MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. **RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956**

MK University, Patan
Faculty of Engineering Technology,
Department of Artificial Intelligence

M. TECH (ARTIFICIAL INTELLIGENCE) SEM-I									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/WEEK	PRACTICAL (HRS.)/WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	MTAI101	ADVANCED MATHEMATICS FOR ENGINEERS	4	0	4	40	60	100
2	MAJOR	MTAI102	FOUNDATION OF AI & ML	4	2	6	90	60	150
3	MAJOR	MTAI103	DEEP LEARNING & NURAL NETWORKS	4	2	6	90	60	150
4	MINOR	MTAI104	RESEARCH METHDOLOGY & TECHNICAL COMMUNICATION	4	0	4	40	60	100
5	SEC	MTAI105	ENTERPRENURSHIP DEVELOPMENT	4	0	4	40	60	100
TOTAL				20	4	24	300	300	600

M. TECH (ARTIFICIAL INTELLIGENCE) SEM-II									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/WEEK	PRACTICAL (HRS.)/WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	MTAI201	NATURAL LANGUAGE PROCESSING	4	0	4	40	60	100
2	MAJOR	MTAI202	REINFORCEMENT LEARNING	4	2	6	90	60	150
3	MAJOR	MTAI203	COMUTER VISION & IMAGE PROCESSING	4	2	6	90	60	150
4	MINOR	MTAI204	AI FOR IOT & EMBEDDED SYSTEMS	4	2	6	90	60	150
5	VAC	MTAI205	BUSINESS COMMUNICATION-I	2	0	2	0	50	50
TOTAL				18	6	24	310	290	600

M. TECH (ARTIFICIAL INTELLIGENCE) SEM-III

SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/ WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	MTAI301	AI IN ROBOTICS & AUTONOMUS SYSTEMS	4	2	6	90	60	150
2	MAJOR	MTAI302	BIG DATA ANALYTICS & CLOUD AI	4	2	6	90	60	150
3	MINOR	MTAI303	MOOC/SWAYAM COURSE	3	0	3	100	00	100
4	VAC	MTAI304	DISSERTATION PHASE-I	0	8	8	100	100	200
TOTAL				11	12	23	380	220	600

M. TECH (ARTIFICIAL INTELLIGENCE) SEM-IV

SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/ WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	MTAI401	INDUSTRY SEMINARS/WORKSHOPS/INTERNSHIP	0	2	2	50	00	50
2	MINOR	MTAI402	COMPREHENSIVE VIVA VOCE	0	2	2	50	00	50
3	MAJOR	MTAI403	DISEERTATION PHASE-II	0	16	16	200	200	400
4	VAC	MTAI404	BUSINESS COMMUNICATION-II	2	0	2	00	50	50
TOTAL				2	20	22	300	250	550

SEMESTER-I

SUBJECT CODE: MTAI101

SUBJECT NAME: ADVANCED MATHEMATICS FOR ENGINEERS

Course Objectives:

- To provide a rigorous mathematical foundation for advanced engineering modeling and analysis.
- To bridge theoretical mathematics with practical engineering applications.
- To develop problem-solving skills using analytical and computational tools.
- To prepare students for research and development in engineering domains requiring mathematical sophistication.

Course Outcomes: At the end of the course students shall be able to

CO1	Formulate and solve engineering problems using advanced techniques in linear algebra and tensor analysis.
CO2	Apply partial differential equations (PDEs) and transform methods to model dynamical systems and boundary value problems.
CO3	Use variational calculus and optimization methods for engineering design and control problems.
CO4	Analyze stochastic systems and uncertainty propagation using probability theory and statistical methods.

Unit	Content	Credit	Weightage
I	**Advanced Linear Algebra & Tensors for Engineers** ○ Review of vector spaces, eigenvalues, SVD, Jordan form ○ Matrix decompositions (LU, QR, Cholesky, Schur) ○ Tensor algebra: notation, operations, invariants ○ Tensor applications: stress-strain, inertia, constitutive models ○ Numerical linear algebra (conditioning, iterative solvers) • **Applications:** Structural analysis, continuum mechanics, control systems, data compression.	1	25%
II	**Partial Differential Equations & Transform Methods** ○ Classification of PDEs (elliptic, parabolic, hyperbolic) ○ Separation of variables, eigenfunction expansions ○ Green's functions for ODEs and PDEs ○ Integral transforms (Fourier, Laplace, Hankel) for PDEs ○ Introduction to finite element and finite volume concepts	1	25%

	• **Applications:** Heat transfer, wave propagation, fluid dynamics, signal processing.		
III	**Calculus of Variations & Optimization** ○ Functional derivatives, Euler–Lagrange equation ○ Constraints (Lagrange multipliers, isoperimetric problems) ○ Direct methods (Ritz, Galerkin) ○ Optimal control theory (Pontryagin's principle, Hamiltonian formulation) ○ Convex optimization basics (gradient descent, KKT conditions) • **Applications:** Optimal design, trajectory optimization, energy minimization, control systems.	1	25%
IV	**Stochastic Processes & Uncertainty Quantification** ○ Probability spaces, random variables, distributions ○ Stochastic processes (Brownian motion, Poisson process, Markov chains) ○ Itô calculus basics (stochastic differential equations) ○ Uncertainty quantification (Monte Carlo, polynomial chaos, sensitivity analysis) ○ Statistical estimation and regression for engineering data • **Applications:** Risk analysis, reliability engineering, random vibrations, financial engineering, signal noise modelling	1	25%

TEXT BOOKS:

- Kreyszig, E. – *Advanced Engineering Mathematics* (10th ed.) – Wiley.
- Strang, G. – *Linear Algebra and Its Applications* (5th ed.) – Cengage.
- Arfken, G.B., Weber, H.J., Harris, F.E. – *Mathematical Methods for Physicists* (7th ed.) – Academic Press.
- J.N. Reddy – *Applied Functional Analysis and Variational Methods in Engineering* – McGraw-Hill.
- Papoulis, A., & Pillai, S.U. – *Probability, Random Variables and Stochastic Processes* (4th ed.) – McGraw-Hill.

REFERENCE BOOKS:

- Riley, K.F., Hobson, M.P., Bence, S.J. – *Mathematical Methods for Physics and Engineering* (3rd ed.) – Cambridge.
- Gelfand, I.M., & Fomin, S.V. – *Calculus of Variations* – Dover.
- Oksendal, B. – *Stochastic Differential Equations: An Introduction with Applications* (6th ed.) – Springer.
- Holmes, M.H. – *Introduction to Numerical Methods in Differential Equations* – Springer.
- Gould, P. – *Introduction to Linear Elasticity* (for tensor applications) – Springer.

ONLINE RESOURCES:

- Coursera:
  - *Mathematics for Engineers Specialization* (The Hong Kong University of Science and Technology)
  - *Data Science Math Skills* (Duke University)

SUBJECT CODE: MTAI102

SUBJECT NAME: FOUNDATION OF AI & ML

Course Objectives:

- To introduce fundamental concepts of Artificial Intelligence and Machine Learning.
- To develop a strong theoretical understanding of classical AI algorithms and ML models.
- To equip students with practical skills for implementing and evaluating ML algorithms.
- To prepare students for advanced topics in AI, deep learning, and real-world AI applications.

Course Outcomes: At the end of the course students shall be able to

CO1	Understand and explain the core principles of AI, problem-solving, and search algorithms.
CO2	Apply supervised and unsupervised learning algorithms for classification, regression, and clustering tasks.
CO3	Implement and evaluate ML models using Python and popular ML libraries.
CO4	Analyze and interpret model performance and apply feature engineering and model selection techniques.

Unit	Content	Credit	Weightage
I	**Introduction to AI & Intelligent Agents** • History and evolution of AI • Intelligent Agents: Types, environments, and performance measures • Problem-solving: State-space representation, search strategies (BFS, DFS, A*) • Constraint satisfaction problems (CSP) • Game playing: Minimax algorithm, alpha-beta pruning • Knowledge representation: Propositional and predicate logic • **Applications:** Robotics, puzzle solving, game AI	1	25%
II	**Supervised Learning** **Topics:** • Introduction to supervised learning • Linear and logistic regression • Decision trees, entropy, information gain, pruning • Support Vector Machines (linear and kernel-based) • Ensemble methods: Bagging, Boosting, Random Forest • Model evaluation metrics: Accuracy, precision, recall, F1-score, ROC-AUC	1	25%

	• Overfitting, underfitting, bias-variance tradeoff • **Applications:** Spam detection, medical diagnosis, credit scoring		
III	**Unsupervised Learning & Dimensionality Reduction** **Topics:** • Clustering algorithms: K-means, hierarchical clustering, DBSCAN • Gaussian mixture models (GMM) • Dimensionality reduction: PCA, t-SNE, LDA • Anomaly detection techniques • Association rule mining: Apriori algorithm • Evaluation of clustering results: Silhouette score, Davies-Bouldin index • **Applications:** Customer segmentation, image compression, recommendation systems	1	25%
IV	**Model Selection, Feature Engineering & ML Pipeline** **Topics:** • Feature selection and extraction methods • Handling missing data, outliers, and normalization • Cross-validation techniques (k-fold, stratified) • Hyperparameter tuning: Grid search, random search • Introduction to ML pipelines using Scikit-learn • Model interpretability: SHAP, LIME • Ethical considerations in ML: Bias, fairness, transparency • **Applications:** Real-world ML project lifecycle, model deployment basics	1	25%

TEXT BOOKS:

- Russell, S., & Norvig, P. – *Artificial Intelligence: A Modern Approach* (4th ed.) – Pearson.
- Bishop, C.M. – *Pattern Recognition and Machine Learning* – Springer.
- Hastie, T., Tibshirani, R., & Friedman, J. – *The Elements of Statistical Learning* – Springer.
- Géron, A. – *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* – O'Reilly.

REFERENCE BOOKS:

- Murphy, K.P. – *Machine Learning: A Probabilistic Perspective* – MIT Press.
- Alpaydin, E. – *Introduction to Machine Learning* (4th ed.) – MIT Press.
- Marsland, S. – *Machine Learning: An Algorithmic Perspective* – CRC Press.
- Shalev-Shwartz, S., & Ben-David, S. – *Understanding Machine Learning: From Theory to Algorithms* – Cambridge.

ONLINE RESOURCES:

- Coursera: *Machine Learning* (Andrew Ng – Stanford)
- edX: *Introduction to Artificial Intelligence* (MITx)
- Kaggle: Datasets, competitions, and kernels for hands-on practice
- NPTEL: *Introduction to Machine Learning* (IIT Madras)

PRACTICAL LIST:

- 1.Implementation of Search Algorithms
  Task: Implement A* search algorithm to solve the 8-puzzle problem. Visualize the solution path and compare performance with BFS and DFS in terms of time and space complexity.
- 2. Supervised Learning – Classification & Regression
  Task: Use the Iris and Boston Housing datasets. Implement and compare:
  Logistic Regression vs. SVM for classification
  Linear Regression vs. Decision Tree for regression
  Evaluate using appropriate metrics and visualize decision boundaries.
- 3.Unsupervised Learning – Clustering & PCA
  Task: Apply K-means and hierarchical clustering on the Wine dataset. Use PCA for dimensionality reduction and visualize clusters in 2D. Compare clustering results using Silhouette score.
- 4. End-to-End ML Pipeline
  Task: Build a complete ML pipeline for a real dataset (e.g., Titanic survival prediction). Perform:
  Data cleaning and feature engineering
  Model training (Random Forest, SVM)
  Hyperparameter tuning using GridSearchCV
  Model evaluation and interpretation using SHAP/LIME

SUBJECT CODE: MTAI103

SUBJECT NAME: DEEP LEARNING AND NEURAL NETWORKS

Course Objectives:

- To provide a comprehensive understanding of neural network architectures and deep learning fundamentals.
- To develop expertise in designing, training, and optimizing various deep learning models.
- To apply deep learning techniques to solve real-world problems in computer vision, NLP, and sequential data.
- To prepare students for research and development in cutting-edge AI applications and emerging deep learning paradigms.

Course Outcomes: At the end of the course students shall be able to

CO1	Design and implement various neural network architectures including CNNs, RNNs, and transformers.
CO2	Train and optimize deep learning models using appropriate regularization, optimization, and hyperparameter tuning techniques.
CO3	Apply deep learning to practical domains such as image recognition, natural language processing, and time-series prediction.
CO4	Evaluate model performance, interpret results, and implement deployment-ready deep learning solutions.

Unit	Content	Credit	Weightage
I	**Neural Network Fundamentals and Optimization** **Topics:** • Biological inspiration and artificial neurons	1	25%

	• Perceptron, multi-layer perceptron (MLP) • Activation functions: Sigmoid, Tanh, ReLU, Leaky ReLU, Softmax • Backpropagation algorithm and gradient computation • Loss functions: MSE, Cross-entropy, Hinge loss • Optimization algorithms: SGD, Momentum, RMSProp, Adam • Regularization: Dropout, L1/L2, Batch normalization • Hyperparameter tuning and learning rate schedules • **Applications:** Digit recognition, basic classification tasks		
II	**Convolutional Neural Networks (CNNs)** **Topics:** • Convolution operations and feature extraction • CNN architectures: LeNet, AlexNet, VGG, ResNet, Inception • Pooling layers: Max pooling, average pooling • Transfer learning and fine-tuning • Data augmentation techniques • Object detection basics: R-CNN, YOLO, SSD • Semantic segmentation: U-Net, FCN • Visualization: Feature maps, Grad-CAM • **Applications:** Image classification, object detection, medical imaging	1	25%
III	**Recurrent Neural Networks and Sequence Models** **Topics:** • RNN architecture and limitations (vanishing/exploding gradients) • LSTM and GRU cells • Bidirectional RNNs • Sequence-to-sequence models • Attention mechanism • Introduction to Transformers • Time-series forecasting with RNNs • **Applications:** Language modeling, machine translation, stock prediction, speech recognition	1	25%
IV	**Advanced Architectures and Emerging Topics** **Topics:** • Autoencoders: Variational Autoencoders (VAEs) • Generative Adversarial Networks (GANs) • Self-supervised learning • Graph Neural Networks (GNNs) • Neural Architecture Search (NAS) • Federated learning basics • Model compression: Pruning, quantization,	1	25%

	knowledge distillation • Ethical considerations in deep learning • **Applications:** Image generation, recommendation systems, drug discovery		

TEXT BOOKS:

- Goodfellow, I., Bengio, Y., & Courville, A. – *Deep Learning* – MIT Press.
- Chollet, F. – *Deep Learning with Python* (2nd ed.) – Manning Publications.
- Aggarwal, C.C. – *Neural Networks and Deep Learning: A Textbook* – Springer.
- Bishop, C.M. – *Pattern Recognition and Machine Learning* – Springer.

REFERENCE BOOKS:

- Geron, A. – *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed.) – O'Reilly.
- Zhang, A., Lipton, Z.C., Li, M., & Smola, A.J. – *Dive into Deep Learning* – Cambridge University Press.
- Raschka, S., & Mirjalili, V. – *Python Machine Learning* (3rd ed.) – Packt Publishing.
- Howard, J., & Gugger, S. – *Deep Learning for Coders with fastai and PyTorch* – O'Reilly.

ONLINE RESOURCES:

- Practical deep learning courses
- Coursera: *Deep Learning Specialization*
- edX: *Deep Learning Fundamentals* (IBM)
- Kaggle: Deep learning competitions and datasets
- Papers With Code: State-of-the-art implementations

PRACTICAL LIST:

1. Building and Training Basic Neural Networks

- Task: Implement a Multi-Layer Perceptron from scratch using NumPy for MNIST digit classification. Compare with TensorFlow/PyTorch implementation. Experiment with different activation functions, optimizers, and regularization techniques.

1. CNN Architecture for Image Classification

Task: Design and train a CNN model on CIFAR-10 dataset. Implement data augmentation, batch normalization, and dropout. Compare performance of custom CNN with pre-trained models (ResNet50) using transfer learning. Visualize learned filters and feature maps.

2. Sequence Modeling with RNN/LSTM

- Task: Build an LSTM-based model for text generation using Shakespeare's works dataset. Implement character-level and word-level models. Extend to sentiment analysis on IMDB reviews using Bi-directional LSTM with attention mechanism.

1. Advanced Models: Autoencoders and GANs

Task:
Part A: Implement a Variational Autoencoder (VAE) for MNIST digit generation and anomaly detection.
Part B: Build a DCGAN (Deep Convolutional GAN) to generate realistic faces using CelebA dataset.
Compare results and discuss mode collapse, training stability, and evaluation metrics.

SUBJECT CODE: MTAI104

SUBJECT NAME: RESEARCH METHDOLOGY AND TECHNICAL COMMUNICATION

Course Objectives:

- To equip engineering graduates with a structured approach to scientific inquiry and problem-solving.
- To develop proficiency in selecting and applying appropriate research methods for engineering investigations.
- To enhance technical communication skills for academia and industry.
- To foster an understanding of research ethics, scholarly publishing, and lifelong learning in research.

Course Outcomes: At the end of the course students shall be able to

CO1	Formulate a research problem, conduct systematic literature reviews, and develop a viable research proposal.
CO2	Design and execute appropriate research methodologies (experimental, numerical, analytical) with consideration for ethics and data integrity.
CO3	Apply statistical tools and software for data analysis, interpretation, and validation of research findings.
CO4	Produce high-quality technical documents (research papers, proposals, theses) and deliver effective technical presentations.

Unit	Content	Credit	Weightage

I	**Foundations of Engineering Research & Problem Formulation**
	• **Topics:**
	  ○ Philosophy of research: inductive vs. deductive reasoning, scientific method in engineering.
	  ○ Types of engineering research: fundamental, applied, experimental, computational, empirical.
	  ○ Problem identification and formulation: research gap analysis.
	  ○ Literature review strategies: databases (Scopus, Web of Science, IEEE Xplore), citation management tools (Zotero, Mendeley), critical analysis of literature.
	  ○ Developing a research proposal: objectives, scope, significance, and work plan.
	• **Applications:** Thesis topic selection, grant proposal writing, project planning.
		1	25%

II	**Research Design, Methods & Ethics**
	• **Topics:**
	  ○ Research design: experimental, quasi-experimental, case study, modeling & simulation.
	  ○ Data collection methods: sensors, surveys, instrumentation, simulation outputs.
	  ○ Design of Experiments (DoE): factorial
		1	25%

	design, Taguchi methods, response surface methodology. ◦ Research ethics: plagiarism, fabrication/falsification, authorship, informed consent. ◦ Ethical approval process and responsible conduct of research (RCR). • **Applications:** Planning a lab/field experiment, setting up a CFD/FEA study, survey design.		
III	**Data Analysis, Statistics & Software Tools** • **Topics:** ◦ Data preprocessing: outlier detection, missing data, normalization. ◦ Descriptive and inferential statistics: hypothesis testing (t-test, ANOVA), confidence intervals. ◦ Regression analysis: linear, multiple, logistic. ◦ Introduction to multivariate analysis and machine learning for engineering data. ◦ Software tools: MATLAB/Python (NumPy, SciPy, pandas), R, MiniTab. ◦ Data visualization principles: effective graphs, charts, and plots. • **Applications:** Analyzing experimental results, validating computational models, interpreting sensor data.	1	25%
IV	**Technical Communication & Research Dissemination** • **Topics:** ◦ Structure of technical documents: research papers, theses, technical reports. ◦ Writing strategies: clarity, conciseness, coherence, and argument development. ◦ Graphical abstracts, data presentation, and table/figure design. ◦ Oral presentations: conference talks, thesis defense, poster design. ◦ Publication process: journal selection, peer review, responding to reviewers. ◦ Intellectual Property Rights (IPR): patents, copyrights, licensing. ◦ Research dissemination: repositories, academic social networks (ResearchGate, LinkedIn), and impact metrics (h-index, citations). • **Applications:** Paper writing, thesis compilation, conference presentation, patent filing.	1	25%

TEXT BOOKS:

- Kothari, C.R. – *Research Methodology: Methods and Techniques* (4th ed.) – New Age International.
- Day, R.A., and Gastel, B. – *How to Write and Publish a Scientific Paper* (9th ed.) – Greenwood.
- Montgomery, D.C. – *Design and Analysis of Experiments* (10th ed.) – Wiley.
- Alley, M. – *The Craft of Scientific Writing* (4th ed.) – Springer.

REFERENCE BOOKS:

- Bordens, K.S., and Abbott, B.B. – *Research Design and Methods: A Process Approach* (11th ed.) – McGraw-Hill.
- Wallwork, A. – *English for Writing Research Papers* (2nd ed.) – Springer.
- Box, G.E.P., Hunter, J.S., and Hunter, W.G. – *Statistics for Experimenters* (2nd ed.) – Wiley.
- IEEE Author Center Guides – *IEEE Publication Services and Products Board*.
- Laplante, P.A. – *Technical Writing: A Practical Guide for Engineers and Scientists* – CRC Press.

ONLINE RESOURCES:

- edX Courses:
    1. *"Principles of Statistical Analysis"* (Microsoft)
    2. *"How to Write and Publish a Scientific Paper"* (KU Leuven)

SEMESTER-II

SUBJECT CODE: MTAI201

SUBJECT NAME: NATURAL LANGUAGE PROCESSING

Course Objectives:

- To provide a comprehensive understanding of linguistic foundations and computational techniques for natural language understanding and generation.
- To develop expertise in modern NLP techniques including statistical models, neural approaches, and transformer architectures.
- To implement and evaluate NLP systems for real-world applications such as text classification, machine translation, and conversational AI.
- To prepare students for research and development in cutting-edge NLP technologies and emerging language AI applications.

Course Outcomes: At the end of the course students shall be able to

CO1	Understand linguistic structures and apply preprocessing techniques for text data.
CO2	Design and implement statistical and neural NLP models for various language tasks.
CO3	Build and fine-tune transformer-based models for advanced NLP applications.
CO4	Develop end-to-end NLP systems and evaluate their performance with appropriate metrics.

Unit	Content	Credit	Weightage
I	**Foundations of NLP and Text Processing** **Topics:** • Introduction to linguistics: Morphology, syntax, semantics, pragmatics • Text preprocessing: Tokenization, stemming, lemmatization, stop-word removal • Regular expressions and pattern matching • N-gram language models and smoothing techniques • Word representations: One-hot encoding, TF-IDF, co-occurrence matrices • Collocations and statistical measures (PMI, chi-square) • Named Entity Recognition (NER) using rule-based and statistical methods • **Applications:** Text normalization, spell checking, basic information extraction	1	25%
II	**Statistical NLP and Classical Approaches** **Topics:** • Part-of-Speech tagging: Hidden Markov Models, Viterbi algorithm • Context-Free Grammars and parsing algorithms (CKY, Earley)	1	25%

	• Dependency parsing and constituency parsing • Semantic analysis: Word sense disambiguation, semantic roles • Sentiment analysis using lexicon-based and machine learning approaches • Topic modeling: Latent Dirichlet Allocation (LDA) • Text classification using Naïve Bayes, SVM, and logistic regression • **Applications:** Document classification, opinion mining, topic extraction		
III	**Neural NLP and Word Embeddings** **Topics:** • Distributed representations: Word2Vec (CBOW, Skip-gram), GloVe • Contextual embeddings: ELMo, ULMFiT • Neural sequence models for NLP tasks • Sequence-to-sequence models with attention mechanism • Neural machine translation • Text generation with RNNs and LSTMs • Neural text classification and sentiment analysis • **Applications:** Semantic similarity, document clustering, basic chatbots	1	25%
IV	**Transformer Models and Advanced NLP** **Topics:** • Transformer architecture: Self-attention, multi-head attention, positional encoding • BERT and its variants (RoBERTa, DistilBERT, ALBERT) • GPT models and autoregressive language modeling • Fine-tuning pre-trained language models • Question Answering systems (SQuAD) • Text summarization: Extractive and abstractive approaches • Dialogue systems and conversational AI • Ethical considerations: Bias in language models, fairness, interpretability • **Applications:** Chatbots, document summarization, intelligent search systems	1	25%

TEXT BOOKS:

- Jurafsky, D., & Martin, J.H. – *Speech and Language Processing* (3rd ed. draft) – Pearson.
- Eisenstein, J. – *Introduction to Natural Language Processing* – MIT Press.
- Goldberg, Y. – *Neural Network Methods for Natural Language Processing* – Morgan & Claypool.
- Clark, K., Khandelwal, U., Levy, O., & Manning, C.D. – *What Does BERT Look At?* – ACL.

REFERENCE BOOKS:

- Manning, C.D., & Schütze, H. – *Foundations of Statistical Natural Language Processing* – MIT Press.
- Bird, S., Klein, E., & Loper, E. – *Natural Language Processing with Python* – O'Reilly.
- Sarkar, D. – *Text Analytics with Python* – Apress.
- Howard, J., & Ruder, S. – *Universal Language Model Fine-tuning for Text Classification* – ACL.

ONLINE RESOURCES:

- Coursera: *Natural Language Processing Specialization*
- Stanford Online: *CS224N: Natural Language Processing with Deep Learning*
- Kaggle: NLP competitions and datasets
- Google Colab: GPU access for training NLP models

SUBJECT CODE: MTAI202

SUBJECT NAME: REINFORCEMENT LEARNING

Course Objectives:

- To provide a comprehensive theoretical foundation of reinforcement learning principles, Markov decision processes, and value-based methods.
- To develop expertise in implementing and analyzing various RL algorithms including policy gradient methods, actor-critic architectures, and deep RL.
- To apply RL techniques to solve complex sequential decision-making problems in robotics, gaming, finance, and autonomous systems.
- To prepare students for research and development in advanced RL topics, multi-agent systems, and real-world RL applications.

Course Outcomes: At the end of the course students shall be able to

CO1	Formulate real-world problems as Markov Decision Processes and understand RL fundamentals.
CO2	Implement and evaluate value-based, policy-based, and model-based RL algorithms.
CO3	Design and train deep reinforcement learning agents for complex environments.
CO4	Apply RL to multi-agent systems, robotics, and emerging applications with ethical considerations.

Unit	Content	Credit	Weightage
I	**Foundations of Reinforcement Learning** **Topics:** • Introduction to sequential decision-making • Markov Decision Processes (MDPs): States, actions, transitions, rewards • Policy, value functions, and Bellman equations • Dynamic programming: Policy iteration, value iteration • Monte Carlo methods for prediction and control • Temporal Difference Learning: SARSA, Q-learning • Exploration vs. exploitation strategies (ε-greedy,	1	25%

	UCB, softmax) • **Applications:** Grid-world navigation, basic game playing		
II	**Value-Based and Model-Based Methods** **Topics:** • Function approximation in RL: Linear and nonlinear • Deep Q-Networks (DQN) and improvements (Double DQN, Dueling DQN) • Experience replay and target networks • Model-based RL: Dyna, MCTS (Monte Carlo Tree Search) • Partial Observable MDPs (POMDPs) • Multi-step TD learning (n-step, λ-returns) • Eligibility traces • **Applications:** Atari game playing, inventory management	1	25%
III	**Policy Gradient and Actor-Critic Methods** **Topics:** • Policy gradient theorem • REINFORCE algorithm • Natural policy gradients • Actor-Critic architectures • Advantage Actor-Critic (A2C, A3C) • Trust Region Policy Optimization (TRPO) • Proximal Policy Optimization (PPO) • Deterministic policy gradients (DDPG) • **Applications:** Robotic control, continuous control tasks	1	25%
IV	**Advanced Topics and Real-World Applications** **Topics:** • Hierarchical RL and options framework • Multi-agent reinforcement learning • Inverse reinforcement learning and imitation learning • Meta-reinforcement learning • Safe RL and constraint optimization • RL in robotics: Simulation to real-world transfer • Ethical considerations in autonomous decision-making • **Applications:** Autonomous vehicles, financial trading, healthcare decision support	1	25%

TEXT BOOKS:

- Sutton, R.S., & Barto, A.G. – *Reinforcement Learning: An Introduction* (2nd ed.) – MIT Press.
- Szepesvári, C. – *Algorithms for Reinforcement Learning* – Morgan & Claypool.

- Francois-Lavet, V., Henderson, P., Islam, R., Bellemare, M.G., & Pineau, J. – *An Introduction to Deep Reinforcement Learning* – Now Publishers.
- Wiering, M., & van Otterlo, M. – *Reinforcement Learning: State-of-the-Art* – Springer.

REFERENCE BOOKS:

- Goodfellow, I., Bengio, Y., & Courville, A. – *Deep Learning* (RL Chapters) – MIT Press.
- Graesser, L., & Keng, W.L. – *Foundations of Deep Reinforcement Learning* – Addison-Wesley.
- Moerland, T.M., Broekens, J., & Jonker, C.M. – *Reinforcement Learning in Finance* – Springer.
- Kober, J., Bagnell, J.A., & Peters, J. – *Reinforcement Learning in Robotics: A Survey* – IJRR.

ONLINE RESOURCES:

- DeepMind Lab & DM Control Suite: Advanced RL environments
- Coursera: *Reinforcement Learning Specialization* (University of Alberta)
- Udacity: *Reinforcement Learning Nanodegree*
- Kaggle: RL competitions

PRACTICAL LIST:

1.**Tabular RL Methods Implementation**

Task: Implement and compare tabular RL algorithms (Q-learning, SARSA, Expected SARSA) on FrozenLake and CliffWalking environments. Analyze convergence properties, exploration strategies, and performance metrics. Visualize learned policies and value functions.

2. **Deep Q-Learning for Game Playing**

Task: Train a DQN agent to play Atari games (Pong or Breakout) using OpenAI Gym. Implement experience replay, target networks, and reward clipping. Compare performance with Double DQN and Dueling DQN variants. Analyze training stability and sample efficiency.

3.**Policy Gradient Methods for Continuous Control**

Task: Implement REINFORCE, A2C, and PPO algorithms for continuous control tasks in MuJoCo environments (HalfCheetah, Hopper). Compare sample efficiency, convergence speed, and final performance. Visualize policy evolution and analyze variance reduction techniques.

4. **Advanced RL: Multi-Agent or Robotics Application**

Task Option A (Multi-Agent): Implement independent Q-learning and MADDPG in multi-agent particle environments. Analyze emergent behaviors, cooperation, and competition dynamics.

Task Option B (Robotics): Train a robotic arm to perform reaching/grasping tasks using RL in PyBullet. Implement domain randomization and analyze sim-to-real transfer challenges.

SUBJECT CODE: MTAI203

SUBJECT NAME: COMPUTER VISION AND IMAGE PROCESSING

Course Objectives:

- To provide a comprehensive understanding of fundamental image processing techniques and computer vision algorithms.
- To develop expertise in feature extraction, object detection, segmentation, and 3D vision using classical and deep learning approaches.
- To implement and evaluate computer vision systems for real-world applications in medical imaging, autonomous vehicles, surveillance, and multimedia.
- To prepare students for research and development in cutting-edge vision technologies, including generative vision models and video understanding.

Course Outcomes: At the end of the course students shall be able to

CO1	Apply image processing techniques for enhancement, restoration, and feature extraction.
CO2	Design and implement classical and deep learning-based computer vision algorithms.
CO3	Develop object detection, segmentation, and recognition systems for practical applications.
CO4	Build and evaluate 3D vision, video analysis, and generative vision models.

Unit	Content	Credit	Weightage
I	**Fundamentals of Image Processing** **Topics:** • Digital image fundamentals: Sampling, quantization, color spaces (RGB, HSV, LAB) • Image enhancement: Histogram equalization, contrast stretching, spatial filtering • Image restoration: Noise models, spatial and frequency domain filtering • Morphological operations: Dilation, erosion, opening, closing • Edge detection: Sobel, Canny, Laplacian of Gaussian • Image segmentation: Thresholding, region-based, watershed algorithm • Frequency domain analysis: Fourier transform, filtering in frequency domain • **Applications:** Medical image enhancement, document image processing, noise reduction	1	25%
II	**Feature Extraction and Classical Vision** **Topics:** • Feature detection: Harris corner, SIFT, SURF, ORB • Feature description and matching	1	25%

	• Geometric transformations: Affine, projective, homography • Camera models and calibration • Epipolar geometry and stereo vision • Structure from Motion (SfM) • Optical flow and motion estimation • Image stitching and panorama creation • **Applications:** Augmented reality, robotic navigation, 3D reconstruction		
III	**Deep Learning for Computer Vision** **Topics:** • CNN architectures for vision: AlexNet, VGG, ResNet, Inception • Transfer learning and fine-tuning for vision tasks • Object detection: R-CNN, Fast R-CNN, Faster R-CNN, YOLO, SSD • Semantic segmentation: FCN, U-Net, DeepLab • Instance segmentation: Mask R-CNN • Face recognition and verification • Visual attention mechanisms • Vision transformers (ViT) • **Applications:** Autonomous driving, medical diagnosis, surveillance systems	1	25%
IV	**Advanced Topics and Emerging Vision Systems** **Topics:** • 3D computer vision: Point clouds, voxel representations, 3D CNNs • Generative models for vision: VAEs, GANs (StyleGAN, CycleGAN) • Neural radiance fields (NeRF) • Video understanding: Action recognition, video captioning • Self-supervised learning for vision • Few-shot and zero-shot learning • Ethical considerations: Bias in vision systems, privacy, fairness • **Applications:** Augmented/virtual reality, creative AI, video analytics	1	25%

TEXT BOOKS:

- Szeliski, R. – *Computer Vision: Algorithms and Applications* (2nd ed.) – Springer.
- Gonzalez, R.C., & Woods, R.E. – *Digital Image Processing* (4th ed.) – Pearson.
- Goodfellow, I., Bengio, Y., & Courville, A. – *Deep Learning* (Vision Chapters) – MIT Press.
- Prince, S.J.D. – *Computer Vision: Models, Learning, and Inference* – Cambridge University Press.

REFERENCE BOOKS:

- Bradski, G., & Kaehler, A. – *Learning OpenCV* (4th ed.) – O'Reilly.
- Forsyth, D.A., & Ponce, J. – *Computer Vision: A Modern Approach* – Pearson.
- Chollet, F. – *Deep Learning with Python* – Manning Publications.
- Hartley, R., & Zisserman, A. – *Multiple View Geometry in Computer Vision* – Cambridge University Press.

ONLINE RESOURCES:

- OpenCV: Comprehensive computer vision library
- Kaggle: Computer vision competitions and datasets
- Coursera: *Deep Learning Specialization* (Computer Vision course)

PRACTICAL LIST:

1. **Classical Image Processing Pipeline**
Task: Build a complete image processing pipeline including noise removal, edge detection, and morphological operations. Implement image stitching for panorama creation using feature detection (SIFT/SURF) and homo graphy estimation. Evaluate on real-world image datasets.
2. **Object Detection and Recognition System**
Task: Implement and compare YOLOv5 and Faster R-CNN for object detection on COCO dataset. Fine-tune models for custom object detection (e.g., traffic signs, medical abnormalities). Evaluate using mAP metrics and analyze failure cases.
3. **Medical Image Segmentation**
Task: Develop a U-Net based segmentation system for medical images (e.g., brain MRI, lung CT scans). Implement data augmentation, Dice loss, and evaluate segmentation accuracy. Compare with traditional segmentation methods (watershed, thresholding).
4. **Generative Vision Models**
Task: Implement a CycleGAN for unpaired image-to-image translation (e.g., horses↔zebras, summer↔winter). Train a StyleGAN for high-resolution face generation. Evaluate using FID scores and analyze latent space interpolation.

SUBJECT CODE: MTAI204

SUBJECT NAME: AI FOR IOT AND EMBEDDED SYSTEMS

Course Objectives:

- To provide a comprehensive understanding of IoT architecture, embedded systems design, and AI integration at the edge.
- To develop expertise in deploying machine learning models on resource-constrained devices and IoT platforms.
- To implement AI-driven IoT solutions for real-world applications in smart cities, healthcare, industrial automation, and environmental monitoring.
- To prepare students for designing scalable, secure, and efficient AIoT systems with considerations for energy efficiency and real-time processing.

Course Outcomes: At the end of the course students shall be able to

CO1	Design and implement IoT systems with embedded AI capabilities.
CO2	Optimize and deploy machine learning models on microcontrollers and edge devices.
CO3	Develop secure, energy-efficient AIoT solutions for real-

	world applications.
CO4	Evaluate and integrate AIoT systems with cloud platforms and communication protocols.

Unit	Content	Credit	Weightage
I	**IoT Architecture and Embedded AI Foundations** **Topics:** • IoT ecosystem and layered architecture (sensing, networking, processing, application) • Embedded system components: Microcontrollers (ARM Cortex-M, ESP32), sensors, actuators • Real-time operating systems (FreeRTOS, Zephyr) for IoT • Edge computing vs. cloud computing paradigms • Introduction to TinyML: ML on microcontrollers • Energy harvesting and power management in IoT • Wireless communication protocols: Bluetooth Low Energy, Zigbee, LoRaWAN • **Applications:** Smart sensors, wearable devices, environmental monitoring	1	25%
II	**Model Optimization for Embedded Deployment** **Topics:** • Model compression techniques: Pruning, quantization, knowledge distillation • Hardware-aware neural architecture search (NAS) • Fixed-point arithmetic and integer-only inference • Memory optimization for embedded deployment • Latency and throughput optimization strategies • Benchmarking AI models on embedded hardware • Frameworks: TensorFlow Lite Micro, PyTorch Mobile, CMSIS-NN • **Applications:** Keyword spotting, gesture recognition, anomaly detection on edge	1	25%
III	**Edge AI Implementation and Real-time Processing** **Topics:** • Implementing neural networks on MCUs (STM32, Arduino Nano 33 BLE) • Real-time sensor data processing with AI • Federated learning for privacy-preserving IoT • On-device training vs. inference-only deployment • Edge AI hardware: Google Coral, NVIDIA Jetson, Raspberry Pi AI kits • Time-series analysis on edge devices • Audio and vision processing on embedded systems • **Applications:** Predictive maintenance, voice-controlled IoT, smart cameras	1	25%
IV	**Advanced AIoT Systems and Security**	1	25%

	Topics: • AI-driven IoT security: Anomaly detection, intrusion prevention • Secure firmware updates and encrypted communication • Blockchain for IoT security and data integrity • Scalable AIoT architectures for smart cities and Industry 4.0 • Digital twins and simulation for AIoT • Energy-efficient AI algorithms for battery-operated devices • Ethical considerations: Privacy, data ownership, bias in edge AI • **Applications:** Industrial IoT, healthcare monitoring, autonomous agricultural systems		

TEXT BOOKS:

- Situnayake, D., & Plunkett, T. – *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers* – O'Reilly.
- Ray, P.P. – *AI and IoT-Based Intelligent Automation in Robotics* – Wiley.
- Mazidi, M.A., Naimi, S., & Naimi, S. – *The STM32F103 ARM Cortex-M3 Microcontroller* – MicroDigitalEd.
- Verma, G., & Srivastava, S. – *Internet of Things: Architecture and Design Principles* – McGraw-Hill.

REFERENCE BOOKS:

- Warden, P., & Situnayake, D. – *TinyML Cookbook* – O'Reilly.
- Noergaard, T. – *Embedded Systems Architecture* (2nd ed.) – Newnes.
- Bahga, A., & Madisetti, V. – *Internet of Things: A Hands-On Approach* – VPT.
- Yiu, J. – *The Definitive Guide to ARM Cortex-M0 and Cortex-M0+ Processors* – Newnes.

ONLINE RESOURCES:

- Edge Impulse: End-to-end TinyML development platform
- Arduino Create: Cloud IDE for embedded AI
- TensorFlow Lite Micro Examples & Documentation
- Coursera: *Introduction to Embedded Machine Learning* (Edge Impulse)

PRACTICAL LIST:

1. **Tiny ML Deployment on Microcontroller**
Task: Train a neural network for keyword spotting ("yes", "no", "stop", "go") using TensorFlow, convert to TensorFlow Lite Micro, and deploy on Arduino Nano 33 BLE Sense. Evaluate accuracy, latency, and power consumption.
2. **Real-time Sensor Analytics with Edge AI**
Task: Build an anomaly detection system for industrial equipment using vibration sensors and an STM32 microcontroller. Implement real-time inference and alert generation. Compare edge vs. cloud processing latency.
3. **Vision-based IoT System with Edge Processing**
Task: Develop a smart camera system using Raspberry Pi with Google Coral USB

Accelerator for object detection. Implement people counting or mask detection. Optimize model for real-time (30 FPS) performance.

4. **Secure A IoT System with Cloud Integration**

Task: Create an end-to-end A IoT system with ESP32 collecting environmental data, performing edge inference for air quality prediction, securely transmitting results to AWS IoT Core, and visualizing on a dashboard with alerting mechanisms.

SEMESTER-III

SUBJECT CODE: MTAI301

SUBJECT NAME: AI IN ROBOTICS AND AUTONOMOUS SYSTEMS

Course Objectives:

- To provide a comprehensive understanding of robotic perception, planning, control, and AI integration for autonomous systems.
- To develop expertise in implementing computer vision, sensor fusion, and machine learning algorithms for robotic navigation and manipulation.
- To design and simulate autonomous systems using modern frameworks (ROS, Gazebo) and deploy AI models on physical robots.
- To prepare students for research and industry roles in autonomous vehicles, industrial robotics, drones, and intelligent automation.

Course Outcomes: At the end of the course students shall be able to

CO1	Design robotic perception systems using sensors, computer vision, and sensor fusion techniques.
CO2	Implement path planning, motion control, and decision-making algorithms for autonomous navigation.
CO3	Develop and train AI models for robotic manipulation, object interaction, and task execution.
CO4	Build and evaluate end-to-end autonomous systems using ROS and simulation environments.

Unit	Content	Credit	Weightage
I	**Robotic Perception and State Estimation** • Robotic sensors: LiDAR, cameras, IMU, GPS, ultrasonic, depth sensors • Camera models and calibration for robotics • Feature extraction and matching for robotic vision • Visual odometry and SLAM (Simultaneous Localization and Mapping) • Sensor fusion: Kalman filters, Extended Kalman Filters (EKF), Particle filters • Point cloud processing with PCL (Point Cloud Library) • Object detection and recognition for robotics • **Applications:** Autonomous vehicle perception, drone navigation, warehouse robotics	1	25%
II	**Motion Planning and Control** **Topics:** • Configuration space and motion planning fundamentals • Sampling-based planners: RRT, RRT*, PRM • Optimization-based planning: Trajectory optimization	1	25%

	• Kinematics and dynamics of robotic manipulators • PID control, computed torque control • Model Predictive Control (MPC) for robotics • Collision avoidance and reactive navigation • **Applications:** Robotic arm path planning, autonomous driving maneuvers, drone flight control		
III	**Learning for Robotics and Autonomous Decision Making** **Topics:** • Reinforcement learning for robotic control • Imitation learning and behavior cloning • Deep learning for robotic perception and control • End-to-end learning for autonomous driving • Multi-agent robotics and swarm intelligence • Human-robot interaction and collaborative robotics • Task and motion planning with learning • **Applications:** Autonomous warehouse robots, robotic surgery, adaptive manufacturing	1	25%
IV	**Advanced Topics and System Integration** **Topics:** • Robot Operating System (ROS 2) architecture and tools • Simulation environments: Gazebo, CARLA, AirSim • Real-time systems and embedded AI for robotics • Safety, verification, and ethics in autonomous systems • Edge computing for robotics • Digital twins and virtual testing • Emerging trends: Soft robotics, bio-inspired robotics, neuromorphic computing • **Applications:** Self-driving cars, delivery drones, robotic exoskeletons, space robotics	1	25%

TEXT BOOKS:

- Siciliano, B., & Khatib, O. – *Springer Handbook of Robotics* (2nd ed.) – Springer.
- Thrun, S., Burgard, W., & Fox, D. – *Probabilistic Robotics* – MIT Press.
- Corke, P. – *Robotics, Vision and Control: Fundamental Algorithms in MATLAB* (2nd ed.) – Springer.
- Kober, J., Bagnell, J.A., & Peters, J. – *Reinforcement Learning in Robotics: A Survey* – IJRR.

REFERENCE BOOKS:

- Lynch, K.M., & Park, F.C. – *Modern Robotics: Mechanics, Planning, and Control* – Cambridge University Press.
- LaValle, S.M. – *Planning Algorithms* – Cambridge University Press.
- Kelly, A. – *Mobile Robotics: Mathematics, Models, and Methods* – Cambridge University Press.

- Quigley, M., Gerkey, B., & Smart, W.D. – *Programming Robots with ROS: A Practical Introduction* – O'Reilly.

ONLINE RESOURCES:

- ROS (Robot Operating System): Official documentation, tutorials, packages
- Gazebo & Webots: Robotic simulation platforms
- CARLA & AirSim: Autonomous vehicle simulators
- NVIDIA Isaac Sim: Robotics simulation with AI
- Coursera: *Robotics Specialization* (University of Pennsylvania)
- edX: *Robotics: Perception* (University of Pennsylvania)

PRACTICAL LIST:

1. **SLAM Implementation with ROS and LiDAR**
Task: Implement a 2D SLAM system using ROS, a simulated LiDAR sensor in Gazebo, and GMapping/RTAB-Map. Deploy on a TurtleBot3 or similar robot. Evaluate map accuracy and localization performance.
2. **Path Planning and Autonomous Navigation**
Task: Develop a complete navigation stack in ROS including global planner (A*/Dijkstra), local planner (DWA/TEB), and obstacle avoidance. Test in simulation and optionally on physical hardware.
3. **Robotic Manipulation with AI**
Task: Train a reinforcement learning agent (using PyBullet or MuJoCo) to perform robotic manipulation tasks (pick-and-place, door opening). Compare with traditional inverse kinematics solutions.
4. **End-to-End Autonomous System**
Task: Build an autonomous drone or rover system integrating perception (object detection with YOLO), planning (RRT*), and control (PID/MPC). Simulate in AirSim/Gazebo and analyze system performance metrics.

SUBJECT CODE: MTAI302

SUBJECT NAME: BIG DATA ANALYTICS AND CLOUD AI

Course Objectives:

- To provide a comprehensive understanding of big data technologies, distributed computing frameworks, and cloud-based AI/ML services.
- To develop expertise in processing, analyzing, and deriving insights from massive-scale datasets using scalable cloud platforms.
- To implement end-to-end machine learning pipelines on cloud infrastructure for production-level AI systems.
- To prepare students for roles in data engineering, cloud AI solution architecture, and large-scale AI system deployment.

Course Outcomes: At the end of the course students shall be able to

CO1	Design and implement scalable data processing pipelines using big data technologies.
CO2	Develop and deploy machine learning models on cloud platforms using managed AI services.
CO3	Optimize distributed ML training and inference for large-scale datasets.
CO4	Architect and implement production-grade cloud-native AI systems with MLOps principles.

Unit	Content	Credit	Weightage
I	**Big Data Foundations and Distributed Processing** **Topics:** • Introduction to Big Data: 5 Vs (Volume, Velocity, Variety, Veracity, Value) • Hadoop ecosystem: HDFS, MapReduce, YARN • Apache Spark: RDDs, DataFrames, Spark SQL, Spark MLlib • Distributed data processing patterns and optimizations • NoSQL databases: MongoDB, Cassandra, Redis • Data lakes vs. data warehouses • Real-time stream processing: Apache Kafka, Apache Flink • **Applications:** Log analysis, clickstream processing, IoT data aggregation	1	25%
II	**Cloud Computing for AI** • Cloud computing fundamentals: IaaS, PaaS, SaaS, FaaS • Major cloud platforms: AWS, Azure, Google Cloud Platform • Cloud storage solutions: S3, Azure Blob, Google Cloud Storage	1	25%

	• Containerization and orchestration: Docker, Kubernetes for AI workloads • Serverless computing for AI: AWS Lambda, Azure Functions • Cloud-based ML platforms: SageMaker, Azure ML, Vertex AI • Cost optimization and resource management in cloud AI • **Applications:** Scalable model training, automated ML pipelines, batch inference		
III	**Distributed Machine Learning and MLOps** **Topics:** • Distributed training frameworks: Horovod, PyTorch DDP, TensorFlow Distributed • Model parallelism and data parallelism • Hyperparameter tuning at scale: Ray Tune, Optuna on cloud • Feature stores: Feast, Tecton • ML pipeline orchestration: Apache Airflow, Kubeflow Pipelines • Model versioning and registry: MLflow, DVC • Monitoring and logging for ML systems: Prometheus, Grafana • CI/CD for machine learning (MLOps) • **Applications:** Large language model training, recommendation systems at scale	1	25%
IV	**Advanced Analytics and Cloud AI Services** **Topics:** • Big data analytics tools: Apache Hive, Presto, Apache Druid • Data visualization at scale: Tableau, Power BI, Apache Superset • Managed AI services: Computer Vision APIs, NLP services, speech recognition • Vector databases and similarity search: Pinecone, Weaviate, Milvus • Graph analytics and processing: Neo4j, Amazon Neptune • Edge-cloud AI integration • Security, compliance, and governance in cloud AI • Ethical considerations: Bias in large-scale AI, data privacy • **Applications:** Real-time recommendation engines, fraud detection, predictive maintenance	1	25%

TEXT BOOKS:

- Chambers, B., & Zaharia, M. – *Spark: The Definitive Guide* – O'Reilly.

- Lakshmanan, V., Robinson, S., & Munn, M. – *Machine Learning Design Patterns* – O'Reilly.
- Kimball, R., & Ross, M. – *The Data Warehouse Toolkit* (3rd ed.) – Wiley.
- Huyen, C. – *Designing Machine Learning Systems* – O'Reilly.

REFERENCE BOOKS:

- White, T. – *Hadoop: The Definitive Guide* (4th ed.) – O'Reilly.
- Guller, M. – *Big Data Analytics with Spark* – Apress.
- Kleppmann, M. – *Designing Data-Intensive Applications* – O'Reilly.
- AWS/Azure/GCP – *Official certification guides and whitepapers*.

ONLINE RESOURCES:

- Databricks Community Edition: Free Spark cluster
- Google Colab Pro: GPU access for distributed ML experiments
- AWS Educate/Azure for Students: Free cloud credits
- Coursera: *Big Data Specialization* (University of California San Diego)
- edX: *Data Science and Machine Learning in the Cloud* (Microsoft)

PRACTICAL LIST:

1. **Big Data Processing with Apache Spark**

Task: Process a multi-terabyte dataset (e.g., Wikipedia dump or Twitter stream) using PySpark on Databricks. Perform data cleaning, aggregation, and analysis. Compare performance between RDD and Data Frame APIs.

2**. End-to-End Cloud ML Pipeline**

Task: Build a complete ML pipeline on AWS SageMaker or Azure ML including data ingestion, feature engineering, model training (XGBoost/Deep Learning), hyperparameter tuning, and deployment. Implement automated retraining with Airflow.

3, **Distributed Deep Learning Training**

Task: Train a large vision or language model (ResNet50 or BERT) using distributed training across multiple GPUs/instances on cloud. Implement with Horovod or PyTorch DDP. Compare training time and cost vs. single instance.

4. **Real-time Analytics and AI Serving**

Task: Create a real-time fraud detection system using Kafka for streaming, Spark Streaming/Flink for processing, and a pre-trained model served via TensorFlow Serving on Kubernetes. Build a monitoring dashboard with Grafana.