from previous courses to identify real-life, industry, or societal problems and design appropriate solutions.

CO2: Demonstrate the technical skills required to design, develop, build, and test engineering systems, products, software, or hardware using modern engineering and IT tools.

CO3: Apply project-management skills — scheduling work, procuring resources/components, budgeting, documenting technical and non-technical details, and working within defined deadlines as an effective team.

CO4: Develop and demonstrate the ability to test, validate, debug, and troubleshoot engineering systems (including software, hardware, and process components) and analyse the results.

CO5: Create technical reports and research papers in the prescribed format and effectively communicate and present the work to evaluators.

Scope and Nature of the Project Work

Final-year students have already undertaken project assignments in their pre-final year through Innovative Product Design / Mini Project work. In the final year, a group of a maximum of four students completes a comprehensive project based on the courses studied. The project work may be internally assigned or externally assigned by research institutes, industry, or similar organisations. Each group is assigned one faculty member as supervisor. The final-year project may be an extension of the Mini Project carried out in the pre-final year.

The main intention of the project work is to enable students to apply the knowledge and skills learnt across the programme to solve or implement a well-defined practical problem. The work may go beyond the scope of the curriculum or build upon courses studied, but the thrust should be on:

- Learning additional skills relevant to the chosen domain;
- Developing the ability to define, design, analyse, and implement a problem and carry it through to completion with proper planning;
- Learning behavioural science and professional ethics by working effectively in a group;
- Selecting a project area aligned with the student's intended higher education, employment, or self-employment goals;
- Choosing a topic that differs from, or meaningfully advances, the Mini Project topic;

- Using the opportunity to learn modern computational techniques, tools, and model development through an appropriate choice of project.

Continuous Assessment and Term Work

The department maintains a proper record of the progress of each project throughout the semester. At the end of the semester, the work is assessed for the award of Term Work (TW) marks by an approved internal faculty member appointed by the Head of the Institute, on the basis of:

- Scope and objectives of the project work;
- Extent and quality of the literature survey;
- Progress of the work (continuous assessment across reviews);
- Design, implementation, and analysis of the project work;
- Results, conclusions, and future scope;
- Report prepared in the prescribed University / institutional format.

Oral / Practical Examination: An approved external examiner together with an internal examiner appointed by the Head of the Institute assesses the project during the oral examination. The oral examination consists of a presentation by the group along with a demonstration of the work done. Each student is assessed individually for their contribution, understanding, and the knowledge gained.

Project Guide / Topic Allocation Policy

The project guide and topic-allocation policy is explained to students during the Orientation held in Semester VI. Each faculty member is allocated a minimum of one and a maximum of four project groups. As per the timeline in the academic calendar, the list of project topics and areas offered by each faculty member — three topics per faculty member — is displayed on the notice board. Allocation is carried out over successive rounds, and in each round students submit their preferences against the available topics and guides.

If a topic is preferred by only one group, that group is allocated to the corresponding guide and topic. If a topic is preferred by more than one group, the concerned faculty member reviews the applications and selects one group; the remaining groups carry forward to the next round. After each round, the list of groups and guides allotted so far is displayed, along with the topics and guides that remain available. Unallotted groups then submit fresh preferences from the available list. This process continues over successive rounds until every group has been allotted a guide, ensuring a uniform distribution of project groups among the faculty members.

Assessment Scheme

The Major Project carries a total of 100 marks — **Term Work (50)** assessed internally through continuous evaluation, and **Oral / Practical Examination (50)** assessed jointly by internal and external examiners. Each criterion is scored against three performance levels — Excellent, Satisfactory, and Below Expectation — as detailed below.

Rubric for Term Work (50 Marks)

Assessment Criteria (mapped CO)	Max	Excellent	Satisfactory	Below Exp.
Problem Definition, Objectives & Literature Survey (CO1)	**10**	8–10	5–7	0–4

Assessment Criteria (mapped CO)	Max	Excellent	Satisfactory	Below Exp.
Design, Methodology & Implementation Progress – Continuous Assessment (CO2)	10	8–10	5–7	0–4
Project Management & Teamwork (CO3)	10	8–10	5–7	0–4
Report / Thesis Writing – Written Communication (CO5)	10	8–10	5–7	0–4
Research Paper Writing / IPR – Patent or Copyright (CO5)	10	8–10	5–7	0–4
TOTAL	50			

Rubric for Oral / Practical / Presentation / Demonstration (50 Marks)

Assessment Criteria (mapped CO)	Max	Excellent	Satisfactory	Below Exp.
Concept & Problem Definition (CO1)	5	4–5	3	0–2
Design, Methodology & Implementation (CO2)	15	12–15	7–11	0–6
Project Management Skills (CO3)	5	4–5	3	0–2
Demonstration, Testing & Result Analysis (CO4)	15	12–15	7–11	0–6
Presentation & Oral Communication (CO5)	10	8–10	5–7	0–4
TOTAL	50			

Descriptive Assessment Rubric (Performance-Level Descriptors)

The following descriptors define each performance level and are applied consistently across all programmes. Examiners map a student's performance to the most appropriate level for each criterion before awarding the corresponding marks from the tables above.

Criteria	Excellent	Satisfactory	Below Expectation
Concept & Problem Definition	All major and minor objectives are identified and correctly prioritised. The problem statement is precise, relevant and well-scoped. All relevant information is gathered from valid sources and the proposed solution is well justified.	Major objectives identified but one or two minor ones missing or priorities not fully set. Problem statement mostly clear. Sufficient information from largely valid sources; solution mostly justified.	Several major objectives not identified. Problem statement vague or too broad. Insufficient information and/or invalid sources; proposed solution not justified.
Literature Survey	Comprehensive, current and well-organised survey of journals, standards, patents and existing products/systems. Research gap / novelty clearly identified and the	Adequate survey covering most key sources. Gap / novelty identified but only partially justified.	Sparse or outdated survey. Little or no identification of the gap; chosen approach not justified.

Criteria	Excellent	Satisfactory	Below Expectation
	chosen approach is well justified.		
Design, Methodology & Implementation	Design is based on stated criteria, sound analysis and realistic constraints. Methodology is appropriate and well justified. Implementation is complete and robust and makes effective use of modern engineering / software tools.	Design is reasonable and methodology mostly appropriate. Implementation is largely complete; deeper analysis of alternatives or constraints could improve the outcome.	Only one solution considered or alternatives ignored. Weak methodology and incomplete implementation. Many constraints and criteria ignored.
Project Management & Teamwork	Responsibilities delegated fairly and every member contributes valuably. All members attend meetings and meet deliverable deadlines. Schedule, resources and budget are well managed.	Minor inequities in delegation; some members contribute more than others but all meet their responsibilities. Meetings mostly attended and deadlines mostly met.	Major inequities in delegation, with free-riders or dominating members. Meetings and deadlines frequently missed; poor planning of schedule and resources.
Demonstration, Testing & Result Analysis	Develops and follows logical testing / validation / experimental procedures. Selects appropriate tools, equipment, platforms or datasets and operates them confidently. Aware of errors / limitations, accounts for them, and interprets and relates the results to the objectives and underlying theory.	Procedures mostly followed with occasional oversight. Needs some guidance in selecting tools / equipment and is tentative in operating them. Aware of errors but accounts for them only minimally.	Does not follow a proper procedure. Cannot select appropriate tools / equipment or operates them incorrectly, needing frequent supervision. Makes no attempt to relate results to objectives.
Report / Thesis Writing (Written Communication)	Report is well organised and clearly written; the logic is easy to follow. Wording is precise; figures, tables and analyses enhance understanding. Grammatically correct, free of spelling errors, properly cited and free of plagiarism.	Report is mostly organised and clear; a few sections are harder to follow. Wording is good with minor exceptions; figures are consistent. Few grammatical / spelling errors that do not hinder the reader; some sources not fully cited.	Report lacks organisation and the logic is hard to follow. Figures absent or inconsistent. Grammatical and spelling errors hinder reading. Evidence of plagiarism or inadequate citation.
Research Paper Writing / IPR (Patent or Copyright)	A research / technical paper is drafted to publication standard in the required format, or a patent / copyright application is prepared and filed. Novelty and	A paper draft or IPR document is prepared and largely follows the required format. Novelty and contribution are stated but only partially developed, with minor	No paper / IPR document, or a weak draft that does not follow the required format. Novelty or contribution is unclear or unsupported, with

Criteria	Excellent	Satisfactory	Below Expectation
	contribution are clearly articulated, claims and results are well supported, and the work is original with proper citations and no plagiarism.	gaps in supporting evidence or formatting.	evidence of plagiarism or inadequate citation.
Presentation & Oral Communication	Slides are error-free and logically present the work and findings; they are readable and the graphics support the main ideas. Speakers are audible and fluent, do not rely on notes and answer questions accurately. Confident body language and eye contact convey strong engagement.	Slides are error-free and logical and mostly readable. Speakers are mostly fluent with minimal reliance on notes and answer most questions well. Slight discomfort or occasional distracting gestures.	Slides contain errors and lack logical flow; key aspects are absent and graphics confuse the audience. Speakers are inaudible or hesitant and rely heavily on notes, with difficulty answering questions. Body language conveys discomfort.

Prepared by **Checked by** **Head of the Department** **Principal**

Program: Artificial Intelligence and Machine Learning	B.Tech	Semester: VII
Course: AI for Cybersecurity (DJS23ACSC701)		

Pre-requisites:

Computer networks, basic cryptography, and Python data frameworks.

Course Objectives:

1. Understand the fundamentals of Artificial Intelligence and Machine Learning in cybersecurity.
2. Apply AI techniques for threat detection, malware analysis, and network anomaly detection.
3. Utilize Generative AI and Large Language Models (LLMs) for cybersecurity operations.
4. Explore AI-assisted threat intelligence, incident response, and security automation.
5. Understand ethical, legal, and security challenges associated with AI in cybersecurity.

Course Outcomes: On completion of the course, the learner will be able to:

1. Explain the role of AI, ML, and Generative AI in cybersecurity.
2. Apply machine learning techniques for threat detection and cyber defense.
3. Analyze malware and network anomalies using AI-driven approaches.
4. Utilize Generative AI tools for threat hunting, incident response, and security documentation.
5. Demonstrate responsible and secure use of AI technologies in cybersecurity environments.

AI for Cybersecurity (DJS23ACSC701)		
Unit	**Description**	**Duration**
1	**Introduction to AI for Cybersecurity:** Cybersecurity Landscape, Artificial Intelligence Fundamentals, Machine Learning Basics, Deep Learning Concepts, Generative AI and Large Language Models (LLMs), Applications of AI in Cybersecurity, AI-driven Security Operations Centers (SOC), Benefits, Challenges, and Limitations of AI in Security.	5
2	**Machine Learning for Threat Detection and Security Analytics:** Security Data Collection and Analysis, Security Logs and Event Data, Supervised and Unsupervised Learning for Cybersecurity, Intrusion Detection Systems (IDS), User and Entity Behavior Analytics (UEBA), Threat Detection Models, Security Analytics, AI-based Threat Intelligence, Performance Evaluation of Security Models.	5
3	**Advanced Malware and Network Anomaly Detection:** Malware Fundamentals, Static and Dynamic Malware Analysis, AI-based Malware Classification, Ransomware Detection, Malware Behavior Analysis, Network Traffic Analysis, Network Anomaly Detection Techniques, Deep Learning for Intrusion Detection, Detection of DDoS, Botnets and Advanced Persistent Threats (APT), Case Studies.	5
4	**Generative AI for Cyber Defense and Responsible AI:** Prompt Engineering for Cybersecurity, AI-assisted Threat Hunting, Log Analysis using LLMs, Phishing Detection, Vulnerability Assessment, Incident Response Automation, Threat Intelligence Generation, Security Documentation and Reporting, Ethical AI, Adversarial AI, AI Risks and Hallucinations, AI Governance, Privacy and Future Trends in AI-driven Cybersecurity.	5
	Total	**20**

Online Course:

1. Introduction to AI for Cybersecurity: https://www.coursera.org/learn/introduction-to-ai-for-cybersecurity
2. Artificial Intelligence for Cybersecurity: https://www.coursera.org/learn/packt-artificial-intelligence-for-cybersecurity
3. Advanced Malware and Network Anomaly Detection : https://www.coursera.org/learn/advanced-malware-and-network-anomaly-detection
4. IBM Generative AI for Cybersecurity Professionals Specialization: https://www.coursera.org/specializations/generative-ai-for-cybersecurity-professionals

Useful Courses:

1. Generative AI and its Impact on Cybersecurity – https://www.coursera.org/programs/nmims-university-faculty-development-program-kjd4b/learn/packt-generative-ai-and-its-impact-on-cybersecurity-1ge45?authProvider=nmims&source=search
2. ChatGPT for Cybersecurity: AI Tools for Threat Defense – https://www.coursera.org/programs/nmims-university-faculty-development-program-kjd4b/learn/packt-chatgpt-for-cybersecurity-ai-tools-for-threat-defense-f7gee?authProvider=nmims&source=search

Prepared by Checked by HoD Principal

Program: Artificial Intelligence and Machine Learning	B. Tech	Semester: VII/VIII
Course: Disaster Management and Preparedness (DJS23ICHSX11)		

Objectives:

1. To provide basic understanding of hazards, disasters and various types and categories of disaster occurring around the world.
2. To identify extent and damaging capacity of a disaster.
3. To study and understand the means of losses and methods to overcome /minimize it.
4. To understand roles and responsibilities of individual and various organizations during and after disaster.
5. To appreciate the significance of GIS, GPS in the field of disaster management.
6. To understand the emergency government response structures before, during and after disaster.

Course Outcomes: On completion of the course, the learner will be able to:

1. Apply disaster management principles & guidelines.
2. Conduct risk assessments.
3. Develop community awareness & participation.
4. Utilize Science & Technology tools (GIS, GPS).
5. Prepare disaster management plans.

Disaster Management and Preparedness (DJS23ICHSX11)		
Unit	**Description**	**Duration**
1	**Understanding Disasters & Hazards:** • **Definition and types of disasters**: Natural, Man-made and hybrid disasters, **Study of Natural disasters**: Flood, drought, cloud burst, Earthquake, Landslides, Avalanches, Volcanic eruptions; Mudflow, Cyclone, Storm, Storm Surge, climate change, global warming, sea level rise, ozone depletion etc. **Study of Human/Technology Induced Disasters**: Chemical, Industrial and Nuclear disasters, internally displaced persons, road and train accidents Fire Hazards, terrorism, militancy, • Hazard & Vulnerability profiles of India (seismic zones, flood-prone areas). • India's vulnerability to disasters, and the impact of disasters on National development.	06
2	**Disaster Risk Reduction (DRR) & Mitigation:** • **Disaster Management Cycle**: Prevention, Mitigation, Preparedness, Response, Recovery. Need for disaster prevention and mitigation, mitigation guiding principles, challenging areas, structural and non-structural measures for disaster risk reduction. • Risk Assessment & Vulnerability Analysis. • **Science & Technology**: Use of information management, Geo informatics like RS, GIS, GPS and remote sensing mitigation measure.	06

3	**Disaster Preparedness & Response:** • Preparedness Planning, Early Warning Systems (EWS), & Communication. • Emergency Response: Search & Rescue, Logistics, Medical Aid. • Psychological Response & Management (Trauma, Stress). • Role of IT, Media, Govt., NGOs, & Community.	**04**
4	**Recovery, Rehabilitation & Reconstruction:** • Post-disaster damage assessment. • Rehabilitation, Reconstruction, & Livelihood Restoration. • Sanitation, Hygiene, & Waste Management.	**04**
5	**Policy, Governance & Capacity Building:** • National Disaster Management Authority (NDMA) & Legislation. • Institutional Mechanisms & Community Mobilization. Non-Structural Mitigation: Community based disaster preparedness, capacity development and training, awareness and education, contingency plans.	**04**
6	**Case studies on disaster (National /International):** • Case study discussion of National Disasters: Tsunami (2004), Bhopal gas tragedy, Kerala and Uttarakhand flood disaster, 26th July 2005 Mumbai flood • Case study discussion of International Disasters: Hiroshima – Nagasaki (Japan), Cyclone Phailin (2013), Fukushima, Daiichi nuclear disaster (2011), Chernobyl meltdown	**04**
	Total	**28**

Books Recommended:

Reference Books and Reports:

1. Disaster Management, by Harsh K. Gupta, Universities Press Publications (2003).
2. Disaster Management: An Appraisal of Institutional Mechanisms in India, by O. S. Dagur, published by Centre for land warfare studies, New Delhi, 2011.
3. Introduction to International Disaster Management, by Damon Copolla, Butterworth Heinemann Elsevier Publications (2015).
4. Disaster Management Handbook, by Jack Pinkowski, CRC Press, Taylor and Francis group (2008).
5. Disaster management & rehabilitation, by Rajdeep Dasgupta, Mittal Publications, New Delhi (2007).
6. Natural Hazards and Disaster Management, Vulnerability and Mitigation, by R B Singh, Rawat Publications (2006).
7. Concepts and Techniques of GIS, by C. P. Lo Albert, K.W. Yonng, Prentice Hall (India) Publications (2006).
8. Risk management of natural disasters, by Claudia G. Flores Gonzales, KIT Scientific Publishing (2010).
9. Disaster Management – a disaster manger's handbook, by W. Nick Carter, Asian Development Bank (2008).

10. Disaster Management in India, by R. K. Srivastava, Ministry of Home Affairs, GoI, New Delhi (2011)
11. The Chernobyl Disaster: Legacy and Impact on the Future of Nuclear Energy, by Wil Mara, Marshall Cavendish Corporation, New York, 2011.
12. The Fukushima 2011 Disaster, by Ronald Eisler, Taylor & Francis, Florida, 2013.

(Learners are expected to refer reports published at national and international level and updated information available on authentic web sites.)

SVKM's DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING Estd. 1994 CREATING ENGINEERS

Prepared by | Checked by | HoD | Principal