BODOLAND UNIVERSITY

Curriculum Structures for 2 Years Syllabus for
Master of Science in Artificial Intelligence and Machine Learning

Year 2025

PROGRAMME STRUCTURE AND DETAILED CURRICULUM

Bodoland University

DEBARGAON, KOKRAJHAR (B.T.R.)

Curriculum Structures for NEP 2 Years Syllabus for M.Sc in AL & ML
No. of Papers= 20, Total Credits= 80, Total Marks =2000

SEMESTER-I						
Paper code	**Subject Name**	**Credit**	**L+T+P**	**Formative**	**Summative**	**Total Marks**
AIMLADL14014	Mathematical Foundation for AI/ML	4	3+1+0	30	70	100
AIMLADL14024	Fundamentals of Artificial Intelligence	4	3+1+0	30	70	100
AIMLADL14034	Introduction to Machine Learning	4	3+1+0	30	70	100
AIMLADL14042	Indian Knowledge System inAI/ML	2	2+0+0	--	50	50
AIMLADL14052	Introduction to Cyber Security	2	2+0+0	--	50	50
AIMLADL14064	Python Programming for AI/ML	4	3+1+0	30	70	100
Total		**20**				500

SEMESTER-II						
Paper code	**Subject Name**	**Credit**	**L+T+P**	**Formative**	**Summative**	**Total Marks**
AIMLSPL14074	Deep Neural Networks	4	3+1+0	30	70	100
AIMLSPL14084	Introduction to Data Science	4	3+1+0	30	70	100
AIMLSPL14094	Elective I (Swayam)	4	4+0+0	00	100	100
AIMLSPL140104	Natural Language Processing	4	3+1+0	30	70	100
AIMLSPL140114	Computational Intelligence Laboratory-I	4	0+1+3	30	70	100
Total		**20**				500

In M.Sc in AI & ML 3rd Semester, learners will choose either Group-A or Group-B or Group C

Option A- only Coursework

SMESTER-III						
Paper code	**Subject Name**	**Credit**	**L+T+P**	**Formative**	**Summative**	**Total Marks**
AIMLSPL25014	Generative AI	4	3+1+0	30	70	100
AIMLSPL25024	Computer Vision & Image Processing	4	3+1+0	30	70	100
AIMLSPL25034	Elective II (Swayam)	4	4+0+0	00	100	100
AIMLSPL25042	Internship	2	0+0+2	--	50	100
AIMLSEM25052	Seminar	2	0+0+2	--	50	100
AIMLSPL25064	Computational Intelligence Laboratory-II	4	0+1+3	30	70	100
		20				500

SEMESTER-IV						
Paper code	**Subject Name**	**Credit**	**L+T+P**	**Formative**	**Summative**	**Total Marks**
AIMLSPL25074	Reinforcement Learning	4	4+0+0	30	70	100
AIMLDIS250816	Dissertation	16		120	280	400
Total		20				500

Guidelines for Seminar:

1. A seminar should be given by an individual students, based on topics chosen from the emerging area and technology of Computer Science and Computer Applications.
2. References from journal such as IEEE, ACM etc… shall be used.
3. A report on this seminar with15-20 pages should be prepared.

Option B- only Research

Research thesis/project with minimum 2 conferences papers. Peer reviewed research publication should be encouraged

AIMLDIS250120: RESEARCH THESIS/ DISSERTATION-I
(Credit 20, Total Marks=500) (Practical = 500 Marks)

Option C- Coursework+ Research

SMESTER-III						
Paper code	**Subject Name**	**Credit**	**L+T+P**	**Formative**	**Summative**	**Total Marks**
AIMLSPL25014	Generative AI	4	3+1+0	30	70	100
AIMLSPL25024	Computer Vision & Image Processing	4	3+1+0	30	70	100
AIMLSPL25034	Elective II (Swayam)	4	4+0+0	00	100	100
AIMLSPL25042	Internship	2	0+0+2	--	50	100
AIMLSEM25052	Seminar	2	0+0+2	--	50	100
AIMLSPL25064	Computational Intelligence Laboratory-II	4	0+1+3	30	70	100
		20				500

SEMESTER-IV

Research thesis/project with minimum 1 conferences papers. Peer reviewed research publication should be encouraged.

Eligibility Criteria for MSc AI & ML Admission

PROGRAMME LEARNING OUTCOME

MASTER OF SCIENCE IN AI &ML (M.Sc. AL/ML)

After completion of the M.Sc Program, learners would be able to:

1. **Computational Knowledge:** Utilize understanding of basic computing principles, specialized computing knowledge, mathematics, and relevant domain expertise to abstract and conceptualize computing models based on specified problems and requirement.
2. **Problem Analysis :** Identify, formulate, review literature, and resolve complex computing issues by drawing well-supported conclusions using foundational principles of mathematics, computer science, and relevant domain fields.
3. **Design /Development of Solutions:**Create and assess solutions for intricate computing challenges, and develop and appraise systems, components, or processes that fulfill defined requirements, considering public health and safety, as well as cultural, social, and environmental factors.
4. **Conduct Investigations of Complex Computing Problems:**Utilize research-based expertise and methodologies, including experimental design, data analysis, and interpretation, and information synthesis to draw valid conclusions.
5. **Modern Tool Usage:**Develop, choose, modify, and use suitable techniques, resources, and contemporary computing tools for complex computing tasks, while recognizing their limitations.

6. **Professional Ethics:**Comprehend and adhere to professional ethics, cyber regulations, responsibilities, and standards of professional computing practice.
7. **Life-long Learning:**Acknowledge the necessity and possess the ability to engage in self-directed learning for ongoing growth as a computing professional.
8. **Project management and finance:**Exhibit knowledge and comprehension of computing and management principles, applying them to one's work as both a team member and leader, to manage projects and operate in multidisciplinary settings.
9. **Communication Efficacy:**Effectively communicate with the computing community and society at large about complex computing tasks by being able to understand and write clear reports, create detailed documentation, deliver impactful presentations, and give and follow precise instructions.
10. **Societal and Environmental Concern:**Comprehend and evaluate societal, environmental, health, safety, legal, and cultural concerns within both local and global contexts, along with the associated responsibilities pertinent to professional computing practice.
11. **Individual and Teamwork:**Operate proficiently both independently and as part of diverse teams, assuming roles as either a member or a leader, within varied multidisciplinary environments.
12. **Innovation and Entrepreneurship:**Recognize a current opportunity and utilize innovation to seize that opportunity, generating value and prosperity for both the individual and society as a whole.

DETAILED SYLLABUS
SEMESTER-I
AIMLADL14014: MATHEMATICAL FOUNDATION OF ARTIFICIAL INTELLIGENCE/MACHINE LEARNING
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objectives:
- To acquaint the students with mathematical/logical fundamentals including numerical techniques,
- To understand probability and graph theory that serve as an essential tool for applications of computer and information sciences.

Course Learning Outcome: Completion of this course, students should be able to do the following:
- Develop mathematical thinking and problem solving skills associated with research and writing proof
- Get ideas about sets and perform operations and algebra on set. Determine different properties of relations, identify equivalence and partial order relations. Identify function and their properties. Get idea about Group Theory.
- Get exposure to a wide variety of mathematical concepts used in computer science discipline like probability.
- Analyze the given propositions and finding results using mathematical logic operators.
- Understand and apply the concepts of graphs and trees. Analyze whether given graphs are isomorphic and different algorithms to find the shortest path.

Unit 1: Set and logic

- Set, subset, power set, properties of set, operations on sets, products on set, representation of set in computer partitions.
- Prepositions, Logical Operations, Tautologies, Negation, Contradiction, Contraposition, Logical Implication, Logical Equivalence, Normal Forms, Theory of Inference and Deduction, Predicates and Quantifiers.

Unit 2: Relation, function, Algebra and Lattice

- Relation and Function: Binary relation, types of relation, equivalence relation and equivalence class, closure of relation, function, different types of functions, recurrence relation, First order linear recurrence relation, The second order linear homogeneous recurrence relation with constant coefficients.
- Algebraic Structures: Introduction to algebraic Structures and properties. Types of algebraic structures: Semi group, Monoid, Group, Abelian group and Properties of group. Subgroup.
- Lattices: Introduction, Partial ordered sets, Combination of Partial ordered sets, Hasse diagram, Introduction of lattices, Properties of lattices – Bounded, Complemented, Modular and Complete lattice.

Unit-3: probability and Statistics

- Basic probability- Addition and Multiplication theorem-problems, Conditional probability.
- Random variables – Discrete random variables, Joint distribution, Binomial and Poisson Distributions, Measures of central tendency- mean, median, mode for grouped data, Measures of dispersion, Mean deviation and Standard deviation,

Unit 4: Linear Programming:

Mathematical model, assumptions of linear programming, Principles of simplex method, Revised simplex method, Applications, Duality, Dual Simplex method, sensitivity analysis

Special type of linear programming problems: Transportation and assignment problems.

Integer programming: introduction, Branch and bound techniques, Assignment and travelling salesman problems with algorithmic approach.

Dynamic programming: deterministic and probabilistic dynamic programming

SUGGESTED READING:

1. S Santha, Discrete mathematics, Cengage Learning
2. Veerarajan, Discrete Mathematics, TMGH
3. Liu and Mahapatra, Discrete Mathematics, TMH
4. Rosen : Discrete Mathematics, 6/e, TMH Kolman : Discrete Mathematics, PHI.
5. J.P. Trembely&R.Manohar, "Discrete Mathematical Structure with application to Computer Science", McGraw Hill.
6. Natarajan, Balasubramani, Tamilasari : Operations Research , Pearson
7. Gillett, B.G.: Introduction to operations research - A Computer oriented Algorithmic approach, McGraw-Hill.

AIMLADL14024: FUNDAMENTALS OF ARTIFICIAL INTELLIGENCE
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objective:

- To understand the various types and functions of Operating Systems
- To learn the process and various scheduling algorithms
- To explore various issues in inter-process communications and address them
- To evaluate the Memory management algorithms, allocation methods and virtual memory implementations
- To analyze file and I/O management in Operating System
- To learn shell scripting fundamentals

Course Learning Outcomes: Completion of this course, students should be able to do the following:

- Understand fundamental operating system architectures, processes, threads, files, semaphores, Inter Process Communication, shared memory regions, etc.
- Analyze algorithms like Process scheduling and memory management algorithms.
- Categorize the operating system's resource management techniques, deadlock management techniques.
- Able to understand paging of memory, page replacement algorithms, file systems.
- Understand the working of secondary storage devices.

Unit-1: Introduction to Artificial Intelligence

- Goals and Objectives: Introduction to AI, its history, goals, and objectives. Overview of AI's impact on various domains (e.g., healthcare, robotics, finance).
- Problem Space and State Representation: Problem characteristics, state space representation, and examples (e.g., tic-tac-toe, 8-puzzle).Problem-solving as search in AI, introduction to state-space graphs.
- Applications of AI: Brief mention of practical applications in modern industries (e.g., recommendation systems, autonomous vehicles).

Unit-2: Problem solving techniques:

Search Techniques:

- Generate-and-test, Hill climbing, search problem reduction.
- Constraint propagation, Means-end analysis.
- Heuristic search techniques: Greedy search, A* search.
- Case-based reasoning: Introduction and examples.

Game Playing:

- Graph search and game trees.
- Minimax and alpha-beta pruning algorithms.
- Applications in modern AI-driven games (e.g., Chess, Go, video games).

Unit-3: Knowledge representation and inference:

Knowledge Representation:

- Procedural and declarative approaches.
- Production systems and predicate logic (first-order and second-order logic).
- Rule-based systems: Semantics nets, conceptual dependencies, and frames.
- Ontologies: Concepts and applications in organizing knowledge.

Inference Techniques:

- Resolution in predicate logic, Unification, and natural deduction.
- Forward and backward reasoning.
- Reasoning under uncertainty: Probabilistic reasoning (Bayesian networks), fuzzy logic, monotonic and non-monotonic reasoning.
- Constraint satisfaction problems: Examples and applications.

Rule-Based Systems and Expertise:

- Meta-languages, transfer of expertise, and self-explaining systems.

Unit-4: Introduction to neural network:

Artificial Neurons: Definition and analogy with biological neurons, structure of artificial neurons, and basic activation functions (e.g., sigmoid, ReLU).

Neural Network Architectures:

- Introduction to multilayer neural networks.
- Concepts of input, hidden, and output layers.
- Overview of tree-layer neural networks.

Learning Algorithms:

- Learning by backpropagation: Gradient descent and loss functions.
- Overfitting and regularization techniques (dropout, weight decay).

SUGGESTED READING :

1. Nilson N J, Principles of Artificial Intelligence. Springer
2. Rich and Night,m Artificial intelligence, MGH.
3. "Artificial Intelligence and Machine Learning" by Chandra S.S. and Vinod Chandra S.S.
4. "Introduction to Artificial Intelligence and Expert Systems" by Dan W. Patterson.
5. "Artificial Intelligence" by Deepak Khemani.

6. "Artificial Intelligence and Neural Networks" by Rajasekaran and Pai.
7. "Soft Computing and Artificial Intelligence" by A. S. Pant and A. J. Raju.

AIMLADL14034: ITRODUCTION TO MACHINE LEARNING
(Credit 4, Total Marks=100)
(Final = 70 Marks, Internal = 30 Marks)

Course Learning Objectives:

- To understand the basic theory underlying machine learning.
- To be able to formulate machine learning problems corresponding to different applications.
- To understand a range of machine learning algorithms along with their strengths and weaknesses.
- To be able to apply machine learning algorithms to solve problems of moderate complexity.
- To apply the algorithms to a real-world problem, optimize the models learned and report on the expected accuracy that can be achieved by applying the models.

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Understand the basics of machine learning: Programming vs. machine learning, applications, types, validation, and evaluation metrics
- Learn feature selection: Importance, filter, wrapper, hybrid methods.
- Gain knowledge of regression: Linear, non-linear, logistic, Poisson regression.
- Explore support vector machines (SVM): SVM classifiers, kernels, R/Python implementation.
- Familiarize with Bayesian classification: Bayes theorem, classifier, naïve Bayes.
- Acquire neural network skills: Perceptron, feed-forward/back-propagation, recurrent, deep networks, R/Python implementation.
- Learn clustering: Partitional, hierarchical, density-based, validation, R/Python implementation.

Unit-1: Foundations of Machine Learning

- Introduction to Machine Learning (ML):
    - Definition, history, and applications of ML.
    - Differences between supervised, unsupervised, and reinforcement learning.
    - Overview of ML workflow: Data collection, preprocessing, model training, and evaluation.
- Mathematical Foundations:
    - Linear algebra basics: Vectors, matrices, and operations.
    - Probability and statistics: Random variables, distributions, mean, variance, and Bayes' theorem.
    - Optimization concepts: Gradient descent and cost functions.
- Programming Tools for ML:
    - Introduction to Python libraries: NumPy, pandas, scikit-learn, and matplotlib.

Unit-2: Supervised Learning

- Regression Analysis:
    - Linear regression and multiple regression.
    - Loss functions: Mean squared error (MSE).
    - Regularization techniques: Ridge and Lasso regression.
- Classification Models:
    - Logistic regression, k-Nearest Neighbors (k-NN).
    - Decision Trees and Random Forests.
    - Support Vector Machines (SVM).
- Model Evaluation and Validation:
    - Metrics: Accuracy, precision, recall, F1-score, and ROC-AUC.
    - Cross-validation techniques.
    - Bias-variance tradeoff.

Unit-3: Unsupervised Learning and Feature Engineering

- Clustering:
    - K-means clustering, hierarchical clustering, and DBSCAN.
    - Applications: Image segmentation, customer segmentation.
- Dimensionality Reduction:
    - Principal Component Analysis (PCA) and t-SNE.
    - Applications: Data visualization, feature selection.
- Feature Engineering:
    - Importance of feature scaling (normalization and standardization).
    - Handling missing data and categorical variables.
    - Feature extraction and importance ranking.

Unit-4: Neural Networks and Advanced Topics

- Introduction to Neural Networks:
    - Perceptrons, activation functions, and multilayer perceptrons.
    - Training neural networks: Backpropagation and optimization.
- Overview of Deep Learning:
    - Convolutional Neural Networks (CNNs) for image data.
    - Recurrent Neural Networks (RNNs) for sequential data.
- Ethics in Machine Learning:
    - Responsible AI: Fairness, accountability, and transparency.

- Bias in ML algorithms and ethical implications.

SUGGESTED READINGS:

1. Tom M. Mitchell (2017), Machine Learning, Me Grew Hills.
2. Ng. A. (2017). Machine learning yearning. (http://www.mlyearning. Org).
3. J Kalita, DK Bhattacharyya, S Roy (2023), Fundamentals of Data Science- Theory & Practice, Elsevier. S. J. Russell, P. Norvig. Artificial Intelligence: A Modern Approach. Third Edition, Prentice-Hall, 2010.
4. C. M. Bishop (2016), Pattern Recognition and Machine Learning, Springer.
5. Marco Gori , Machine Learning: A Constraint-Based Approach, Morgan Kaufmann. 2017
6. Ethem Alpaydin, Machine Learning: The New AI, MIT Press-2016
7. "Introduction to Machine Learning with Python: A Guide for Data Scientists" by Andreas C. Müller and Sarah Guido.
8. "Google Machine Learning Crash Course": A free, interactive introduction to ML by Google(online Resources)
9. Andrew Ng's Machine Learning Course (Coursera): A foundational course covering ML basics with mathematical rigor.
10. "Python Machine Learning" by Sebastian Raschka and Vahid Mirjalili

AIMLADL14042: INDIAN KNOWLEDGE SYSTEM IN ARTIFICIAL INTELLIGENCE/ MACHINE LEARNING

(Credit 2, Total Marks=50)
(Final = 50 Marks)

Course Learning Objectives

- To understand the Indian Knowledge System among scholars, academicians, and other interested individuals
- For discussions, intellectual exchanges, and scholarly pursuits related to the Indian knowledge system.
- Exploring how Indian knowledge frameworks can be applied to today's social and academic settings.

Course Learning Outcomes

- Learn about Indian ancient contributions to Computer Science.
- Learn about contribution of Computer Scientists and Technologists.
- Revisit evolution of Indian Computers.
- Learn about to Information Technology Revolutions.
- Learn about Future perspectives of Computer Science in India.

Unit 1: Foundations of Indian Knowledge Systems and AI/ML

- Introduction to IKS: Overview and relevance to modern computational sciences.
- Logical Foundations:
  - Nyaya Sutras and Indian logic as precursors to computational reasoning.

    - Comparative study: Nyaya logic vs. propositional logic.
- Mathematical Heritage:
    - Decimal system, zero, and infinity.
    - Contributions of Aryabhata and Brahmagupta to algorithms and calculations.
- Relevance to AI/ML: How ancient methodologies influence data science and decision-making models

Unit 2: Indian Linguistics and Natural Language Processing (NLP)

- Panini's Ashtadhyayi:
    - Computational grammar of Sanskrit and its linguistic precision.
    - Relevance to modern NLP frameworks (e.g., tokenization, parsing).
- Linguistic Models in AI:
    - Sandhi and Samasa for text preprocessing.
    - Sanskrit as a context-free grammar for building NLP models.
- Applications: Case studies on language modeling and translation using Sanskrit grammar.

Unit-3: Knowledge Representation and Ethical AI

- Knowledge Representation in Ancient Texts:
    - Taxonomies and ontologies in Indian philosophy (e.g., Vaisheshika categories).
    - Representation of medical knowledge in Ayurveda and its parallels in modern data organization.
- Ethical AI through IKS:
    - Indian philosophical perspectives on ethics (e.g., Dharma, interconnectedness).
    - Frameworks for bias reduction and fair AI inspired by Advaita Vedanta.
- Case Studies:
    - Applying IKS principles to ethical dilemmas in AI.

Unit-4: Applications and Research in AI/ML using IKS

- Pattern Recognition in IKS:
    - Ragas and Indian classical music: Applying AI for pattern detection.
    - Ancient agricultural prediction methods and machine learning integration.
- IKS-Inspired Machine Learning Models:
    - Optimization techniques from Indian astronomy and algorithms.
    - Prosody and rhythmic patterns (Chandas) for computational applications.

SUGGESTED READINGS:

1. "The Nyaya Sutras of Gautama" by Gautama.
2. "The Vedic Mathematics Sutras" by Jagadguru Swami Sri Bharati Krishna Tirthaji Maharaja.
3. "Ashtadhyayi of Panini" Translation and Commentary by: S.C. Vasu.
4. "Artificial Intelligence and Indian Philosophy" by V.S. Ramachandran (available as papers and articles in journals).
5. "Computational Sanskrit and AI" by Rishi Rajpopat.
6. "Ethics in AI: Ancient Indian Perspectives" by Edited volume by scholars in AI ethics and Indian philosophy.

AIMLADL14052: INTRODUCTION TO CYBER SECURITY
(Credit 2, Total Marks=50)
(Theory: 50 Marks)

Course Learning Objectives:

- To comprehend the importance of cybersecurity in protecting digital assets.
- To explain the CIA triad and identify common types of cyber threats and vulnerabilities.
- To recognize the role of firewalls, IDS, and VPNs in safeguarding networks.
- To apply basic endpoint and system security measures to mitigate threats.
- To understand the ethical implications of cybersecurity practices.
- To identify best practices in ensuring secure communication and data protection.

Course Learning Outcomes:

- Define cybersecurity principles and the CIA triad, and identify common types of threats, vulnerabilities, and risks.
- Apply symmetric and asymmetric cryptographic methods (e.g., AES, RSA) and utilize hashing (e.g., SHA) and digital signatures for secure data handling.
- Outline the steps for detecting, containing, and recovering from cyber incidents.
- Analyze case studies to understand common types of cyberattacks and their impact.
- Identify security challenges in IoT and explore how AI and machine learning enhance cybersecurity measures.
- Demonstrate an understanding of ethical considerations and best practices in protecting digital

assets and user privacy.

Unit 1: Introduction to Cyber Security

- Basics of Cyber Security:
  - Definition, objectives, and importance.
  - The CIA triad (Confidentiality, Integrity, Availability).
  - Types of cyber threats: Malware, phishing, and DoS attacks.
- Cybersecurity Landscape:
  - Common vulnerabilities and risks.
  - Brief overview of cybersecurity frameworks (ISO 27001, NIST).

Unit 2: Cryptography Essentials

- Introduction to Cryptography:
  - Symmetric key cryptography: DES, AES.
  - Asymmetric key cryptography: RSA.
- Hashing and Digital Signatures:
  - Basics of hashing (MD5, SHA).
  - Use of digital signatures in security.
- Applications:
  - SSL/TLS for secure communication.

Unit 3: Network and System Security

- Network Security Fundamentals:
  - Firewalls and intrusion detection systems (IDS).
  - Basics of Virtual Private Networks (VPNs).
- System Security:
  - Password policies and authentication mechanisms.
  - Best practices for endpoint security (antivirus, patches).

Unit 4: Cybersecurity in Practice

- Incident Response Basics:
  - Steps in handling cyber incidents: Detection, containment, recovery.
  - Case studies: Common types of data breaches.
- Emerging Threats:
  - Overview of IoT security challenges.
  - Brief introduction to AI and machine learning in cybersecurity.

SUGGESTED READINGS:

1. "Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives" by Nina Godbole and Sunit Belapure.
2. "Cryptography and Network Security: Principles and Practice" by William Stallings.
3. "Network Security Essentials: Applications and Standards" by William Stallings.
4. "Cybersecurity: Attack and Defense Strategies" by Yuri Diogenes and Erdal Ozkaya.

AIMLADL14064: PYTHON PROHRAMMING FOR AI/ML
(Credit 4, Total Marks=100)
(Final = 70 Marks, Internal = 30 Marks)

Course Learning Objectives:

- Provide foundational knowledge of Object-Oriented Programming (OOP) principles and their practical applications.
- Develop Python programming skills for AI-related tasks.
- Explore Python libraries and tools for machine learning, data analysis, and AI.
- Build real-world problem-solving skills through hands-on projects.

Course Learning Outcome:

- Differentiate between procedural and object-oriented programming paradigms.
- Use object-oriented techniques to design and implement real-world applications.
- Use Python for advanced data handling, including file operations and data processing with libraries like NumPy and Pandas.
- Develop machine learning models using Python and libraries such as scikit-learn.
- Gain proficiency in Python libraries such as NumPy, Pandas, Matplotlib, TensorFlow, and PyTorch.
- Analyze real-world problems and identify object-oriented and AI-based solutions.

Unit 1: Fundamentals of Programming and OOP Concepts

- **Introduction to Programming:**

- Overview of programming paradigms: Procedural, Functional, and Object-Oriented Programming.
- Python as a versatile programming language for OOP and AI.

- **Core OOP Concepts:**
  - Principles of OOP: Encapsulation, Abstraction, Inheritance, and Polymorphism.
  - Classes and Objects in Python:
    - Defining classes and objects.
    - Methods, attributes, and constructors.
- **Programming Essentials:**
  - Variables, data types, and operators in Python.
  - Control structures: Loops, conditionals, and functions.

Unit 2: Advanced OOP and Python Data Handling

- **Advanced OOP Techniques:**
  - Inheritance and Polymorphism:
    - Method overriding and overloading.
    - Abstract classes and interfaces.
  - Encapsulation with access modifiers: Public, private, and protected.
  - Error and Exception Handling:
    - Using try-except blocks in Python.
    - Creating custom exceptions.
- **Data Handling in Python:**
  - Introduction to NumPy for numerical computations:
    - Arrays, matrix operations, and linear algebra.
  - Working with Pandas:
    - DataFrames, data cleaning, and transformations.
    - Grouping, merging, and aggregating data.

Unit 3: Python for Data Visualization and AI Basics

- **Data Visualization:**
  - Creating plots with Matplotlib:
    - Line charts, bar plots, histograms, and scatter plots.
  - Advanced visualizations with Seaborn:
    - Heatmaps, pair plots, and customizations.
- **Introduction to AI and Machine Learning:**
  - Overview of AI and its applications.
  - Introduction to machine learning in Python:
    - Scikit-learn basics: Linear regression, classification, and clustering.
    - Model evaluation metrics: Accuracy, precision, recall.
- **Hands-On Practice:**
  - Implement basic AI models using Python.
  - Visualize and interpret results.

Unit 4: AI Applications and Capstone Project

- **Python for Advanced AI Applications:**
  - Deep Learning Basics:
    - Neural networks and introduction to TensorFlow/PyTorch.
    - Simple neural network implementation (e.g., classification task).
  - Natural Language Processing (NLP):
    - Text preprocessing with NLTK and spaCy.

        - Implementing a sentiment analysis application.
- **Capstone Project:**
    - Develop a comprehensive project combining OOP principles and Python for AI.
    - Example projects:
        - AI-based inventory management system.
        - Text classification or sentiment analysis tool.
        - Image classifier using a basic CNN.

SUGGESTED READING:

1. "Object-Oriented Programming with C++ and Python" by Balagurusamy E.
2. "Python Programming for Beginners: A Guide for Engineers" by Harsh Bhasin
3. "Artificial Intelligence and Machine Learning with Python" by Prateek Jain and Anurag Bhardwaj
4. "Artificial Intelligence with Python" by Santanu Chattopadhyay.
5. "Data Science and Machine Learning with Python" by U. Dinesh Kumar and Manaranjan Pradhan
6. "Programming Python" by Mark Lutz

SEMESTER-II

AIMLSPL14074: DEEP NEURAL NETWORKS

(Credit 4, Total Marks=100)

(Theory: 70 Marks, Internal: 30 Marks)

Course Learning Objective:

- To provide students with an understanding of the theoretical and mathematical foundations of deep learning and neural networks.
- To study various neural network architectures, including feedforward networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs).
- To introduce students to advanced topics such as generative adversarial networks (GANs), autoencoders, transfer learning, and attention mechanisms.
- To explore emerging areas such as transformers and explainable AI (XAI) while fostering awareness of ethical concerns in AI development and deployment.

Course Learning Outcome:

- Explain the theoretical foundations of deep learning, including neural network structures and activation functions.
- Apply regularization techniques to address overfitting and improve generalization.
- Demonstrate proficiency in using advanced optimization algorithms like Adam and RMSprop.
- Utilize generative models such as GANs and autoencoders for image synthesis and data compression.
- Develop familiarity with transformer architectures, attention mechanisms, and vision transformers (ViTs).

- Discuss the ethical implications of deep learning applications, including bias, privacy, and fairness**.**

Unit 1: Introduction to Deep Learning and Neural Networks

- Fundamentals of Deep Learning:
  - Overview of deep learning and its applications.
  - Differences between traditional ML and deep learning.
  - Biological and artificial neurons.
- Basic Neural Network Concepts:
  - Perceptrons and multilayer perceptrons (MLPs).
  - Activation functions: Sigmoid, ReLU, Tanh, Softmax.
  - Feedforward networks and their working principles.
- Training Neural Networks:
  - Forward and backward propagation.
  - Cost functions and loss functions.
  - Gradient descent and optimization techniques.

Unit 2: Architectures of Deep Neural Networks

- Deep Feedforward Neural Networks:
  - Structure and design principles.
  - Overfitting and underfitting.
  - Regularization techniques: L1, L2, and dropout.
- Convolutional Neural Networks (CNNs):
  - Architecture: Convolutional layers, pooling layers, and fully connected layers.
  - Applications in image classification, object detection, and segmentation.
- Recurrent Neural Networks (RNNs):
  - Overview of sequential modeling and RNN basics.
  - Challenges: Exploding and vanishing gradients.
  - Variants: Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU).

Unit 3: Advanced Deep Neural Network Concepts

- Deep Learning Optimization Techniques:
  - Adam, RMSprop, and other advanced optimizers.
  - Batch normalization and learning rate scheduling.
- Autoencoders and Variants:
  - Structure and applications of autoencoders.
  - Denoising autoencoders and variational autoencoders (VAEs).
- Generative Models:
  - Generative Adversarial Networks (GANs): Architecture and training.
  - Applications in image synthesis and style transfer.
- Transfer Learning:
  - Pretrained models and fine-tuning for domain-specific tasks.

Unit 4: Practical Applications and Emerging Trends

- Deep Learning in Practice:
  - Training pipelines: Dataset preparation, model evaluation, and hyperparameter tuning.
  - Tools and frameworks: TensorFlow, PyTorch, and Keras.
- Explainable AI (XAI):
  - Need for interpretability in deep neural networks.
  - Techniques for visualizing and understanding models.

- Emerging Trends in Deep Learning:
    - Transformers and attention mechanisms.
    - Vision Transformers (ViTs) and GPT models.
    - Advances in deep reinforcement learning.

SUGGESTED READINGS:

- "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville.
- "Neural Networks and Deep Learning" by Michael Nielsen (Online Book).
- "Artificial Intelligence: A Guide to Intelligent Systems" by Michael Negnevitsky.
- "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron.
- "Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play" by David Foster.
- "Introduction to Artificial Neural Networks" by Sivanandam S. N. and Deepa S. N.

AIMLSPL14084: INTRODUCTION TO DATA SCIENCE
(Credit 4, Total Marks=100)
(Theory: 70 Marks, Internal: 30 Marks)

Course Learning Objective:

- To develop the ability for problem analysis and decision-making
- To gain practical, hands-on experience with statistics programming languages and big data tools through coursework and applied research experiences
- To apply data science concepts and methods to solve problems in real-world contexts and will communicate these solutions effectively
- To develop the ability to build and assess data-based models

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Understand Data Science fundamentals: Concepts, principles, tasks, and applications.
- Learn Data Preparation: Types, normalization, missing values, feature selection.
- Explore Proximity Measures: Characteristics, types for numeric, categorical, and mixed data.
- Acquire Classification skills: Models, validation methods, metrics, Bayesian and Naive Bayes classification.
- Familiarize with Decision Trees: Rules, attribute selection, gain, Gini Index.
- Learn Distance-Based Classification: Nearest Neighbor Classifier.
- Gain knowledge of SVM and Neural Networks.

Unit-1: Introduction to data science and data preparation

- Concepts, Principle, Knowledge discovery and Data Science, what is not Data Science? Evolution of Data Science, Data Science Tasks, Applications.
- Data types-Nominal, Ordinal, Categorical, and Ratio, Normalization, Discretization, Missing Value Estimation, Sampling, Feature Selection.

Unit-2: Proximity measures and introduction to classification

- Proximity Measures and its Characteristics, Identity, Non-negativity, Symmetricity, Types of Proximity Measures-Numeric, Categorical, Mixed-type.
- Definition, Machine Learning Model, Classification and Prediction, Data Separability and Decision Boundary, Validation Methods, Cross-validation, Bootstrapping, Assessment Metrics Confusion Matrix, Sensitivity and Specificity, Accuracy.

Unit-3: Classification models

- Bayesian Classification, Naive Bayes Classification.
- Decision Tree: Decision Rules, Attribute Selection Measures, Gain, Gini Index, ID3
- Distance Based Classification: Simple Approach, Nearest Neighbour Classifier.
- Support Vector Machines: SVM Classifier, Vector Kernels, Application-specific Kernels, Implementing SVM using R/Python.
- Neural Network: Biological brain, Single Layer Perceptron, Multilayer Perceptron, Feed-Forward and Back-Propagation, Implementing ANN using R/Python.

Unit-4: Clustering and big data analysis

- Euclidean Distance, Manhattan Distance, Minkowski Distance, Partitional, Hierarchical, Density-based techniques, Assessment methods.
- Big Data, Overview, Big Data Analytics Architecture, Big Data Analytics Tool, Design of Big Data Mining Algorithms, Incremental and Distributed Algorithm.

SUGGESTED READINGS:

1. J. Kalita, DK Bhattacharyya, S Roy (2023), Fundamentals of Data Science Theory & Practice, Elsevier.
2. Tom M. Mitchell (2017), Machine Learning, Me Grew Hills.
3. Dunham, M. H. (2006). Data mining: Introductory and advanced topics. Pearson Education India.
4. Han, J., Pei, J., & Tong, H. (2022). Data mining concepts and techniques. Morgan kaufmann.
5. Vander Plas, J. (2016). Python data science handbook: Essential tools for working with data. O'Reilly Media, Inc.

AIMLSPL14094:ELECTIVE-I (MOOC Courses)
(Credit 4, Total Marks=100)
(Theory = 100 Marks)

Guidelines for MOOC Elective:

- Students can register for MOOC Elective and credit transfer happens only during 3rd Semester after submission of this Certificate.
- Students can opt for any one course either from odd semester or from even semester as offered by NPTEL in the respective semester. The subject will be approved by DC.
- The list of SWAYAM based online credit courses for the ensuing semester shall be notified on the SWAYAM platform before the 1st November for the January semester and before the 1st June for the July semester, every year.
- The Department Co-Coordinator shall incorporate the marks or grades obtained by the student in the marks sheet that counts for final processing of the equivalent grading as per the regulation for the PG Programme.

AIMLSPL140104: NATURAL LANGUAGE PROCESSING
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objectives:

- Students will be able to comprehend the importance of using natural language processing when resolving issues in the real world.
- Enables students to apply and match the proper processing technique to a given situation.
- Students will be in a position to exhibit the necessary design abilities for large collection sets. Additionally, capable of understanding and presenting cutting-edge, sophisticated NLP research materials to an audience.

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Discern the concept of POS tagging and CFG for the English language.
- Cognize the Vector Representation of words and skip-gram models
- Explore semantic analysis algorithms and deep learning techniques, to apply them in various NLP applications.
- Acquainted with Mathematical and programming tools for implementing NLP applications.

Unit-1: Basics of Machine Learning

Python Programming language, Basics of Probability, Introduction - terminologies - empirical rules – Statistical Properties of words – Probability and NLP – Vector Space Models - Pre-processing Tokenization, Parts-Of-Speech (POS) tagging, chunking, syntax parsing, Dependency parsing.

Unit-2: Vector Representation of words

Contextual Understanding of text – Cooccurrence of matrix – N-grams – Dense Word Vector. Word2Vec – CBOW and Skip-gram Models – One-word learning architecture- Forward pass for Word2Vec – Reduction of complexity – subsampling and negative sampling. Continuous Skip-Gram Model, GloVe, BERT, XLNet.

Unit-3: NLP Applications

Named Entity Recognition, Sentiment analysis, Text categorization using Machine learning algorithms, SVD and Latent semantic Indexing, Probabilistic Latent Semantic Indexing (pLSI) and Latent Dirichlet Allocation (LDA). Deep Learning for NLP: Neural Networks Basics, Feedforward Neural Network, Recurrent Neural Networks, LSTM, An Introduction to Transformers and Sequence-to-Sequence Learning.

Unit-4: Historical Approaches to Machine Translation

Statistical Machine Translation – Translation Models – Healthcare Data analysis and Text visualization: Summarizing lengthy blocks of narrative text, such as a clinical note or academic journal article. Answering unique free-text queries that require the synthesis of multiple data sources. Introduce Mathematical and programming tools to visualize a large collection of text documents.

SUGGESTED READINGS:

1. C.D. Manning et al, “Foundations of Statistical Natural Language Processing,” MIT Press. MIT Press, 1999. isbn: 9780262133609.
2. James Allen, “Natural Language Processing with Python”, O’Reilly Media, July 2009.
3. NiladriSekhar Dash and S. Arulmozi, Features of a Corpus. Singapore: Springer Singapore,2018, pp. 17– 34. ISBN: 978-981-10-7458-5.
4. Ian Goodfellow, YoshuaBengio, and Aaron Courville, Deep Learning, http://www.deeplearningbook.org. MIT Press, 2016.
5. NitinIndurkhya and Fred J Damerau, ”Handbook of natural language processing,” Chapman and Hall/CRC, 2010.
6. Daniel Jurafsky and James H. Martin ”Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition,” 1st. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000. isbn: 0130950696.

AIMLSPL140114: COMPUTATIONAL INTELLINGENCE LABORATORY-I
(Credit 4, Total Marks=100)
(Practical = 70 Marks, Internal = 30 Marks)

Course Learning Objectives:

- Equip students with a strong foundation in Python programming.
- Provide a solid understanding of OOP principles: Encapsulation, Abstraction, Inheritance, and Polymorphism.
- Enable students to design and implement classes and objects in Python.
- Build advanced skills in Python, including error handling, advanced OOP techniques (inheritance, polymorphism), and data encapsulation.
- Train students to use libraries such as **NumPy** for numerical computations and **Pandas** for data manipulation, cleaning, and transformation.
- Enable students to develop and deploy machine learning models, including regression, classification, and clustering.
- Introduce deep learning frameworks such as **TensorFlow** and **PyTorch** to build neural networks and tackle AI challenges.
- Familiarize students with **Natural Language Processing (NLP)** techniques using libraries like **NLTK** and **spaCy**.

Course Learning Objectives:

- Develop well-structured, efficient, and readable Python code.
- Apply fundamental programming concepts such as variables, data types, operators, control structures, and functions to solve real-world problems.
- Create and work with custom classes and objects, and apply advanced OOP concepts such as method overriding, abstract classes, and inheritance to design scalable and reusable code.
- Leverage libraries like **NumPy** and **Pandas** for efficient data manipulation, cleaning, and transformation.
- Use **Matplotlib** and **Seaborn** to create various types of data visualizations, including line charts, bar plots, histograms, heatmaps, and scatter plots.
- Build and train simple deep learning models, including neural networks, using frameworks like **TensorFlow** or **PyTorch**.
- Apply **NLP** techniques to process and analyze textual data using libraries like **NLTK** and **spaCy**.

Unit 1:Machine Learning

- Basic ML Implementations: Implement Linear Regression from scratch using NumPy, Build a Logistic Regression classifier using Scikit-learn, Write a program to calculate and visualize confusion matrices, Create a Python program to load, clean, and preprocess datasets using Pandas.
- Supervised Learning Algorithms: Code a Decision Tree algorithm from scratch, Implement k-Nearest Neighbors (k-NN) without using libraries, Build and train an SVM classifier using Scikit-learn, Train and evaluate a Random Forest model on a real-world dataset.
- Unsupervised Learning: Implement the K-Means clustering algorithm from scratch, Perform Principal Component Analysis (PCA) using Scikit-learn, Create a Python script for hierarchical clustering visualization, Perform dimensionality reduction using t-SNE on a dataset.
- End-to-End ML Project: Build an ML pipeline: data cleaning, model training, and evaluation, Train a Gradient Boosting model for classification, Save and deploy a trained model using Flask/Streamlit, Create a web app to visualize predictions.

Unit 2:Deep Neural Networks Programming

- Foundations of Neural Networks: Build a Perceptron in Python from scratch using NumPy, Implement a feedforward neural network for binary classification, Write Python code for common activation functions: ReLU, Sigmoid, Tanh, Visualize loss function behavior with different learning rates.
- Building Neural Networks: Train a DNN using PyTorch on the MNIST dataset, Write a Python program to apply Dropout for regularization, Implement Batch Normalization in PyTorch, Experiment with different optimizers: SGD, Adam, RMSprop.
- Advanced Architectures: Code a Convolutional Neural Network (CNN) for image classification, Build an RNN/LSTM model for text generation, Fine-tune a pretrained model like ResNet on a custom dataset, Implement a VariationalAutoencoder (VAE) for image generation.
- DNN Projects: Image classification project using transfer learning, Train a chatbot using sequence models, Create a GAN to generate synthetic images, Deploy a trained model as an API with FastAPI.

Unit 3: Data Science Programming

- Data Wrangling and EDA: Write a Python script to clean and preprocess a messy dataset, Perform exploratory data analysis (EDA) and visualization using Matplotlib and Seaborn,

Create a correlation heatmap to identify relationships in data, Implement outlier detection and removal using statistical methods.

- Statistical Programming: Write Python code for hypothesis testing (t-tests, chi-square tests), Implement linear regression using statsmodels, Simulate A/B tests with Python, Program a bootstrap sampling method for statistical inference.
- Data Modeling: Create a Python pipeline for predictive modelling, Implement clustering algorithms (e.g., K-Means) and visualize results, Perform time series analysis using ARIMA in Python, Build a recommender system using collaborative filtering.
- Real-World Data Science Projects: Predict house prices using regression models, Perform customer segmentation using clustering techniques, Build a dashboard to visualize sales trends using Plotly/Dash, Train a fraud detection model using anomaly detection techniques.

Unit 4: Natural Language Processing Programming

- Text Processing: Write Python scripts for tokenization, stemming, and lemmatization using NLTK, Implement a TF-IDF vectorizer from scratch, Create a Python program to generate word embeddings using GloVe, Visualize text data using word clouds.
- Building NLP Models: Train a Naive Bayes classifier for sentiment analysis. Build a sequence-to-sequence model for text summarization, Fine-tune BERT for text classification tasks, Create an LSTM model to perform named entity recognition (NER).
- Advanced NLP Programming: Train a Transformer model for machine translation, Implement attention mechanisms in sequence models, Fine-tune mBART for multilingual tasks, Perform topic modeling using Latent Dirichlet Allocation (LDA).
- NLP Projects: Build a sentiment analysis app for social media data, Develop a machine translation system for a low-resource language, Create a chatbot using Hugging Face Transformers, Deploy an NLP model for question answering using FastAPI.

SUGGESTED READINGS:

1. Natural Language Processing in Action (Indian Adaptation) by Hitesh Choudhary.
2. Introduction to Machine Learning with Python by Andreas C. Müller and Sarah Guido.
3. Natural Language Processing with Python by Steven Bird, Ewan Klein, and Edward Loper.
4. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aurélien Géron.
5. Towards Data Science - ML Tutorials.
6. Kaggle - Machine Learning Tutorials.
7. Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville.
8. Neural Networks and Deep Learning by Michael Nielsen (available free online).
9. Deep Learning with PyTorch (Official Tutorials).
10. Analytics Vidhya - Deep Learning Tutorials.
11. Python for Data Analysis by Wes McKinney.
12. Speech and Language Processing by Daniel Jurafsky and James H. Martin.
13. Natural Language Processing with Python by Steven Bird, Ewan Klein, and Edward Loper.

SEMEATER-III (Only for A and C)
AIMLSPL25014: GENERATIVE AI
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course learning Objectives:

- Understand the fundamental principles and techniques underlying Generative AI.
- Explore a variety of generative models, including GANs, VAEs, and autoregressive models.
- Gain hands-on experience with implementation tools such as TensorFlow, PyTorch, and HuggingFace.
- Apply generative techniques to real-world domains, including text, image, audio, and video generation.
- Examine ethical concerns, current challenges, and emerging trends in the field of Generative AI.

Course learning Outcomes:

- Understand and explain the architecture of major generative models.
- Design and implement generative models such as GANs and VAEs.
- Evaluate the performance of generative models using appropriate metrics
- Apply generative models to NLP, computer vision, and audio processing.
- Discuss ethical implications and societal impacts of generative technologies.

UNIT I: Introduction to Generative AI

- Generative vs. Discriminative Models

- Types of Generative Models: Explicit vs. Implicit
- Applications: Text Generation, Image Synthesis, Style Transfer, Data Augmentation
- Mathematical Foundations: Probability, Bayes' Theorem, Likelihood Estimation

UNIT II: Variational Autoencoders (VAEs) and Diffusion Models

- Autoencoders: Architecture and Loss Functions
- Variational Autoencoders: Latent Space, KL Divergence
- VAE Limitations and Variants
- Introduction to Diffusion and Score-Based Generative Models

UNIT III: Generative Adversarial Networks (GANs)

- GAN Framework: Generator, Discriminator, Adversarial Loss
- Training Challenges: Mode Collapse, Convergence Issues
- Advanced GAN Variants: DCGAN, StyleGAN, CycleGAN, BigGAN
- Evaluation Metrics: Inception Score (IS), Fréchet Inception Distance (FID)

UNIT IV: Applications & Ethics in Generative AI

- Generative Models for NLP: GPT, BERT, Transformers as Generators
- Text-to-Image (e.g., DALL·E, Midjourney), Image-to-Text, and Audio Generation
- Transfer Learning and Fine-tuning of Pre-trained Models
- Ethical Considerations: Deepfakes, Bias, Misinformation, and Responsible AI
- Case Studies and Future Trends.

SUGGESTED READINGS:

1. Deep Learning – Ian Goodfellow, Yoshua Bengio, Aaron Courville
2. Generative Deep Learning – David Foster
3. Hands-On GANs with Keras – Rafael Valle
4. Papers with Code, Arxiv.org
5. HuggingFace Tutorials & Official Docs for PyTorch, TensorFlow, Transformers

AIMLSPL25024: COMPUTER VISION AND IMAGE PROCESSING
(Credit 4, Total Marks=100)
(Theory: 70 Marks, Internal: 30 Marks)

Course Learning Objective:

- Grasp the fundamental principles of image formation, representation, and processing.
- Learn about the mathematical and algorithmic foundations of computer vision, such as linear algebra, geometry, and optimization.
- Apply feature matching techniques for applications like panorama stitching and image retrieval.
- Solve practical problems like face detection, object tracking, and motion analysis.
- Work on domain-specific applications such as autonomous driving, medical imaging, and augmented reality.
- Discuss the ethical considerations and societal impacts of computer vision technologies, such as privacy concerns and algorithmic bias.

Course Learning Outcome:

- Demonstrate an understanding of basic image processing concepts, including image representation, transformations, and feature extraction.
- Analyze and apply mathematical models for camera calibration, 3D reconstruction, and object recognition.
- Evaluate the performance of computer vision algorithms using appropriate metrics and optimize them for improved accuracy and efficiency.

- Collaborate effectively in teams to develop and present computer vision projects, demonstrating strong communication skills in conveying technical concepts.

Unit-1: Fundamentals of Image Processing

- Image Representation:
  - Pixels, grayscale vs. color images
  - Image histograms and color spaces (RGB, HSV)
- Image Enhancement Techniques:
  - Contrast adjustment, histogram equalization
  - Smoothing and sharpening filters
- Image Segmentation:
  - Thresholding (binary and adaptive)
  - Edge detection (Sobel, Canny)
- Feature Extraction:
  - Corners and edges (Harris detector)
  - Histogram of gradients (HoG basics)

Unit-2: Machine Learning for Vision

- Supervised Learning Basics:
  - Image classification using SVM, k-NN
- Unsupervised Learning:
  - K-means clustering for segmentation
- Introduction to CNNs:
  - Convolution, pooling, activation
- Transfer Learning:
  - Using pre-trained models (MobileNet, VGG)

Unit-3: Tools, Programming

- Programming Environment:
  - Python for computer vision
  - OpenCV basics: image reading, filtering, transformation
- Deep Learning Libraries:
  - Basics of TensorFlow or PyTorch
  - Using Keras for model building
- Data Preprocessing:
  - Image resizing, normalization, augmentation
- Mini Projects:
  - Real-world tasks: plant disease app, mask detector, barcode scanner
- Presentation of Results:
  - Accuracy metrics (precision, recall)
  - Visualization of model predictions

Unit-4: Applications of Computer Vision

- Object Detection:
  - Introduction to bounding boxes
  - Using pre-trained detectors (YOLO-lite or SSD)
- Image-Based Projects:

    - Leaf disease detection (relevant for agriculture)
    - Face detection and emotion recognition
- Video & Motion Basics:
    - Frame-by-frame processing
    - Simple object tracking using OpenCV

SUGGESTED READINGS:

1. Sonka & Hlavac – Image Processing
2. Gonzalez – Digital Image Processing
3. Szeliski – Computer Vision: Algorithms & Applications
4. Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods.
5. Computer Vision: Algorithms and Applications by Richard Szeliski
6. Digital Image Processing by S. Sridhar.
7. Pattern Recognition and Image Analysis by Earl Gose (Indian Adaptation).
8. Deep Learning for Computer Vision with Python by Adrian Rosebrock.
9. Practical Machine Learning and Image Processing by Himanshu Singh.

AIMLSPL25034: ELECTIVE-II (MOOC Courses)
(Credit 4, Total Marks = 100)
(Theory = 100 Marks)

Guidelines for MOOC Elective:

- Students can register for MOOC Elective and credit transfer happens only during 3rd Semester after submission of this Certificate.
- Students can opt for any one course either from odd semester or from even semester as offered by NPTEL in the respective semester. The subject will be approved by DC.
- The list of SWAYAM based online credit courses for the ensuing semester shall be notified on the SWAYAM platform before the 1st November for the January semester and before the 1st June for the July semester, every year.
- The Department Co-Coordinator shall incorporate the marks or grades obtained by the student in the marks sheet that counts for final processing of the equivalent grading as per the regulation for the PG Programme.

AIMLSPL25042: INTERNSHIP
(Credit 2, Total Marks=50)
(Practical = 50 Marks)

Guidelines for Internship:

- Individual students must carry out Internship training at industry/ University/ Collage about 4 weeks.
- Student shall submit a detailed report on internship work (15 to 20 Pages) in a format as specified by the department. Internal guide and industry University/ Collage personnel will jointly evaluate the student performance for 50 Marks by taking demonstration and presentation of the work carried during internship.
- Internship to be carried out between end of 2nd Semester and begining of 3rd semester vacation only.

AIMLSPL25052: SEMINAR
(Credit 2, Total Marks=50)
(Practical = 50 Marks)

Course learning Objectives:

- To create an environment to engage students in delivering and listening to interesting talks that promotes discussion
- To provide students with opportunity to learn new concepts and skills acquired in core courses and further extend these ideas to solve research/industry related problems
- Know how to read research papers critically and efficiently
- To learn fundamental principles, generalizations and important theories of Computer Science
- To enable students to find their own field of interest in academia, industry or entrepreneurship
- To help students develop their own learning and teaching styles and communication skills

Course learning Outcomes: Completion of this course, students should be able to do the following:

- Explain factual knowledge (terminology, classifications, methods, trends). of current areas of research.
- State and explain some fundamental principles, generalizations, or theories the student has learned in this course.
- To apply gained knowledge in thinking, problem solving, or decisions making process.
- To achieve specific skills, competencies, and points of view needed by computing professionals.
- To judge the value of different contributions
- To identify promising new directions

Contents are:

A seminar should be given by an individual student, based on topics chosen from the emerging areas and technologies of Computer Science and Computer Applications. References from journals such as IEEE, ACM etc...shall be used. A report on this seminar with 15 – 20 pages should be prepared.

- Student shall present their work till literature survey.
- Student shall present their complete work.

SUGGESTED READINGS:

A list of works will be posted by mentors/teachers at the start of the course. The students also have the option of choosing works according to his/her own areas of interest.

AIMLSPL25064: COMPUTATIONAL INTELLIGENCE LABORATORY-II
(Credit 4, Total Marks=100)
(Practical = 70 Marks, Internal = 30 Marks)

Course learning Objectives:

- Explain the concepts of generative models such as VAEs, GANs, and transformers.
- Demonstrate knowledge of latent space and its use in data generation.
- Build and train basic GANs and VAEs for image and text generation.
- Fine-tune pretrained transformer-based models (e.g., GPT).
- Train and fine-tune CNNs for image classification and object detection.
- Explore pretrained models (ResNet, YOLO, Mask R-CNN) for advanced tasks.
- Develop applications like real-time face detection, autonomous navigation, and medical imaging systems.
- Deploy models on edge devices and optimize them for mobile platforms.

Course learning Outcomes:

- Demonstrate the ability to implement and train generative models such as GANs, VAEs, and transformer-based architectures.
- Generate high-quality synthetic data, including text, images, and audio, using advanced generative models.
- Develop and deploy multimodal AI systems for applications like text-to-image generation or video synthesis.
- Evaluate the performance of generative models using metrics like FID, BLEU, and perplexity.

- Create and optimize generative applications for real-world use cases, such as content creation and personalization.
- Apply fundamental image processing techniques like edge detection, thresholding, and histogram equalization.
- Design and train deep learning models (e.g., CNNs, YOLO, and Mask R-CNN) for vision tasks like object detection, segmentation, and classification.
- Utilize advanced vision architectures such as Vision Transformers (ViTs) for state-of-the-art performance in image analysis.

Unit1: Generative AI Programming

- Basics of Generative AI: Implement Variational Autoencoders (VAEs) for image generation, Write a basic Generative Adversarial Network (GAN) from scratch, Generate text using Markov Chains and N-grams, Experiment with pretrained generative models like GPT-2.
- Advanced Generative Models: Implement Conditional GANs (cGANs) for targeted image generation, Train CycleGAN for style transfer (e.g., photo to painting), Fine-tune a transformer model for text generation, Use Stable Diffusion for advanced image generation.

Unit 2: Multimodal Generative AI:

- Implement text-to-image generation using CLIP and DALL·E, Create a script for audio-to-text generation using Whisper, Build a multimodal chatbot with image and text inputs, Experiment with generative music using OpenAI Jukebox.
- Generative AI Applications: Create an AI Art Generator using GANs, Build a personalized chatbot using GPT-3, Develop a video generation tool from text prompts, Deploy a generative model as a web application.

Unit 3: Computer Vision and Image Processing:

- Image Processing Basics, Perform image transformations (resize, rotate, crop) using OpenCV, Implement edge detection using Sobel and Canny operators, Apply histogram equalization for contrast adjustment, Segment images using thresholding techniques.
- Deep Learning for Vision: Train a CNN from scratch for image classification, Fine-tune a pretrained ResNet or EfficientNet model, Implement YOLOv5 for real-time object detection, Train a UNet for semantic segmentation.

Unit 4: Advanced Vision Techniques:

- Build a Mask R-CNN for instance segmentation, Implement optical flow for motion tracking in videos, Use self-supervised techniques like SimCLR for representation learning, Explore vision transformers (ViTs) for image classification
- Vision Applications: Create a real-time face recognition system, Build an autonomous vehicle's lane detection system, Perform medical image analysis for disease prediction, Deploy an object detection model on a mobile device.

SUGGESTED READINGS:

1. Generative Deep Learning by David Foster.
2. Deep Learning Illustrated by Jon Krohn et al.
3. Deep Learning with Python by Nikhil Ketkar.
4. Practical Machine Learning and Image Processing by Himanshu Singh.

5. Deep Learning for Computer Vision by Rajalingappaa Shanmugamani.
6. Towards Data Science - GANs and Variational Autoencoders.
7. Hugging Face Blog.

SEMESTER-IV (Only A)
AIMLSPL25074: RAINFORCEMENT LEARNING
(Credit 4, Total Marks= 100)
(Theory = 70 Marks, Internal = 30 Marks)

Course learning Objectives:

- Understand the core concepts of RL, including agents, environments, states, actions, and rewards
- Learn the concepts of policy (deterministic and stochastic) and value functions (state-value and action-value).
- Explore dynamic programming techniques for solving RL problems, including policy evaluation, policy improvement, and value iteration.
- Study model-free methods like Monte Carlo (MC) methods, Temporal Difference (TD) Learning, and Q-Learning.
- Understand how function approximation is used in RL, especially with large or continuous state and action spaces.
- Understand how to evaluate RL agents using metrics like cumulative reward and episode success rate.

Course learning Outcomes:

- Understand key components of RL, Formulate problems as MDPs, Solve small MDPs using Dynamic Programming.
- Distinguish between model-based and model-free methods, implement SARSA and Q-Learning for different environments, understand convergence and stability in learning algorithms.
- Understand how deep learning integrates with RL, train Deep Q-Networks for complex state spaces, apply RL in continuous or high-dimensional environments.

- Implement basic policy gradient algorithms, understand and apply PPO for continuous control tasks, get exposure to real-world applications of RL.

Unit 1: Foundations of Reinforcement Learning

- Introduction to RL: What and Why?
- Elements of RL: Agent, Environment, Rewards, States, Actions
- Markov Decision Processes (MDPs)
- Policies, Value Functions, and Bellman Equations
- Dynamic Programming: Policy Evaluation, Policy Iteration, Value Iteration

Unit 2: Model-Free Prediction and Control

- Monte Carlo Methods
- Temporal-Difference Learning
- SARSA and Q-Learning
- Exploration vs Exploitation (ε-greedy, Softmax)
- Eligibility Traces (Intro to n-step methods)

Unit 3: Function Approximation and Deep RL

- Function Approximation in RL
- Linear vs Non-linear Approximators
- Deep Q-Networks (DQN)
- Experience Replay and Target Networks
- Challenges: Overestimation, Instability

Unit 4: Policy Gradient Methods and Advanced Topics

- Introduction to Policy Gradients
- REINFORCE Algorithm
- Actor-Critic Methods
- Proximal Policy Optimization (PPO)
- Multi-Agent RL (brief), Applications in Robotics, Game AI

SUGGESTED READINGS:

1. **"Reinforcement Learning: An Introduction"** by Richard S. Sutton and Andrew G. Barto
2. David Silver's RL Course (UCL)
3. OpenAI Spinning Up in Deep RL
4. Papers from NeurIPS / ICML on RL advances

AIMLDIS250816: DISSERTATION
(Credit 16, Total Marks=400)
(Practical = 200 Marks)

Course learning Objectives:

- Develop the ability to identify and formulate a clear, researchable problem within the domain of AI and ML.
- Design and implement a research methodology appropriate to the chosen topic.
- Demonstrate a deep understanding of core AI and ML concepts, such as neural networks, optimization, probabilistic modeling, and data analysis.
- Apply advanced AI/ML techniques to solve real-world problems in areas such as natural language processing, computer vision, reinforcement learning, or generative AI.
- Gain expertise in programming languages (e.g., Python, R) and frameworks (e.g., TensorFlow, PyTorch, Scikit-learn) for implementing AI and ML models.
- Foster a mindset for continuous learning in the rapidly evolving field of AI and ML.
- Demonstrate independence, initiative, and a systematic approach to problem-solving and research.

Course learning Outcomes: Completion of this course, students should be able to do the following:

- Gain hands-on experience in the Artificial Intelligence decision cycle.
- Perform exploratory data analysis to understand the dataset.
- Build and fine -tune models using cutting-edge AI algorithms
- Present and interpret results obtained from the models.

Dissertation Overview:

The dissertation is a cornerstone of the M.Sc. program and is a vital component of the curriculum. It is designed to provide students with the opportunity to demonstrate their ability to independently tackle challenging problems, apply advanced knowledge, and contribute original ideas to their field of study. Undertaking the dissertation with utmost seriousness is essential, as it is a key step in preparing students for real-world challenges and further academic pursuits.

The dissertation represents a significant milestone in the M.Sc. program, offering students an invaluable opportunity to showcase their research capabilities, technical skills, and creative thinking. By dedicating an entire semester to this endeavour, students will not only full-fill academic requirements but also set the foundation for future achievements in their professional or academic careers.

Dissertation Report Requirements:

Format

The Detailed Report should be between 60 to 80 pages and adhere to the format specified by the department. Key sections include:

1. Title Page
    - Project title
    - Student's name and roll number
    - Supervisor's name
    - Department and institution name
    - Date of submission
2. Abstract
    - A concise summary of the research problem, methodology, results, and conclusions.
3. Acknowledgements
    - A section to express gratitude to those who supported the project.
4. Table of Contents
    - A structured outline of the report's sections and subsections with page numbers.
5. Introduction
    - Background of the topic.
    - Significance of the study.
    - Objectives and scope.
6. Literature Review
    - Analysis of existing work and identification of research gaps.
7. Methodology
    - Detailed explanation of the approach, tools, and techniques used.
8. Implementation
    - Step-by-step description of the development or problem-solving process.
9. Results and Analysis
    - Presentation of findings using tables, graphs, and charts.
    - Discussion of the implications and limitations.
10. Conclusion and Future Work
    - Summary of the research.
    - Suggestions for further exploration.
11. References
    - A complete list of all cited works in the specified format.
12. Appendices (if necessary)
    - Supplementary material such as code, datasets, or detailed calculations.

SUGGESTED READINGS:

1. Research papers from journals like IEEE Transactions on Neural Networks and Learning Systems, Journal of Machine Learning Research (JMLR), and Nature Machine Intelligence.
2. Online resources and documentation for tools and frameworks such as TensorFlow, PyTorch, and Scikit-learn.
3. Tutorials and courses from platforms like Coursera, edX, and Kaggle Learn to stay updated on current trends and practices in AI and ML.

SEMESTER III (For Option B only)
AIMLDIS250120: RESEARCH THESIS/ DISSERTATION-I
(Credit 20, Total Marks=500)
(Practical = 500 Marks)

Course learning Objectives:

- Develop the ability to identify and formulate a clear, researchable problem within the domain of AI and ML.
- Design and implement a research methodology appropriate to the chosen topic.
- Demonstrate a deep understanding of core AI and ML concepts, such as neural networks, optimization, probabilistic modeling, and data analysis.
- Apply advanced AI/ML techniques to solve real-world problems in areas such as natural language processing, computer vision, reinforcement learning, or generative AI.
- Gain expertise in programming languages (e.g., Python, R) and frameworks (e.g., TensorFlow, PyTorch, Scikit-learn) for implementing AI and ML models.
- Foster a mindset for continuous learning in the rapidly evolving field of AI and ML.
- Demonstrate independence, initiative, and a systematic approach to problem-solving and research.

Course learning Outcomes: Completion of this course, students should be able to do the following:

- Gain hands-on experience in the Artificial Intelligence decision cycle.
- Perform exploratory data analysis to understand the dataset.
- Build and fine -tune models using cutting-edge AI algorithms
- Present and interpret results obtained from the models.

Dissertation Overview:

The dissertation is a cornerstone of the M.Sc. program and is a vital component of the curriculum. It is designed to provide students with the opportunity to demonstrate their ability to independently tackle challenging problems, apply advanced knowledge, and contribute original ideas to their field of study. Undertaking the dissertation with utmost seriousness is essential, as it is a key step in preparing students for real-world challenges and further academic pursuits.

The dissertation represents a significant milestone in the M.Sc. program, offering students an invaluable opportunity to showcase their research capabilities, technical skills, and creative thinking. By dedicating an entire semester to this endeavour, students will not only full-fill academic requirements but also set the foundation for future achievements in their professional or academic careers.

Dissertation Report Requirements:

Format

The Detailed Report should be between 60 to 80 pages and adhere to the format specified by the department. Key sections include:

13. Title Page
    - Project title
    - Student's name and roll number
    - Supervisor's name
    - Department and institution name
    - Date of submission
14. Abstract
    - A concise summary of the research problem, methodology, results, and conclusions.
15. Acknowledgements
    - A section to express gratitude to those who supported the project.
16. Table of Contents
    - A structured outline of the report's sections and subsections with page numbers.
17. Introduction
    - Background of the topic.
    - Significance of the study.
    - Objectives and scope.
18. Literature Review
    - Analysis of existing work and identification of research gaps.
19. Methodology
    - Detailed explanation of the approach, tools, and techniques used.
20. Implementation
    - Step-by-step description of the development or problem-solving process.
21. Results and Analysis
    - Presentation of findings using tables, graphs, and charts.
    - Discussion of the implications and limitations.
22. Conclusion and Future Work
    - Summary of the research.
    - Suggestions for further exploration.
23. References
    - A complete list of all cited works in the specified format.
24. Appendices (if necessary)
    - Supplementary material such as code, datasets, or detailed calculations.

NOTE: Research thesis/project with minimum 1 conferences papers. Peer reviewed research publication should be encouraged.

SUGGESTED READINGS:

1. Research papers from journals like IEEE Transactions on Neural Networks and Learning Systems, Journal of Machine Learning Research (JMLR), and Nature Machine Intelligence.
2. Online resources and documentation for tools and frameworks such as TensorFlow, PyTorch, and Scikit-learn.
3. Tutorials and courses from platforms like Coursera, edX, and Kaggle Learn to stay updated on current trends and practices in AI and ML.

SEMESTER IV (For Option B only)
AIMLDIS250220: RESEARCH THESIS/ DISSERTATION-II
(Credit 20, Total Marks=500)
(Practical = 500 Marks)

Course learning Objectives:

- Develop the ability to identify and formulate a clear, researchable problem within the domain of AI and ML.
- Design and implement a research methodology appropriate to the chosen topic.
- Demonstrate a deep understanding of core AI and ML concepts, such as neural networks, optimization, probabilistic modeling, and data analysis.
- Apply advanced AI/ML techniques to solve real-world problems in areas such as natural language processing, computer vision, reinforcement learning, or generative AI.
- Gain expertise in programming languages (e.g., Python, R) and frameworks (e.g., TensorFlow, PyTorch, Scikit-learn) for implementing AI and ML models.
- Foster a mindset for continuous learning in the rapidly evolving field of AI and ML.
- Demonstrate independence, initiative, and a systematic approach to problem-solving and research.

Course learning Outcomes: Completion of this course, students should be able to do the following:

- Gain hands-on experience in the Artificial Intelligence decision cycle.
- Perform exploratory data analysis to understand the dataset.
- Build and fine -tune models using cutting-edge AI algorithms
- Present and interpret results obtained from the models.

Dissertation Overview:

The dissertation is a cornerstone of the M.Sc. program and is a vital component of the curriculum. It is designed to provide students with the opportunity to demonstrate their ability to independently tackle challenging problems, apply advanced knowledge, and contribute original ideas to their field of study. Undertaking the dissertation with utmost seriousness is essential, as it is a key step in preparing students for real-world challenges and further academic pursuits.

The dissertation represents a significant milestone in the M.Sc. program, offering students an invaluable opportunity to showcase their research capabilities, technical skills, and creative thinking. By dedicating an entire semester to this endeavour, students will not only full-fill academic requirements but also set the foundation for future achievements in their professional or academic careers.

Dissertation Report Requirements:

Format

The Detailed Report should be between 60 to 80 pages and adhere to the format specified by the department. Key sections include:

25. Title Page
    - Project title
    - Student's name and roll number
    - Supervisor's name
    - Department and institution name
    - Date of submission
26. Abstract
    - A concise summary of the research problem, methodology, results, and conclusions.
27. Acknowledgements
    - A section to express gratitude to those who supported the project.
28. Table of Contents
    - A structured outline of the report's sections and subsections with page numbers.
29. Introduction
    - Background of the topic.
    - Significance of the study.
    - Objectives and scope.
30. Literature Review
    - Analysis of existing work and identification of research gaps.
31. Methodology
    - Detailed explanation of the approach, tools, and techniques used.
32. Implementation
    - Step-by-step description of the development or problem-solving process.
33. Results and Analysis
    - Presentation of findings using tables, graphs, and charts.
    - Discussion of the implications and limitations.
34. Conclusion and Future Work
    - Summary of the research.
    - Suggestions for further exploration.
35. References
    - A complete list of all cited works in the specified format.
36. Appendices (if necessary)
    - Supplementary material such as code, datasets, or detailed calculations.

Note: Research thesis/project with minimum 1 conferences papers. Peer reviewed research publication should be encouraged.

That isiIn each III and IVsemester students have at last one conference paper. Peer reviewed research publication should be encouraged.

SUGGESTED READINGS:

1. Research papers from journals like IEEE Transactions on Neural Networks and Learning Systems, Journal of Machine Learning Research (JMLR), and Nature Machine Intelligence.
2. Online resources and documentation for tools and frameworks such as TensorFlow, PyTorch, and Scikit-learn.
3. Tutorials and courses from platforms like Coursera, edX, and Kaggle Learn to stay updated on current trends and practices in AI and ML.

SEMESTER IV (For C Only)
AIMLDIS250720: RESEARCH THESIS/ DISSERTATION-I
(Credit 20, Total Marks=500)
(Practical = 500 Marks)

Course learning Objectives:

- Develop the ability to identify and formulate a clear, researchable problem within the domain of AI and ML.
- Design and implement a research methodology appropriate to the chosen topic.
- Demonstrate a deep understanding of core AI and ML concepts, such as neural networks, optimization, probabilistic modeling, and data analysis.
- Apply advanced AI/ML techniques to solve real-world problems in areas such as natural language processing, computer vision, reinforcement learning, or generative AI.
- Gain expertise in programming languages (e.g., Python, R) and frameworks (e.g., TensorFlow, PyTorch, Scikit-learn) for implementing AI and ML models.
- Foster a mindset for continuous learning in the rapidly evolving field of AI and ML.
- Demonstrate independence, initiative, and a systematic approach to problem-solving and research.

Course learning Outcomes: Completion of this course, students should be able to do the following:

- Gain hands-on experience in the Artificial Intelligence decision cycle.
- Perform exploratory data analysis to understand the dataset.
- Build and fine -tune models using cutting-edge AI algorithms
- Present and interpret results obtained from the models.

Dissertation Overview:

The dissertation is a cornerstone of the M.Sc. program and is a vital component of the curriculum. It is designed to provide students with the opportunity to demonstrate their ability to independently tackle challenging problems, apply advanced knowledge, and contribute original ideas to their field of study. Undertaking the dissertation with utmost seriousness is essential, as it is a key step in preparing students for real-world challenges and further academic pursuits.
The dissertation represents a significant milestone in the M.Sc. program, offering students an invaluable opportunity to showcase their research capabilities, technical skills, and creative thinking. By dedicating an entire semester to this endeavour, students will not only full-fill academic requirements but also set the foundation for future achievements in their professional or academic careers.

Dissertation Report Requirements:

Format

The Detailed Report should be between 60 to 80 pages and adhere to the format specified by the department. Key sections include:

37. Title Page
    - Project title
    - Student's name and roll number
    - Supervisor's name
    - Department and institution name
    - Date of submission
38. Abstract
    - A concise summary of the research problem, methodology, results, and conclusions.
39. Acknowledgements
    - A section to express gratitude to those who supported the project.
40. Table of Contents
    - A structured outline of the report's sections and subsections with page numbers.
41. Introduction
    - Background of the topic.
    - Significance of the study.
    - Objectives and scope.
42. Literature Review
    - Analysis of existing work and identification of research gaps.
43. Methodology
    - Detailed explanation of the approach, tools, and techniques used.
44. Implementation
    - Step-by-step description of the development or problem-solving process.
45. Results and Analysis
    - Presentation of findings using tables, graphs, and charts.
    - Discussion of the implications and limitations.
46. Conclusion and Future Work
    - Summary of the research.
    - Suggestions for further exploration.
47. References
    - A complete list of all cited works in the specified format.
48. Appendices (if necessary)
    - Supplementary material such as code, datasets, or detailed calculations.

Note: Research thesis/project with minimum 1 conferences papers. Peer reviewed research publication should be encouraged.

SUGGESTED READINGS:

1. Research papers from journals like IEEE Transactions on Neural Networks and Learning Systems, Journal of Machine Learning Research (JMLR), and Nature Machine Intelligence.
2. Online resources and documentation for tools and frameworks such as TensorFlow, PyTorch, and Scikit-learn.
3. Tutorials and courses from platforms like Coursera, edX, and Kaggle Learn to stay updated on current trends and practices in AI and ML.