

HONOURS

S.No	Course Code	Course Title	Scheme of Instructions Hours per Week				Scheme of Examination Maximum Marks		
			L	T	P	C	I	E	Total
1	23HRCSM1	Advanced Algorithms For AI & ML	3	-	-	3	30	70	100
2	23HRCSM2	Deep Learning an Neural Network Architectures	3	-	-	3	30	70	100
3	23HRCSM3	Reinforcement Learning & Decision Making	3	-	-	3	30	70	100
4	23HRCSM4	AI For Robotics and Automation	3	-	-	3	30	70	100
5	23HRCSM5	AI Ethics, Fairness & Explainability	3	-	-	3	30	70	100
6	23HRCSM6	AI and Machine Learning Lab	-	-	3	1.5	30	70	100
7	23HRCSM7	Robotics and Autonomous Systems Lab	-	-	3	1.5	30	70	100

23HRCSM1	ADVANCED ALGORITHMS FOR AI & ML	L	T	P	C
		3	0	0	3

Course Objective:

- To deepen understanding of algorithmic principles for designing scalable and efficient AI/ML solutions.
- To explore advanced topics such as optimization algorithms, randomized and approximation algorithms, and online learning.
- To analyze computational complexity, tractability, and convergence of AI models.
- To apply graph-based, evolutionary, and heuristic approaches in solving real-world AI/ML problems.
- To integrate algorithmic strategies for large-scale machine learning, reinforcement learning, and neural network training.

Course Outcomes:

After successful completion of the course, students will be able to:

1. Analyze and apply classical algorithmic techniques including divide and conquer, dynamic programming, approximation, and randomized algorithms in the context of AI/ML.
2. Implement advanced graph algorithms for shortest paths, flows, and community detection, and apply them to AI problems like NLP and recommender systems.
3. Apply convex and non-convex optimization strategies, gradient-based learning, and regularization techniques to train and tune AI/ML models effectively.
4. Use evolutionary, swarm intelligence, and reinforcement learning-based metaheuristic methods for neural architecture search and complex optimization tasks in AI.
5. Evaluate and design scalable algorithmic solutions with fairness and interpretability for AI/ML applications, referencing case studies like AlphaGo, GPT, and AutoML systems.

UNIT I: Foundations of Advanced Algorithmic Techniques

Review of Time and Space Complexity, Divide and Conquer, Dynamic Programming, and Greedy Algorithms, Recurrence Relations and Master Theorem, Approximation Algorithms: Vertex Cover, TSP, Set Cover, Randomized Algorithms: Monte Carlo and Las Vegas Types, Probabilistic Analysis and Tail Bounds, Applications in ML Preprocessing and Feature Selection

UNIT II: Graph Algorithms and AI Applications

Graph Representations and Traversal Algorithms, Shortest Path: Dijkstra's, Bellman-Ford, Floyd-Warshall, Minimum Spanning Trees: Kruskal and Prim, Network Flows and Max Flow-Min Cut Theorem, Graph-Based Semi-Supervised Learning, PageRank, Centrality, and Community Detection, Applications in NLP, Vision, and Recommender Systems

UNIT III: Optimization in AI/ML

Convex and Non-Convex Optimization, Gradient Descent Variants: SGD, Momentum, Adam, Convergence Analysis and Learning Rates, Duality and Lagrange Multipliers, Regularization: L1, L2,

ElasticNet, Hyperparameter Optimization: Grid, Random, Bayesian, Constrained Optimization in SVMs and Deep Learning

UNIT IV: Evolutionary & Metaheuristic Algorithms

Genetic Algorithms and Evolutionary Strategies, Swarm Intelligence: PSO, Ant Colony Optimization, Simulated Annealing and Tabu Search, Multi-objective Optimization, Reinforcement Learning and Policy Gradient Methods, Neuroevolution: Evolving Neural Networks, Use Cases in Feature Engineering and Neural Architecture Search (NAS)

UNIT V: Advanced Topics and Case Studies

Online Learning and Regret Minimization, Bandit Algorithms: Multi-Armed Bandits, Thompson Sampling, Large-Scale Algorithms: MapReduce, Apache Spark MLlib, Algorithmic Fairness, Interpretability, and Ethics in AI, Case Studies: AlphaGo, GPT, BERT, Recommendation Engines, Research Trends in Algorithmic ML and AutoML, Capstone Problem Solving using Hybrid Algorithms

Text Books:

1. Introduction to Algorithms – Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein (MIT Press)
2. Algorithms for Machine Learning – Giuseppe Bonaccorso, Packt Publishing
3. Convex Optimization – Stephen Boyd and Lieven Vandenberghe, Cambridge University Press
4. Reinforcement Learning: An Introduction – Richard S. Sutton and Andrew G. Barto

Reference Books:

1. Machine Learning: A Probabilistic Perspective – Kevin P. Murphy
2. The Elements of Statistical Learning – Trevor Hastie, Robert Tibshirani, Jerome Friedman
3. Evolutionary Computation – Kenneth A. De Jong
4. Handbook of Approximation Algorithms and Metaheuristics – Teofilo F. Gonzalez

Online Courses:

1. Coursera – Advanced Algorithms and Complexity (UC San Diego)
2. edX – Algorithmic Design and Techniques (UC San Diego)
3. MIT OpenCourseWare – Advanced Algorithms
4. Udemy – Optimization Algorithms in Machine Learning
5. Stanford Online – Convex Optimization

23HRCSM2	DEEP LEARNING & NEURAL NETWORK ARCHITECTURES	L	T	P	C
		3	0	0	3

Course Objectives:

- To introduce the fundamental concepts and mathematical foundations of deep learning.
- To explore different neural network architectures including CNNs, RNNs, LSTMs, and Transformers.
- To enable students to implement, train, and optimize deep neural networks.
- To analyze the performance and limitations of various architectures in different AI tasks.
- To develop the ability to apply deep learning models to real-world applications such as image recognition, language modeling, and autonomous systems.

Course Outcomes (COs):

Upon successful completion of this course, the student will be able to:

- CO1: Understand the theoretical foundations of neural networks and deep learning.
- CO2: Implement and train multilayer perceptrons, CNNs, RNNs, and other architectures.
- CO3: Analyze and optimize deep learning models using advanced regularization and tuning techniques.
- CO4: Evaluate the applicability of different neural network architectures for various AI problems.
- CO5: Apply state-of-the-art models such as Transformers and GANs in real-world domains.

UNIT I: Foundations of Neural Networks

Introduction to Artificial Neural Networks, Biological Neuron vs. Artificial Neuron, Perceptron, Multilayer Perceptron (MLP), Activation Functions: ReLU, Sigmoid, Tanh, Softmax, Backpropagation and Gradient Descent, Loss Functions: MSE, Cross Entropy, Overfitting, Regularization (L1/L2), Dropout

UNIT II: Convolutional Neural Networks (CNNs)

Convolution Operation and Feature Maps, Pooling Layers: Max and Average Pooling, CNN Architectures: LeNet, AlexNet, VGG, ResNet, Transfer Learning and Fine-tuning, Image Classification, Object Detection Basics, Implementation with TensorFlow/PyTorch

UNIT III: Recurrent Neural Networks (RNNs) and Variants

Sequential Data and Time Series, RNN Basics and Backpropagation Through Time (BPTT), Vanishing and Exploding Gradients, LSTM and GRU Architectures, Applications in Text, Speech, and Music, Sequence-to-Sequence Models

UNIT IV: Advanced Architectures & Optimization

Autoencoders and Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), Deep Reinforcement Learning Overview, Batch Normalization, Early Stopping, Hyperparameter Tuning and Optimization, Performance Metrics and Evaluation

UNIT V: Transformer Models & Applications

Attention Mechanism and Self-Attention, Transformers and BERT Architecture, Positional Encoding, Multi-head Attention, Pre-trained Language Models and Fine-Tuning, Applications in NLP: Text Classification, Translation, Large Language Models and Transfer Learning

Text Books:

1. Deep Learning – Ian Goodfellow, Yoshua Bengio, and Aaron Courville (MIT Press)
2. Neural Networks and Deep Learning – Michael Nielsen (Online Book)
3. Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow – Aurélien Géron (O'Reilly)

Reference Books:

1. Pattern Recognition and Machine Learning – Christopher M. Bishop
2. Deep Learning for Computer Vision – Rajalingappaa Shanmugamani
3. Natural Language Processing with Transformers – Lewis Tunstall, Leandro von Werra, Thomas Wolf
4. Reinforcement Learning: An Introduction – Richard S. Sutton and Andrew G. Barto

Recommended Online Courses:

1. Deep Learning Specialization – Andrew Ng (Coursera)
2. CS231n: Convolutional Neural Networks for Visual Recognition (Stanford)
3. Fast.ai – Practical Deep Learning for Coders
4. Deep Learning with PyTorch (Udacity)
5. Transformers by Hugging Face (free course)

23HRCSM3	REINFORCEMENT LEARNING & DECISION MAKING	L	T	P	C
		3	0	0	3

Course Objectives:

- To introduce the fundamentals of reinforcement learning (RL) and its mathematical foundation.
- To understand the Markov Decision Process (MDP) framework for decision making under uncertainty.
- To explore various RL algorithms including value-based, policy-based, and model-based approaches.
- To analyze deep reinforcement learning techniques for real-world applications.
- To study the integration of reinforcement learning with planning, exploration, and control strategies.

Course Outcomes (COs):

After successful completion of this course, students will be able to:

1. Understand the fundamentals of reinforcement learning, including agent-environment interaction, types of RL, and solving decision-making problems using Markov Decision Processes and Bellman equations.
2. Apply dynamic programming and Monte Carlo methods to perform policy evaluation, policy improvement, and control in model-based RL settings.
3. Implement temporal-difference learning algorithms like TD(0), Sarsa, and Q-learning, and extend them using eligibility traces and function approximation techniques.
4. Develop and analyze policy gradient and actor-critic methods, including REINFORCE and PPO, to optimize policies in continuous and high-dimensional action spaces.
5. Employ deep reinforcement learning techniques (DQN, DDPG, A3C, SAC) and exploration strategies to solve complex tasks in robotics, games, and autonomous systems, considering safety and ethical decision-making.

UNIT I: Introduction to Reinforcement Learning & MDPs

Foundations of RL: Agent-Environment Interaction, Types of RL: Model-based vs. Model-free, Reward Signals, Return, and Discounting, Markov Decision Processes (MDPs), Bellman Equations and Optimality

UNIT II: Dynamic Programming & Monte Carlo Methods

Policy Evaluation and Policy Improvement, Value Iteration and Policy Iteration, Monte Carlo Prediction and Control, First-visit and Every-visit Methods, Limitations of DP and MC Approaches

UNIT III: Temporal-Difference Learning & Function Approximation

TD(0), Sarsa, and Q-Learning Algorithms, Eligibility Traces: TD(λ), Sarsa(λ), Off-policy vs. On-policy Learning, Linear Function Approximation, Generalization in RL

UNIT IV: Policy Gradient Methods and Actor-Critic Algorithms

Policy Gradient Theorem, REINFORCE Algorithm, Baselines and Variance Reduction, Actor-Critic Architectures, Trust Region and Proximal Policy Optimization (PPO)

UNIT V: Deep Reinforcement Learning and Applications

Deep Q-Networks (DQN) and Experience Replay, DDPG, A3C, and SAC Algorithms, Exploration Techniques: ϵ -greedy, UCB, Intrinsic Rewards, RL in Robotics, Game AI, and Autonomous Systems, Safety, Ethics, and Fairness in Decision Making

Textbooks:

1. **Richard S. Sutton and Andrew G. Barto** – *Reinforcement Learning: An Introduction*, 2nd Edition, MIT Press
2. **Ian Goodfellow, Yoshua Bengio, Aaron Courville** – *Deep Learning*, MIT Press

Reference Books:

1. **David Silver's RL Course Slides & Lectures** – DeepMind, University College London
2. **Marco Wiering & Martijn van Otterlo (Eds.)** – *Reinforcement Learning: State of the Art*, Springer
3. **Csaba Szepesvári** – *Algorithms for Reinforcement Learning*, Morgan & Claypool
4. **Yuxi Li** – *Deep Reinforcement Learning: An Overview*, arXiv survey

Online Courses & Resources:

1. **DeepMind x UCL Reinforcement Learning Lectures by David Silver**
2. **Coursera: Reinforcement Learning Specialization – University of Alberta**

23HRCSM4	AI FOR ROBOTICS & AUTOMATION	L	T	P	C
		3	0	0	3

Course Objectives:

- To introduce the principles of software engineering augmented with artificial intelligence.
- To explore AI-driven tools for software requirement analysis, design, testing, and maintenance.
- To understand DevOps practices and integrate AI techniques for CI/CD, deployment, and monitoring.
- To automate software lifecycle management using machine learning and data-driven insights.
- To develop intelligent pipelines for software delivery with adaptive testing and feedback systems.

Course Outcomes (COs):

By the end of the course, students will be able to:

- CO1: Describe the role of AI in modern software engineering processes and lifecycle stages.
- CO2: Apply AI/ML models for requirements gathering, code generation, defect prediction, and testing.
- CO3: Implement intelligent DevOps practices including CI/CD, release automation, and anomaly detection.
- CO4: Analyze data from software pipelines to drive informed decisions and improve quality.
- CO5: Develop an end-to-end AI-enabled software delivery pipeline with automated learning-based optimizations.

UNIT I: Foundations of AI-Driven Software Engineering

Introduction to Software Engineering Lifecycle, Traditional vs. AI-Driven Software Development, AI/ML in Software Engineering: Overview and Scope, Natural Language Processing (NLP) for Requirements Engineering, AI for Software Design Recommendation, Intelligent Code Completion (e.g., GitHub Copilot)

UNIT II: AI in Testing and Defect Prediction

Static and Dynamic Testing with AI, Automated Test Case Generation, Defect Detection and Prediction using ML Models, Sentiment and Bug Report Analysis, AI in Refactoring and Code Review, Tools: SonarQube, DeepCode

UNIT III: DevOps Principles and Practices

DevOps Overview: CI/CD Pipelines, Infrastructure as Code (IaC), Configuration Management Tools: Ansible, Puppet, Monitoring and Logging Tools: Prometheus, Grafana, Containerization and Orchestration: Docker, Kubernetes, Agile and Lean Practices in DevOps

UNIT IV: AI for DevOps Automation and Intelligence

Predictive Analytics for Deployment Success, AI for Log Analytics and Root Cause Analysis, Self-Healing Systems and Auto-Scaling, Feedback Loops in DevOps using Reinforcement Learning, Data-Driven Decision Making in Release Management, ChatOps and AIOps Platforms

UNIT V: Case Studies and Emerging Trends

Case Study: AI-Augmented DevOps in Enterprises, ML-Ops vs. DevOps vs. DataOps, Security in DevOps (DevSecOps), Explainability and Ethics in AI-Driven Software Engineering, Generative AI in Software Development, Future Trends and Industry Standards

Textbooks:

1. Tim Menzies, Diomidis Spinellis – Artificial Intelligence and Software Engineering: Status and Future Directions
2. Len Bass, Ingo Weber, Liming Zhu – DevOps: A Software Architect's Perspective, Addison-Wesley
3. Thomas Erl, Ricardo Puttini, Zaigham Mahmood – AI & Analytics for DevOps, Pearson

Reference Books:

1. Carlos Nunes Silva – AI in Software Engineering
2. Gene Kim, Jez Humble, Patrick Debois, John Willis – The DevOps Handbook
3. Andrew Ng – Machine Learning B.Techning (AI Systems Engineering Perspective)

Online Resources & Courses:

1. Coursera – AI for Software Engineering (IBM)
2. DevOps with Microsoft Azure – edX
3. Udacity – AI for DevOps Engineers Nanodegree

23HRC5M5	AI ETHICS, FAIRNESS & EXPLAINABILITY	L	T	P	C
		3	0	0	3

Course Objectives:

- To understand ethical concerns and responsibilities in the development and deployment of AI systems.
- To study fairness, bias, and accountability issues in AI and machine learning models.
- To explore techniques and frameworks for interpreting and explaining AI decisions.
- To analyze societal impacts of AI and build trust through transparent systems.
- To promote responsible and inclusive AI development aligned with human values.

Course Outcomes (COs):

After successful completion of the course, students will be able to:

1. Describe the ethical principles, historical context, and responsibilities associated with AI deployment across domains like healthcare and law enforcement.
2. Identify different forms of bias in datasets and algorithms, and apply fairness metrics and mitigation strategies to ensure equitable AI systems.
3. Demonstrate the need for explainability in AI models and utilize tools such as LIME, SHAP, and Grad-CAM to generate local and global model explanations.
4. Design AI systems with accountability by integrating human oversight, ethical documentation (e.g., Model Cards, Datasheets), and adherence to global guidelines.
5. Critically assess the broader societal and legal implications of AI in areas such as surveillance, misinformation, and inclusivity, and explore international policy frameworks.

UNIT I: Foundations of AI Ethics

Historical background of AI ethics, Core principles: beneficence, non-maleficence, autonomy, justice, Moral and legal responsibilities in AI systems, Risk assessment and governance in AI, Ethical AI case studies from healthcare, policing, hiring

UNIT II: Fairness and Bias in AI

Types of bias: dataset bias, label bias, historical bias, Fairness definitions: demographic parity, equal opportunity, individual fairness, Disparate impact and fairness metrics, Algorithmic audits and bias detection, Fairness-aware learning and mitigation strategies

UNIT III: Explainable Artificial Intelligence (XAI)

Need for interpretability in AI models, Taxonomy of XAI methods: model-agnostic, model-specific, LIME, SHAP, Grad-CAM, Partial Dependence Plots, Local vs Global explanations, Trade-offs: accuracy vs interpretability

UNIT IV: Accountability and Responsible AI Design

Transparent AI systems, Human-in-the-loop and AI-assisted decision-making, Accountability frameworks (e.g., IEEE, NIST, EU Guidelines), Documentation tools: Datasheets for datasets, Model Cards, Responsible AI lifecycle management

UNIT V: Societal Impacts and Policy Considerations

AI in surveillance, misinformation, and social manipulation, Ethical implications in autonomous systems (vehicles, weapons), AI and inclusion: accessibility, gender, race, socioeconomic impacts, Public policy, legal frameworks, and global initiatives, Future challenges and global governance of AI

Textbooks:

1. **Virginia Dignum** – *Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way*, Springer
2. **Cathy O’Neil** – *Weapons of Math Destruction*, Crown Publishing
3. **Mark Coeckelbergh** – *AI Ethics*, The MIT Press

Reference Books:

1. **Nick Bostrom & Eliezer Yudkowsky** – *The Ethics of Artificial Intelligence*
2. **Shalini Kantayya (Film)** – *Coded Bias* (documentary, 2020)
3. **Floridi, Luciano** – *Ethics of Information*, Oxford University Press

Online Courses & Resources:

1. **Coursera** – **AI For Everyone (Andrew Ng)**
2. **edX** – **Ethics of AI and Big Data (Linux Foundation)**

23HRCSM6	AI & MACHINE LEARNING LAB	L	T	P	C
		0	0	3	1.5

Course Objectives

1. To provide hands-on experience in implementing AI and machine learning algorithms.
2. To develop and evaluate models using real-world datasets.
3. To introduce optimization and hyperparameter tuning techniques.
4. To build intelligent systems for classification, prediction, and clustering.

Course Outcomes (CO)

After completing this lab, students will be able to:

CO1: Implement key machine learning algorithms from scratch and using libraries.

CO2: Preprocess data and select suitable features for modeling.

CO3: Train, test, and evaluate models for accuracy and performance.

CO4: Apply AI techniques to solve classification, regression, and decision-making problems.

CO5: Develop simple AI agents and use neural networks for predictive tasks.

Tools Required

- Python (NumPy, Pandas, Scikit-learn, TensorFlow/Keras, OpenCV)
- Jupyter Notebook / Google Colab
- Datasets from UCI, Kaggle, Scikit-learn
- Anaconda / VS Code

List of 12 Experiments

1. **Data Preprocessing** – Cleaning, normalization, encoding, and splitting data.
2. **Linear Regression** – Implement simple and multiple linear regression.
3. **Logistic Regression** – Binary classification on datasets like breast cancer or Titanic.
4. **K-Nearest Neighbors (KNN)** – Classification task with evaluation metrics.
5. **Decision Trees and Random Forests** – Tree-based classification and visualization.
6. **Support Vector Machines (SVM)** – Margin classification with kernel trick.
7. **Naïve Bayes** – Text classification with spam dataset.
8. **K-Means Clustering** – Unsupervised clustering with customer segmentation.
9. **Principal Component Analysis (PCA)** – Dimensionality reduction and visualization.
10. **Artificial Neural Networks (ANNs)** – Implement basic neural network using Keras.
11. **Model Evaluation & Tuning** – Use cross-validation, GridSearchCV, and confusion matrices.
12. **AI Agent Search Algorithms** – Implement A*, DFS, BFS for pathfinding problems.

23HRCSM7	ROBOTICS & AUTONOMOUS SYSTEMS LAB	L	T	P	C
		0	0	3	1.5

Course Objectives:

- To provide hands-on experience in deploying machine learning models into production environments.
- To introduce the tools and practices of MLOps for automating ML workflows.
- To train students in containerization, orchestration, monitoring, and CI/CD pipelines.
- To understand model versioning, reproducibility, and lifecycle management.
- To develop skills in using cloud platforms and APIs for scalable AI applications.

Course Outcomes:

By the end of this course, students will be able to:

- Package and deploy AI models using tools such as Flask, FastAPI, and Docker.
- Automate machine learning workflows using CI/CD pipelines and MLOps tools.
- Monitor and manage deployed models in real-time environments.
- Apply version control and model registry techniques effectively.
- Deploy models on cloud platforms like AWS, Azure, or GCP and use MLflow, Kubeflow, etc.

List of 12 Lab Experiments:

1. Experiment 1: Build a simple ML model and serve it via Flask or FastAPI.
2. Experiment 2: Containerize the model application using Docker.
3. Experiment 3: Deploy a Dockerized model on a local or cloud-based Kubernetes cluster.
4. Experiment 4: Implement CI/CD pipeline using GitHub Actions or GitLab CI.
5. Experiment 5: Track experiments and manage model versions using MLflow.
6. Experiment 6: Use DVC (Data Version Control) for tracking data and pipeline stages.
7. Experiment 7: Automate model retraining and deployment using Jenkins.
8. Experiment 8: Model monitoring using Prometheus and Grafana.
9. Experiment 9: Introduce model drift detection and retraining triggers.
10. Experiment 10: Deploy a model on a cloud platform (e.g., AWS SageMaker, GCP AI Platform).
11. Experiment 11: Use Kubeflow pipelines for end-to-end ML workflow management.
12. Experiment 12: Capstone: Full-cycle ML project from training to monitoring using MLOps best practices.

Textbooks:

1. Mark Treveil and Alok Shukla, AI and Analytics in Production: How to Implement Successful AI and Analytics Applications, O'Reilly Media.
2. Emmanuel Ameisen, Building Machine Learning Powered Applications, O'Reilly Media.

Reference Books:

1. Chris Fregly and Antje Barth, Data Science on AWS: Building End-to-End Applications, O'Reilly.
2. Alfredo Deza and Noah Gift, Practical MLOps, O'Reilly Media.
3. Soham Kamani, Learning MLOps, Packt Publishing.