# SEMESTER VII

To be effective from year-2024

# MAJOR SUBJECTS

To be effective from year-2024

## COURSE TITLE: Java Programming

| Course Code: IMDAMJJP0724 | | | | | | Examination Scheme | T | P |
|---|---|---|---|---|---|---|---|---|
| Total number of LectureHours:42<br>Total number of Practical Hours: 28 | | | | | | External | 80 | 40 |
| | | | | | | Internal | 20 | 10 |
| Lecture(L): | 3 | Practical(P): | 1 | Tutorial(T): | 0 | Total Credits | 4 | |

**Course Objectives**

- Understand the foundational concepts of Java programming, including its syntax, structure, and execution model.
- Gain proficiency in using object-oriented programming principles, such as classes, inheritance, polymorphism, and encapsulation.
- Develop skills in exception handling and implementing robust error management in Java applications.
- Learn to create and manage Java applets and understand event-driven programming.
- Explore multithreading and synchronization concepts to build concurrent applications.
- Acquire knowledge of I/O operations, including file handling and object serialization.

| Course Content | TEACHING HOURS |
|---|---|
| **Unit I: Introduction to Java Programming and Class Fundamentals** | **Hrs.** |
| **Introduction to Java Language** - Java Virtual Machine (JVM) and Byte-code. Java Language Overview: Lexical issues Installing JDK.PATH variable. Java program Structure, Compilation and Execution. Java Class libraries (System Class).main () method.<br>**Data types, Variables and Arrays:** Primitive Data-types and Typed-Literals. Variables - Declaration, Initialization, Scope and Lifetime. Arrays - Single and Multidimensional. Type Conversion and Expression Promotion, Operators, Expressions and Control statements.<br>**Class Fundamentals:** Class Structure (Variable and Method declaration). Modifiers (Access Modifiers and Other Modifiers). Components of Class, Variable and Method declaration. Constructor and finalize (). Garbage Collection. Passing parameters to methods. Variable hiding. Method overloading. Constructor overloading and chaining. Use of this keyword. Code blocks - Static and non-static. | 14 |
| **Unit II: Object-Oriented Concepts, Exception Handling, and Strings** | **Hrs.** |
| **Inheritance:** Mechanism. Role of Access Modifiers. Method Overriding and Shadowing. Use of super keyword. Polymorphism Early and late binding. Abstract Class and Interface. Components of Interface declaration. Implementing Interfaces. **Exception Handling:** Mechanism - Exception-Object, Throwing an Exception, and Exception Handler. Catch or Specify | 14 |

| | |
|---|---|
| policy. Types of Exception - Checked vs Unchecked, Built-in vs User defined. Catching an Exception try-catch-finally. Specifying an Exception throws. Manually throwing an Exception throw. Custom Exceptions. Chained Exceptions.<br><br>**Packages:** Creating and Importing Packages. CLASSPATH variable. static import. **Strings:** Mutable and Immutable Strings. Creating Strings. Operations on Strings. | |
| **Unit III: Multithreading, Event-Driven Programming, and I/O Operations** | **Hrs.** |
| **Threads:** Creating Threads in Java. Java Thread Lifecycle. Multithreading in Java: Synchronization and Inter-process communication (IPC) in Threads.<br>**Event-Driven Programming:** Java 1.1 Event Delegation Model - Source object, Event object and Listener object. Methods associated with Source, Event and Listener objects. Low-level vs Semantic events. Adapter classes, Inner classes, and Anonymous Inner classes.<br>**I/O Streams:** Byte, Character, Buffered, Data, and Object Streams. Standard Streams. **File I/O Basics:** Reading and Writing to Files. Serializing Objects. **Networking Classes and Interfaces:** TCP/IP Server Sockets in Java. | **14** |

**Textbooks**

1. H. Schildt, Java: The Complete Reference, 9th Edition, Tata McGraw Hill, 2014.

**Reference Books**

1. K. Sierra, Sun Certified Programmer For Java 5, Wiley India, 2006.
2. K. Sierra and B. Bates, Head First Java (Java 5), 2nd Edition, O'Reilly, 2003.
3. H.M. Dietel and P.J. Dietel, Java: How to Program, 6th Edition, Pearson Education, 2007.
4. C.S. Horstmann and G. Cornell, Java 2 Vol-1 Fundamentals, 7th Indian Reprint, Pearson Education, 2006.
5. E. Balagurusamy, Programming with Java: A Primer, 4th Edition, Tata Mcgraw Hill, 2010.

**Lab Manual**

**Week 1**

- Write a program in C++ to insert, delete, and update the contents of an array.
- Write a program in C++ to search an element in an array.
- Write a program in C++ to perform various operations on matrices.
- Write a program in C++ to implement different string manipulation operations?
- Write a program to search an element in array using Binary Search.
- Write a program to implement Selection sort
- Write a program to implement bubble sort

**Week 2**

- Write a program to implement insertion sort
- Write a program to implement quick sort
- Write a program to implement merge sort
- Write a program to add two sparse matrices?
- Write a program to multiply two sparse matrices?

**Week 3**

- Write a program to implement singly linked list?
- Write a program to implement different operations like adding a node at beginning, end, center, after a certain element, after a certain count of nodes in a linked-list.
- Write a program to implement different operations like deleting a node at beginning, end, center, after a certain element, after a certain count of nodes in a linkedlist.
- Write a program in C++ to reverse a linked list by changing the link in the nodes?

**Week 4**

- Write a program to add two polynomials represented as linked list?
- Write a program in C++ to multiply two polynomials represented as linked lists?
- Write a program in C++ to implement a doubly linked list?
- Write a program to implement different operations like adding a node at beginning, end, center, after a certain element, after a certain count of nodes in a doubly linkedlist.
- Write a program to implement different operations like deleting a node at beginning, end, center, after a certain element, after a certain count of nodes in a doubly linkedlist.

**Week 5**

- Write a program to implement different operations of a circular linked list.
- Write a program to implement various operations on an array based stack?
- Write a program to implement various operations on a stack represented using linked list.

**Week 6**

- Write a program to demonstrate the use of stack in checking whether the arithmetic expression is properly parenthesized?
- Write a program to demonstrate the use of stack in converting an arithmetic expression from infix to postfix?
- Write a program to demonstrate the use of stack in evaluating an arithmetic expression in postfix notation?

## Week 7

- Write a program to demonstrate the use of stack in implementing quicksort algorithm to sort an array of integers in ascending order.
- Write a program to demonstrate the implementation of various operations on a linear queue represented using a linear array
- Write a program to demonstrate the implementation of various operations on a Circular queue represented using a linear array.

## Week 8

- Write a program to demonstrate the implementation of various operations on a queue represented using a linked list?
- Write a program to demonstrate the use of multiple stacks?

## Week 9

- Write a program to delete a node in a binary search tree?
- Write a program to implement the different operations of an AVL tree
- Write a program to implement the different operations of a threaded binary tree.
- Write a program to implement the different operations of a M-way search tree?

## Week 10

- Write a program to implement the different operations of a B- tree?
- Write a program in C++ to implement the different operations of a B+tree.
- Write a program in C++ to implement the graph using different representations.

## Week 11

- Write a C++ program to illustrate the traversal of a graph using Breadth FirstSearch.
- Write a C++ program to illustrate the traversal of a graph using Depth FirstSearch.
- Write a program in C++ to find the edges of a spanning tree using Prims Algorithm.
- Write a program in C++ to find the shortest path in a graph using Warshalls Algorithm.

## Week 12

- Write a C++ program to in C++ to find the shortest path in a graph using Dijkstra's Algorithm.
- Write a C++ program in C++ to implement Euler Graphs?
- Write a program in C++ to implement Hamilton Graphs?

## Week 13

- Write a program in C++ to implement Planner Graphs?
- Write a program to C++ to implement Kruskals Algorithm?

## Week 14

- Write a C++ program to implement a simple hash table using linear probing to resolve collisions.
- Write a C++ program to create Max and Min heaps?

**COURSEOUTCOMES(CO):**

Upon completing the course, students will be able to:

**CO1:** Write and execute basic Java programs, utilizing proper syntax and structure.
**CO2:** Design and implement classes and objects, applying object-oriented principles effectively.
**CO3:** Handle exceptions and errors using Java's exception handling mechanisms.
**CO4:** Create and manipulate applets, implementing event-driven programming techniques.
**CO5:** Develop multithreaded applications, managing synchronization and inter-process communication.
**CO6:** Perform file I/O operations, including reading and writing files and serializing objects for data persistence.

**LEVELOFCO-PO MAPPINGTABLE**

| | Pos | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cos | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 1 | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | |

## COURSE TITLE: Optimization Techniques

| Course Code: IMDAMJOT0724 | | | | | | Examination Scheme | T | P |
|---|---|---|---|---|---|---|---|---|
| Total number of Lecture Hours: 56 | | | | | | External | 80 | - |
| Total number of Practical Hours: - | | | | | | Internal | 20 | - |
| Lecture(L): | 4 | Practical (P): | - | Tutorial(T): | - | Total Credits | | 4 |

**Course Objectives**

- Solve linear programming problems using various methods, including the Simplex Algorithm and Duality Theorems.
- Apply methods for solving transportation and assignment problems, including handling unbalanced cases.
- Use game theory and sequencing models to solve optimization problems and manage project scheduling.
- Address advanced operations research topics, including integer programming, network flow optimization, and multi-objective decision-making.

| Course Content | TEACHING HOURS |
|---|---|
| **UNIT 1:Linear Programming and Duality** | **14 Hrs** |
| Linear Programming Problem (LPP): Formulating LPPs, Simplex Algorithm, Big-M Method, Two Phase Method, Sensitivity Problem. Duality in LPP: Duality Theorems, Dual Simplex Method | |
| **UNIT 2: Transportation and Assignment Problems** | **14Hrs** |
| Transportation Problems: Mathematical Formulation of Transportation problem, Methods of selecting initial basic feasible solution: Matrix minima method, North-West Corner Rule, Vogel's Approximation Method; Unbalanced Transportation Problem; Degeneracy in Transportation Problem and its resolution through MODI Method (U-V Method). Assignment problems: Algorithm, Unbalanced Assignment Problem, Hungarian Method | |
| **UNIT 3:Game Theory and Sequencing Models** | **14Hrs** |

To be effective from year-2024

| | |
|---|---|
| Game theory: Definition and Terminologies; Pure Strategy: saddle point, Game with two saddle points; Mixed strategies: games without saddle points, 2Xn games, Dominance Property. Sequencing models: Sequencing of n jobs on two machines and three machines with no passing. CPM- Determination of critical tasks. PERT- probability of completing the project on schedule. | |
| **UNIT 4: Advanced Topics in Operations Research** | **14Hrs** |
| **Integer Programming and Non-linear Programming:** Formulate and solve integer programming problems using branch and bound techniques, and apply non-linear programming methods to handle constraints and objective functions with non-linearity.<br>**Network Flow Problems:** Analyze and solve network flow problems including the Max Flow Min Cut Theorem, and use algorithms like Ford-Fulkerson and Edmonds-Karp for network optimization.<br>**Multi-Objective Optimization:** Explore methods for optimizing multiple conflicting objectives, including goal programming and Pareto efficiency, and apply these methods to real-world decision-making scenarios. | |

**Textbooks**

1. N.D. Vohra "Quantitative Techniques", 2010, 2nd Edition.

**Reference Books**

1. H.A.TAHA," Operations Research". Pearson Education, 2021, 11th Edition.
2. S.D. Sharma," Operations Research & Optimization", 2021, 2nd Edition.
3. KantiSwaroop, " Operations Research and Applications, 2017, 3rd Edition
4. R. PanneerSelvam : Operations Research, 2022, 6th Edition.

**COURSEOUTCOMES(CO):**

**CO1:** Efficiently solve linear programming problems and apply duality concepts to optimize and analyze solutions.

**CO2:** Develop accurate solutions for transportation and assignment problems, addressing issues like degeneracy and unbalanced scenarios.

**CO3:** Utilize game theory and sequencing models to make informed decisions in strategic interactions and project scheduling.

**CO4:** Apply advanced techniques in integer programming, network flow problems, and multi-objective optimization to solve complex operations research challenges.

## COURSE TITLE: Deep Learning Architecture

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Course Code:** | **IMDAMJDL0724** | | | | | **Examination Scheme** | **T** | **P** |
| **Total number of Lecture Hours: 42**<br>**Total number of Practical Hours: 28** | | | | | | **External** | **80** | **-** |
| | | | | | | **Internal** | **20** | **-** |
| **Lecture(L):** | **3** | **Practical(P):** | **1** | **Tutorial(T):** | **-** | **Total Credits** | **4** | |

**Course Objectives**

- To equip students with the skills to design, implement convolutional neural networks (CNNs) for various applications.
- Apply deep learning techniques to solve computer vision tasks such as image classification, object detection, and image segmentation.
- To introduce students to advanced deep learning architectures, techniques, and challenges, including regularization, transfer learning, and neural architecture search.
- To explore cutting-edge topics in deep learning, such as Graph Neural Networks, Meta Learning, Autoencoders, Generative Adversarial Networks (GANs), and Deep Reinforcement Learning.

| Course Content | TEACHING HOURS |
|---|---|
| **UNIT I** | **14 Hrs** |
| Convolutional Network Regularization: Data Augmentation, L2, Dropout, DropConnect. Batch Normalization, Transfer Learning and Fine-Tuning Pretrained Models.<br><br>Advanced Convolutional Architectures: AlexNet, Visual Geometry Group, Residual Networks, Inception Networks: V1, V2, V3, V4 and Inception-Res<br><br>Applications of Convolutional Neural Network: Image classification, object detection, and image segmentation.<br><br>Transfer Learning and Fine Tuning, Limitations of Convolutional Networks. | |
| **Unit II** | **14 Hrs** |
| Overview of NAS: History and Motivation, significance in Deep Learning. Baseline methods and its limitations: Random and Grid search. | |

To be effective from year-2024

| | |
|---|---|
| Evolutionary based Search, Reinforcement Learning based Search, Gradient based methods-DARTS, Challenges in NAS | |
| **UNIT III** | **14 Hrs** |
| Restricted Boltzmann Machine, Auto encoders, Deep Belief Networks, Generative Adversarial Networks, Recurrent Neural Networks, GraphNeural Networks, Meta Learning- Zero shot and One shot learning.<br><br>Capsule Networks: Structure of Capsule Network ,Deep Reinforcement Learning: introduction and applications.<br><br>Current Research Trends in Deep Learning | |

**LAB MANUAL**

1. Familiarization with Deep Learning Libraries-Keras and Tensor flow
2. Demonstrate Neural Networks using different Activation Function
3. Use a deep learning framework like TensorFlow or Keras to build and train a neural network for image classification.
4. Experiment the above problem with different learning rates, batch sizes, and activation functions.
5. Apply dropout regularization to a neural network and observe its impact on overfitting.
6. Experiment with L1 and L2 regularization techniques and understand their effect on model performance.
7. Implement a basic Convolutional Neural Network for image classification.
8. Perform binary text classification on IMDb movie reviews dataset.
9. Implement a basic Generative Neural Network to generate synthetic images.
10. Implement Deep Belief Network on Handwritten Digit Classification.
11. Implement Auto encoders on dataset of your choice.
12. Use any data augmentation technique for image classification.
13. Analyze how batch normalization helps stabilize and speed up the training of CNNs.
14. Implementing dropout regularization in CNN layers.
    - Investigating the effect of dropout rates on CNN training and generalization.
15. Comparative study of different regularization techniques applied to CNN architectures.
16. Implement a pre-trained model for:
    - prediction,
    - feature extraction,
    - fine-tuning.
17. Write a program to visualize the patterns that image classification models learn.
18. Implement a deep convolutional autoencoder for image denoising, mapping noisy digits' images from the MNIST dataset to clean digit images.
19. Build a convolutional network and use it to classify images (faces, melanomas, etc.) based

on patterns and objects that appear in them. Use these networks to learn data compression and image denoising.

20. Implement VGG-Net on image classification dataset.
21. Implement residual blocks to classify CIFAR-10 DATASET
22. Implement image segmentation using U-net architecture.
23. Build a Graph Neural Network Model
    - Prepare the data for the graph model
    - Implement a graph convolution layer
    - Implement a graph neural network node classifier
    - Train the GNN model
    - Examine the GNN model predictions

**Textbooks:**


1. Deep Learning by Ian Goodfellow, MIT Press, 2016, 1st Edition
2. Advanced Deep Learning with Python by Ivan Vasilev, Packt Publishing, 2019, 2nd Edition
3. Advances in Deep Learning by M. Arif Wani, Springer, 2019, 1st Edition


**Reference Books:**


6. Deep Learning with Python, Francois Chollet, 2nd edition, 2021
7. Deep Reinforcement Learning Hands-On, MaximLapan, 2nd edition, 2020
8. Automated Machine Learning Methods, Systems, Challenges, 2019
9. Deep Learning: A Visual Approach, Andrew Glassner, 2021
10. Hands-On Deep Learning Architectures with Python, Yuxi (Hayden)Liu and Saransh Mehta.
11. Selected Journal and Conference Papers.


**COURSEOUTCOMES(CO):**

**CO1:** Apply deep learning techniques to solve computer vision tasks such as image classification, object detection, and image segmentation.
**CO2:** Apply transfer learning techniques to adapt pre-trained models to new, related tasks effectively.
**CO3:** Apply regularization techniques, design and analyze convolutional neural networks (CNNs), and explore advanced architectures.
**CO4:** Explore advanced deep learning concepts including Graph Neural Networks, Meta Learning, Restricted Boltzmann Machines, Auto encoders, and Generative Adversarial Networks (GANs).

To be effective from year-2024

**LEVELOFCO-PO MAPPINGTABLE**

| COs | POs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 1 | 3 | 2 | 2 | 2 | 2 | 1 | 1 | - | 1 | 1 | - | 2 |
| 2 | 3 | 3 | 2 | 2 | 3 | - | 1 | - | 1 | 1 | 1 | 3 |
| 3 | 3 | 3 | 3 | 3 | 3 | - | 1 | 1 | 2 | 2 | 2 | 3 |
| 4 | 3 | 3 | 3 | 3 | 3 | 2 | 1 | 2 | 2 | 2 | 1 | 3 |

## COURSE TITLE: Computer Vision

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Course Code:** | **IMDAMJCV0724** | | | | | **Examination Scheme** | |
| **Total number of Lecture Hours: 56** | | | | | | **External** | **80** |
| | | | | | | **Internal** | **20** |
| **Lecture(L):** | **4** | **Practicals (P):** | **-** | **Tutorial(T):** | **-** | **Total Credits** | **4** |

**Course Objectives**

- To provide a comprehensive introduction to the fundamental concepts and applications of computer vision.
- To understand and implement image processing techniques for analyzing digital images.
- To introduce feature detection and matching techniques, enabling object detection and recognition.
- To explore basic machine learning algorithms and their application in computer vision tasks.

| **Course Content** | **TEACHING HOURS** |
|---|---|
| **Unit 1: Introduction to Computer Vision** | **16 Hrs** |
| Introduction to Computer Vision: History and Applications. Basics of Digital Images: Image Representation, Pixels, and Color Spaces. Image Formation: Cameras, Lenses, and Image Sensors. Introduction to Image Processing: Histogram Equalization, Smoothing, and Sharpening. Image Filtering: Convolution, Edge Detection (Sobel, Canny). | |
| **Unit 2: Feature Detection and Matching** | **14Hrs** |
| Interest Point Detection: Harris Corner Detector, SIFT, SURF. Feature Descriptors: HOG, SIFT, and SURF. Feature Matching: Brute Force Matching, FLANN. RANSAC and Homography Estimation. Applications of Feature Matching in Image Stitching. | |
| **UNIT 3: Object Detection and Recognition** | **14hrs** |
| Template Matching and Object Detection. Introduction to Machine Learning for Vision: K-Nearest Neighbors, SVMs. Object Classification: Bag of Visual Words, CNNs (Convolutional Neural Networks) Basics. Region-Based Detection: R-CNN, Fast R-CNN, and YOLO (Basics). Face Detection and Recognition (Haar Cascades, Deep Learning Techniques). | |

| **UNIT 4 : Motion and 3D** | **14hrs** |
|---|---|
| Optical Flow: Lucas-Kanade, Horn-Schunck Methods. Motion Detection: Background Subtraction, Frame Differencing. Introduction to Structure from Motion (SfM). 3D Vision: Stereo Vision, Depth Estimation, and Point Clouds. | |

**Textbooks**

1. Digital Image Processing, 4th Edition (2018)" by Rafael C. Gonzalez and Richard E. Woods
2. Computer Vision: Algorithms and Applications, 1st Edition (2022)" by Richard Szeliski

**Reference Books**

1. Learning OpenCV 4: Computer Vision with Python, 3rd Edition (2019)" by Gary Bradski and Adrian Kaehler
2. Multiple View Geometry in Computer Vision, 2nd Edition (2004)" by Richard Hartley and Andrew Zisserman
3. Pattern Recognition and Machine Learning, 1st Edition (2006)" by Christopher M. Bishop

**COURSEOUTCOMES(CO):**

**CO1:** Understand and explain the core principles of computer vision and image processing.

**CO2:** Apply image processing techniques such as filtering, edge detection, and histograms to enhance images.

**CO3:** Detect and match features in images using algorithms like SIFT, SURF, and RANSAC.

**CO4:** Implement basic object detection and recognition techniques, including template matching and convolutional neural networks (CNNs).

**LEVEL OF CO-PO MAPPINGTABLE**

| | POs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **COs** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| 1 | 2 | 2 | 1 | 1 | 2 | - | - | - | - | - | - | 1 |
| 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | - | 1 | - | 1 | 2 |
| 3 | 3 | 1 | 3 | 2 | 2 | 1 | 1 | - | - | 1 | 1 | 2 |
| 4 | 2 | 2 | 2 | 2 | 1 | - | - | - | - | 2 | 2 | 1 |

To be effective from year-2024

## COURSE TITLE: Big Data Analytics

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Course Code:** | **IMDAMJBD0724** | | | | | **Examination Scheme** | |
| **Total number of Lecture Hours: 56** | | | | | | **External** | **80** |
| | | | | | | **Internal** | **20** |
| **Lecture(L):** | **4** | **Practicals (P):** | **-** | **Tutorial(T):** | **-** | **Total Credits** | **4** |

**Course Objectives**

- To introduce the fundamental concepts of Big Data and its applications.
- To explore key technologies and tools used for processing Big Data.
- To understand data storage, management, and distributed computing models.
- To provide hands-on experience with Big Data frameworks like Hadoop and Spark.
- To learn basic techniques for analyzing and visualizing Big Data.
- To understand the challenges and best practices in handling real-time data streams.

| **Course Content** | **TEACHING HOURS** |
|---|---|
| **Unit 1: Introduction to Big Data** | **17 Hrs** |
| Overview of Big Data: Definition, characteristics, and importance (Volume, Velocity, Variety, Veracity, and Value). Differences between traditional data management and Big Data. Challenges of Big Data. Introduction to Hadoop Ecosystem: HDFS, MapReduce. Introduction to NoSQL databases: Key-value stores, Document databases, Column-family stores. | |
| **Unit 2: Big Data Technologies** | **14Hrs** |
| Introduction to Hadoop: Architecture, HDFS (Hadoop Distributed File System), and MapReduce framework. Overview of Apache Spark: Architecture, RDD (Resilient Distributed Dataset), and comparison with MapReduce. Introduction to Data Warehousing: Hive, HBase. Introduction to Stream Processing: Apache Storm, Kafka. | |
| **UNIT 3: Big Data Analytics Techniques** | **14hrs** |
| Data Preprocessing for Big Data. Descriptive analytics, Predictive analytics, and Prescriptive analytics. Introduction to Machine Learning techniques for Big Data: Classification, Clustering. Tools for Big Data Analytics: R, Python, and Weka. | |

To be effective from year-2024

| UNIT 4 : Applications of Big Data | 14hrs |
|---|---|
| Big Data in Healthcare, Retail, Finance, and Social Media. Sentiment Analysis, Recommender Systems. Case Studies: How Big Data is used in companies like Google, Amazon, and Facebook. Ethical and privacy concerns with Big Data. | |

**Textbooks**

- **Big Data: Principles and Best Practices of Scalable Real-Time Data Systems" (2nd Edition, 2017)** by Nathan Marz and James Warren.
- **Big Data: Concepts, Technology, and Architecture" (1st Edition, 2021)** by Thomas Erl, Wajid Khattak, and Paul Buhler.
-

**Reference Books**

- Hadoop: The Definitive Guide" (4th Edition, 2015) by Tom White.
- Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking" (1st Edition, 2013) by Foster Provost and Tom Fawcett.

**COURSEOUTCOMES(CO):**

**CO1:** Demonstrate an understanding of Big Data concepts and challenges.

**CO2:** Apply data storage and processing techniques using distributed frameworks.

**CO3:** Use Hadoop, Spark, and other Big Data tools for large-scale data management.

**CO4:** Perform basic data analytics and visualization on Big Data.

**CO5:** Analyze real-time data streams and implement processing pipelines.

**CO6:** Evaluate and implement solutions to Big Data problems in various domains.

**LEVEL OF CO-PO MAPPINGTABLE**

| | POs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| COs | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 1 | 3 | 2 | 1 | 1 | 2 | - | - | - | - | - | - | 1 |
| 2 | 3 | 3 | 3 | 2 | 3 | 1 | 1 | - | 1 | - | 1 | 2 |
| 3 | 3 | 3 | 3 | 2 | 3 | 1 | 1 | - | - | 1 | 1 | 2 |
| 4 | 2 | 3 | 2 | 2 | 3 | - | - | - | - | 2 | 2 | 1 |
| 5 | 3 | 2 | 3 | 3 | 3 | 1 | - | - | 1 | 1 | 1 | 3 |
| 6 | 3 | 3 | 3 | 3 | 3 | 2 | 1 | 1 | 1 | 1 | 1 | 3 |

To be effective from year-2024

# SEMESTER VIII

# MAJOR SUBJECTS

To be effective from year-2024

## COURSE TITLE: Intrusion Detection Systems

| Course Code: | IMDAMJID0824 | Examination Scheme | T | P |
|---|---|---|---|---|
| Total number of Lecture Hours: 56<br>Total number of Practical Hours: - | | External | 80 | |
| | | Internal | 20 | |

| Lecture(L): | 4 | Practical(P): | 0 | Tutorial(T): | 0 | Total Credits | 4 |
|---|---|---|---|---|---|---|---|

**Course Objectives**

- Understand the fundamentals of intrusion detection systems (IDS) and intrusion prevention systems (IPS).
- Learn about different types of intrusion detection techniques, including signature-based, anomaly-based, and hybrid detection methods.
- Gain practical knowledge of deploying and managing IDS in real-world scenarios.
- Explore the use of machine learning techniques in intrusion detection

| Course Content | TEACHING HOURS |
|---|---|
| **UNIT I** | **14 Hrs** |
| Overview of Intrusion detection and its importance in cybersecurity, Types of Intrusion Detection Systems.<br>Attack trees and Correlation of Alerts, Autopsy of Worms and Botnets, Malware Detection, Obfuscation, Email/IM security issues, Viruses/ Spam, From Signatures to thumbprints to zero-day Detection, Insider Threat Issues, Masquerade and Impersonation Traitors, Decoys and Deception. | |
| **UNIT II** | **14 Hrs** |
| Basic concepts of Intrusion detection system and Intrusion Prevention System, Components of IDS: Sensors, analyzers, and user interfaces.<br>Types of IDS: Host-based IDS (HIDS), Network-based IDS (NIDS), and Hybrid IDS.<br>**Signature-Based Detection:** Fundamentals of signature-based IDS, Rule-based detection: Snort and Suricata, Pattern matching and analysis techniques. | |
| *UNIT III* | **14 Hrs** |
| **Anomaly-Based Detection:** Statistical anomaly detection techniques, Threshold-based and behaviour-based models.<br><br>**Hybrid IDS**: Combining signature and anomaly-based methods. Advantages and challenges of anomaly detection. | |

| | |
|---|---|
| Intrusion detection in cloud computing environments, IDS for Internet of Things (IoT) security, Mobile device intrusion detection. | |
| **Unit IV** | **14 Hrs** |
| Use of Artificial Intelligence (AI) and machine learning in next-gen IDS, Intrusion detection using blockchain technology, Real-time detection and response systems. Future challenges in IDS: adversarial attacks on IDS, data privacy issues. | |

**Textbooks:**

1. Intrusion detection Systems by Robert Barnard, 1st Edition, 2004

**References**


12. **Intrusion Detection and Prevention Systems: Concepts and Techniques"** by Ali A. Ghorbani, Wei Lu, and Mahbod Tavallaee, 1st Edition (2010)
13. **"Network Intrusion Detection"** by Stephen Northcutt and Judy Novak, 3rd Edition (2002)
14. **"Guide to Computer Network Security: Intrusion Detection and Prevention"** by Joseph Kizza, 3rd Edition (2015)
15. **"Computer Security: Principles and Practice"** by William Stallings and Lawrie Brown, 4th Edition (2018).


**COURSE OUTCOMES**

**CO1: Understand Core Concepts of Intrusion Detection Systems (IDS):**
Develop a foundational understanding of IDS, including its architecture, components, and the differences between IDS and Intrusion Prevention Systems (IPS).

**CO2: Apply Signature-Based and Anomaly-Based Detection Techniques:**
Gain the ability to implement and evaluate signature-based and anomaly-based detection methods to identify various types of cyber threats and intrusions.

**CO3: Utilize Advanced Machine Learning Techniques for Intrusion Detection:**
Master the application of machine learning such as clustering, classification to enhance the detection capabilities of IDS.

**CO4: Deploy and Evaluate IDS in Specialized Environments:**
Learn to effectively deploy, configure, and manage IDS in various environments, including cloud, IoT assessing their performance and limitations.

**LEVEL OF CO-PO MAPPINGTABLE**

| Cos | Pos | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 1 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 2 | 3 | 3 | 3 | 2 | 1 | 1 | 1 | 1 | 2 | 1 | 1 |
| 3 | 2 | 2 | 2 | 3 | 3 | 2 | 2 | 1 | 1 | 2 | 2 | 1 |
| 4 | 1 | 2 | 3 | 3 | 2 | 2 | 2 | 1 | 1 | 2 | 2 | 2 |

## COURSE TITLE: Generative AI

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Course Code:** | **IMDAMJGA0824** | | | | | **Examination Scheme** | |
| **Total number of Lecture Hours: 56** | | | | | | **External** | **80** |
| | | | | | | **Internal** | **20** |
| **Lecture(L):** | **4** | **Practicals (P):** | **-** | **Tutorial(T):** | **-** | **Total Credits** | **4** |

**Course Objectives**

- To introduce students to the fundamental concepts and techniques of Generative AI.
- To explore various generative models, including VAEs, GANs, and Boltzmann Machines.
- To develop an understanding of the statistical foundations of generative modeling.
- To provide practical knowledge of training, evaluating, and applying generative models in real-world scenarios.
- To familiarize students with advanced generative models and their deployment challenges.
- To raise awareness of the ethical implications and responsible use of generative AI technologies..

| Course Content | TEACHING HOURS |
|---|---|
| **Unit 1: Introduction to Generative AI** | **18 Hrs** |
| Introduction to Generative AI: What is Generative AI? Historical context and development. Applications of Generative AI **(4L)**<br><br>Generative vs. Discriminative Modeling. The rise of generative modeling, the generative modeling framework. **(5L)**<br><br>Probability and Statistics in Generative AI, Probabilistic Generative Model. Naïve bayes. The challenges of Generative modeling **(5L)** | |
| **Unit 2: Generative Models** | **14Hrs** |
| Generative Models: Overview. Types of Generative Models (e.g., Variational Autoencoders, GANs, Boltzmann Machines). Strengths and Weaknesses of each model **(5L)**<br><br>Deep Generative Models: Boltzmann Machines. Restricted Boltzmann Machines. Deep Belief Networks. **(5L)**<br><br>Variational Autoencoders (VAEs). Architecture and working principles, Training VAEs, Analysis of VAEs, Applications of VAEs **(4L)** | |

To be effective from year-2024

| **UNIT 3: Introduction to Generative Adversarial Networks (GANs)** | **14hrs** |
|---|---|
| Introduction to Generative Adversarial Network: what are GANs? How do GANs work? GAN training, reaching equilibrium. Applications of GAN **(5L)**<br><br>GAN evaluation: Inception score (IS). Fréchet inception distance (FID). GAN Training Challenges and Hacks **(4L)**<br>GAN Models: DCGAN- Architectural changes and working, Conditional GANs, WGAN **(5L)** | |
| **UNIT 4 : Advanced Topics and Future Directions in Generative AI** | **14hrs** |
| Recent Advances in Generative AI: Denoising Diffusion Probabilistic Models (DDPM), Neural Radiance Fields (NeRFs), Flow-based Models, Score-based Generative Models. **(5L)**<br>Generative AI in Real-world Applications: Healthcare, Art, Content Creation, and Video Synthesis. Challenges in deployment. **(5L)**<br>Ethical Considerations in Generative AI: Bias, Fairness, Deepfakes, and Privacy Issues. Regulation of AI, future trends, and responsible AI development. **(4L)** | |

| **Textbooks** |
|---|
| • Hands-On Generative Adversarial Networks with Keras (2019, 1st Edition) by Rafael Valle<br>• Deep Learning (2016, 1st Edition) by Ian Goodfellow, YoshuaBengio, and Aaron Courville<br>• The GAN Handbook (2020, 1st Edition) by Maximilian Jaritz |
| **Reference Books** |
| • "Probabilistic Machine Learning: Advanced Topics" (2023, 1st Edition) by Kevin P. Murphy<br>• "Neural Networks and Deep Learning" (2018, 1st Edition) by Charu C. Aggarwal |
| **COURSEOUTCOMES(CO):**<br><br>**CO1:** Demonstrate a clear understanding of the differences between generative and discriminative modeling.<br><br>**CO2:** Explain and apply core generative models like VAEs, GANs, and Boltzmann Machines in different contexts.<br><br>**CO3:** Analyze and evaluate the performance of generative models using appropriate metrics.<br><br>**CO4:** Implement and train generative models to solve real-world problems across various |

To be effective from year-2024

domains.

**CO5:** Understand the challenges of generative model training and suggest optimization techniques.

**CO6:** Assess ethical issues surrounding generative AI, such as deep fakes and bias, and propose responsible AI solutions.

**LEVELOFCO-PO MAPPINGTABLE**

| COs | POs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 1 | 3 | 2 | 1 | - | 2 | - | - | 2 | 1 | - | - | 1 |
| 2 | 3 | 3 | 2 | 2 | 3 | - | - | 2 | 2 | 1 | 1 | 2 |
| 3 | 3 | 3 | 3 | 3 | 3 | - | 1 | 1 | 1 | 2 | 1 | 2 |
| 4 | 3 | 3 | 3 | 2 | 3 | 2 | 1 | - | 1 | 2 | 2 | 2 |
| 5 | 3 | 3 | 3 | 3 | 3 | 1 | 1 | - | 2 | 2 | 2 | 2 |
| 6 | 3 | 2 | 2 | - | 2 | 3 | 3 | 3 | 2 | 2 | 1 | 1 |

## COURSE TITLE: Reinforcement Learning

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Course Code:** | **IMDAMJRL0824** | | | | | **Examination Scheme** | |
| **Total number of Lecture Hours: 56** | | | | | | **External** | **80** |
| | | | | | | **Internal** | **20** |
| **Lecture(L):** | **4** | **Practicals(P):** | **-** | **Tutorial(T):** | **-** | **Total Credits** | **4** |

**Course Objectives**

- To introduce the fundamental principles of reinforcement learning and its applications.
- To understand the key concepts of Markov Decision Processes (MDP) and their role in RL.
- To explore various RL algorithms such as dynamic programming, Monte Carlo methods, and temporal difference learning.
- To gain insight into deep reinforcement learning and its applications in real-world scenarios.
- To develop problem-solving skills in reinforcement learning through practical examples and coding exercises.

| Course Content | TEACHING HOURS |
|---|---|
| **Unit 1: Introduction to Reinforcement Learning** | **14 Hrs** |
| Overview of Reinforcement Learning (RL) and its applications. Differences between Supervised, Unsupervised, and Reinforcement Learning. Key concepts: Agent, Environment, Actions, Rewards, Policy, and Value Functions. The Markov Decision Process (MDP): states, actions, rewards, state transitions. Exploration vs. Exploitation dilemma. | |
| **Unit 2: Dynamic Programming and Monte Carlo Methods** | **14 Hrs** |
| Dynamic Programming: Policy Evaluation, Policy Iteration, and Value Iteration. Bellman Equations. Monte Carlo methods for prediction and control. Model-free vs. model-based RL. Exploration strategies: epsilon-greedy, softmax. | |
| **UNIT 3: Temporal Difference Learning** | **14 hrs** |
| TD Learning: TD(0), SARSA, and Q-learning. Eligibility traces and the TD($\lambda$) algorithm. Function Approximation: linear, nonlinear (including neural networks). Off-policy vs. on-policy learning. Q-learning with function approximation. | |

To be effective from year-2024

| **UNIT 4 : Advanced Topics in Reinforcement Learning** | **14hrs** |
|---|---|
| Deep Reinforcement Learning: Introduction to Deep Q-Networks (DQN). Policy Gradient Methods: REINFORCE algorithm. Actor-Critic methods and their variants. Applications of RL in real-world scenarios (robotics, games, etc.). Introduction to multi-agent RL. | |

**Textbooks**


1. **Sutton, R.S., &Barto, A.G.** (2018). *Reinforcement Learning: An Introduction* (2nd Edition). MIT Press.
2. **Kaelbling, L.P., & Littman, M.L.** (2022). *Introduction to Reinforcement Learning and Control* (1st Edition).
3. **Szepesvári, C.** (2010). *Algorithms for Reinforcement Learning* (1st Edition). Morgan & Claypool Publishers.


**ReferenceBooks**


1. **Kober, J., Bagnell, J.A., & Peters, J.** (2013). *Reinforcement Learning: State-of-the-Art* (1st Edition). Springer.
2. **Arulkumaran, K., Deisenroth, M.P., Brundage, M., &Bharath, A.A.** (2017). *A Brief Survey of Deep Reinforcement Learning* (1st Edition). Now Publishers.


**COURSEOUTCOMES(CO):**

**CO1:** Students will be able to understand and explain the core concepts and techniques of reinforcement learning.

**CO2:** Students will be able to formulate real-world problems as Markov Decision Processes (MDPs).

**CO3:** Students will be able to apply dynamic programming, Monte Carlo methods, and temporal difference learning to solve RL problems.

**CO4:** Students will be able to implement and experiment with deep reinforcement learning algorithms.

**CO5:** Students will gain the ability to critically analyze and evaluate reinforcement learning models and their performance in various applications.

**LEVELOFCO-PO MAPPINGTABLE**

| | POs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| COs | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 1 | 3 | 3 | 2 | 2 | 1 | - | - | - | - | 2 | - | 3 |
| 2 | 3 | 3 | 3 | 3 | 2 | - | 1 | - | 1 | 1 | - | 3 |
| 3 | 3 | 3 | 3 | 3 | 3 | - | - | - | 2 | 2 | 1 | 3 |
| 4 | 3 | 3 | 3 | 2 | 3 | 1 | - | - | 2 | 2 | 1 | 2 |
| 5 | 3 | 3 | 2 | 3 | 3 | 1 | 1 | 1 | 3 | 2 | 2 | 3 |

To be effective from year-2024

# COURSE TITLE: Natural Language Processing

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Course Code:** | | **IMDAMJNL0824** | | | | **Examination Scheme** | |
| **Total number of Lecture Hours: 56** | | | | | | **External** | **80** |
| | | | | | | **Internal** | **20** |
| **Lecture(L):** | **4** | **Practicals(P):** | **0** | **Tutorial(T):** | **0** | **Total Credits** | **4** |

**Course Objectives**

1. To develop a solid understanding of the principles and algorithms used in text and speech processing.
2. Learn and apply key natural language processing techniques, including tokenization, part-of-speech tagging, named entity recognition, and sentiment analysis.
3. Understand and implement speech recognition and synthesis technologies, including acoustic modeling, language modeling, and text-to-speech systems.
4. Gain hands-on experience with tools and libraries for text and speech processing, such as NLTK, SpaCy, and speech recognition APIs.
5. Implement and evaluate text and speech processing solutions to address real-world problems and challenges.

| **Course Content** | **TEACHING HOURS** |
|---|---|
| **UNIT 1: Introduction to Natural Language Processing** | **- Hrs** |
| History and applications of NLP. Overview of NLP tasks and challenges. Text preprocessing: tokenization, stemming, lemmatization. Part-of-speech tagging and named entity recognition. Regular expressions and text pattern matching. Speech processing: Phonetics, speech sounds and phonetic transcription, phonological categories and pronunciation variation. | 14 |
| **UNIT 2: Natural Language Processing (NLP) Techniques** | **- Hrs** |
| Text classification and sentiment analysis: Sequence labeling and parsing. Word embeddings. Feature Engineering for Text representation, Bag of Words model, Bag of N-Grams model, TF-IDF model. Named Entity Recognition (NER). | 14 |
| **UNIT 3: Speech Recognition Technologies** | **- Hrs** |
| Overview of speech recognition systems. Acoustic modeling: HMMs and DNNs. Language modeling: N-grams and neural language models. Speech Synthesis: Introduction to text-to-speech (TTS) systems. Techniques in speech synthesis: concatenative and parametric synthesis. Voice quality and prosody modeling | 14 |

| UNIT 4: Deep Learning for NLP | - Hrs |
|---|---|
| Deep learning approaches in NLP (e.g., RNNs, CNNs, LSTMs, Transformers). Machine translation and text generation. Contextual embeddings (e.g., BERT, GPT). Building a text processing and speech Recognition system using Deep Neural Networks. Python programming concept. | 14 |

**Textbooks**


1. Jurafsky, Daniel, and James H. Martin. "*Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*". 3rd ed., Pearson, 2021.
2. Chen, K. S., et al. "*Natural Language Processing with Python and spaCy: A Practical Guide to Building Intelligent Applications*". 1st ed., O'Reilly Media, 2021.
3. Liu, J., et al. "*Fundamentals of Speech Recognition*". 1st ed., Springer, 2020.


**Reference Books**


1. Allen, James. "*Natural Language Understanding*". 2nd ed., Addison-Wesley, 1995.
2. Haffner, P., et al. "*Speech Recognition: Algorithms and Applications*". 1st ed., Wiley, 2021.
3. Charniak, Eugene, and Mark Johnson. "*Speech and Language Processing*". 3rd ed., Prentice Hall, 2021.


**COURSE OUTCOMES(CO):**

**CO1**: Explain key concepts and techniques used in the analysis and processing of text and speech data.
**CO2**: Develop and apply algorithms for various NLP tasks such as tokenization, parsing, and entity recognition.
**CO3**: Implement and evaluate speech recognition and synthesis systems, including acoustic and language modeling.
**CO4**: Use programming libraries and tools for text and speech processing to perform practical analysis and synthesis tasks.
**CO5**: Apply text and speech processing techniques to real-world scenarios and evaluate their effectiveness.

**LEVEL OF CO-PO MAPPING TABLE**

| | POs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| COs | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 1 | 3 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 2 | 2 | 1 | 2 |
| 2 | 3 | 3 | 3 | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 2 | 3 |
| 3 | 3 | 3 | 3 | 3 | 3 | 2 | 1 | 1 | 2 | 3 | 2 | 3 |
| 4 | 2 | 2 | 2 | 2 | 3 | 1 | 1 | 1 | 3 | 3 | 2 | 2 |
| 5 | 3 | 3 | 3 | 3 | 2 | 3 | 2 | 2 | 3 | 3 | 2 | 3 |

To be effective from year-2024

## COURSE TITLE: Fuzzy Systems

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Course Code:** | | **IMDAMJFS0824** | | | | **Examination Scheme** | |
| **Total number of Lecture Hours: 56** | | | | | | **External** | **80** |
| | | | | | | **Internal** | **20** |
| **Lecture(L):** | **3** | **Practicals(P):** | **1** | **Tutorial(T):** | **0** | **Total Credits** | **4** |

**Course Objectives**

- To provide students with a fundamental understanding of fuzzy logic and its contrast to traditional binary logic.
- To explain the theory behind fuzzy sets and the mathematical tools used in fuzzy systems.
- To equip students with the skills to design and implement fuzzy inference systems for real-world applications.
- To explore the application of fuzzy logic in various domains such as control systems, decision making, and pattern recognition.
- To develop practical skills in evaluating and optimizing fuzzy systems for performance and accuracy.

| Course Content | TEACHING HOURS |
|---|---|
| **UNIT 1: Introduction to Fuzzy Logic** | **-Hrs** |
| History and evolution of fuzzy logic. Comparison between classical and fuzzy logic. Applications of fuzzy logic in various domains. Concept of fuzzy sets and membership functions. Operations on fuzzy sets: Union, Intersection, Complement. Types of membership functions: Triangular, Trapezoidal, And Gaussian. Fuzzy relations and their properties. Fuzzy logic operators: AND, OR, NOT. | **14** |
| **UNIT 2: Fuzzy Inference Systems** | **- Hrs** |
| Construction of fuzzy rules. Rule-based system design: Mamdani and Sugeno models. Fuzzification: Conversion of input data into fuzzy sets. Rule evaluation and aggregation. Defuzzification: Conversion of fuzzy outputs into crisp values. Design and implementation of fuzzy controllers. | **14** |
| **UNIT 3: Fuzzy System Design and Optimization** | **- Hrs** |
| Design methodologies for fuzzy systems. Parameter tuning and optimization. Techniques for evaluating the performance of fuzzy systems. Metrics: Accuracy, stability, robustness. Hybrid systems combining fuzzy logic with neural networks or genetic algorithms. Adaptive fuzzy systems. Applications of Fuzzy Logic | **14** |

To be effective from year-2024

**Textbooks**


1. Zadeh, Lotfi A., and Janos F. G. "*Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lotfi A. Zadeh*". 1st ed., World Scientific, 2021.
2. Ross, Timothy J. "*Fuzzy Logic with Engineering Applications*". 4th ed., Wiley, 2021.
3. Klir, George J., and Bo Yuan. "*Fuzzy Sets and Fuzzy Logic: Theory and Applications*". 1st ed., Prentice Hall, 2018.


**Reference Books**


1. Mendel, Jerry M. "*Uncertainty, Fuzziness, and Knowledge-Based Systems: Volume 1: Fuzzy Systems*". 1st ed., Wiley, 2020.
2. Dagli, C. H., et al. "*Fuzzy Logic and Intelligent Systems: Applications to Engineering and Business*". 1st ed., Springer, 2021.
3. Wang, L. X., and M. M. S. P. L. "*Fuzzy Systems and Fuzzy Control*". 1st ed., Wiley, 2019.


**Lab Manual-Fuzzy Systems**
**IMDAMJFS0824**

1. Installation and setup of fuzzy logic software.
2. Define triangular, trapezoidal, and Gaussian membership functions.
3. Plot membership functions and analyze their properties.
4. Apply fuzzy set operations and visualize results.
5. Hands-on with fuzzy logic simulation tools (e.g., MATLAB Fuzzy Logic Toolbox).
6. Define fuzzy rules for a basic problem (e.g., temperature control).
7. Implementing Mamdani and Sugeno fuzzy inference systems.
8. Apply fuzzification to convert crisp inputs to fuzzy values.
9. Perform defuzzification to convert fuzzy outputs to crisp values.
10. Building and testing fuzzy logic controllers.
11. Developing a fuzzy system for a chosen application area.
12. Evaluating and optimizing fuzzy systems.
13. Implement performance metrics (e.g., accuracy, stability).
14. Analyze and interpret system performance using test data.

**COURSE OUTCOMES(CO):**

**CO1:** Understand and apply the basic principles and concepts of fuzzy logic and fuzzy set theory.
**CO2:** Design and implement fuzzy inference systems using Mamdani and Sugeno methods.
**CO3:** Analyze and interpret the performance of fuzzy systems and refine them to improve accuracy and efficiency.
**CO4:** Apply fuzzy logic to solve complex problems in real-world applications such as control systems, decision-making processes, and pattern recognition.
**CO5:** Develop and present a comprehensive project that demonstrates the practical use of fuzzy logic in a selected domain.

**LEVEL OF CO-PO MAPPING TABLE**

| COs | POs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 1 | 3 | 2 | 2 | 1 | 2 | 1 | 1 | 1 | 1 | 2 | 1 | 2 |
| 2 | 3 | 2 | 3 | 2 | 3 | 2 | 1 | 1 | 2 | 2 | 1 | 2 |
| 3 | 3 | 3 | 2 | 2 | 3 | 2 | 1 | 2 | 2 | 2 | 2 | 2 |
| 4 | 3 | 2 | 3 | 2 | 2 | 2 | 1 | 2 | 2 | 3 | 2 | 3 |
| 5 | 2 | 2 | 3 | 2 | 2 | 2 | 1 | 2 | 3 | 3 | 2 | 3 |