ETERNAL UNIVERSITY

(ESTABLISHED UNDER HIMACHAL PRADESH GOVERNMENT ACT NO.3 OF 2009)

BARU SAHIB
HIMACHAL PRADESH

WORLD PEACE THROUGH VALUE BASED EDUCATION

AKAL COLLEGE OF ENGINEERING & TECHNOLOGY

M.TECH. COMPUTER SCIENCE CURRICULUM (SEMESTER I TO IV)

APPROVED VIDE ANNEXURE 4.5.6 OF 87TH ACADEMIC COUNCIL MEETING HELD ON 25TH JULY, 2025

TO BE IMPLEMENTED FROM THE ACADEMIC SESSION 2025-26

TSBaring

Dean
Academic Affairs
Eternal University
Baru Sahib (H.P.) 173101

Registrar (Officiating)
Eternal University
Baru Sahib (H.P.) 173101

ETERNAL UNIVERSITY

(Established Under Himachal Pradesh State Act No.3 of 2009)

STUDY SCHEME AND SYLLABUS

FOR

Master of Technology

in

Computer Science & Engineering (M.Tech. CSE)

2 YEAR – PG COURSE

2025-2027

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AKAL COLLEGE OF ENGINEERING & TECHNOLOGY

ETERNAL UNIVERSITY

BARU SAHIB, SIRMAUR H.P.

To be implemented from Academic Session 2025-2026

M.Tech. in Computer Science & Engineering

Under the

Department of Computer Science and Engineering

Akal College of Engineering & Technology

Eternal University, Himachal Pradesh

Major Area / Department: Computer Science and Engineering

Vision and Mission of the Department

Vision

To be an academic leader in the areas of Computer Science and Engineering, Information Technology, and other potential areas of Computer Science with worldwide recognition.

Mission

1. Provide high quality graduate educational programs in Computer Science and Engineering.
2. Contribute significantly to the research and the discovery of new knowledge and methods in computing.
3. Offer expertise, resources, and service to the community.
4. To retain the present faculty members by providing opportunities for professional Development.

Objectives of the Program

Program Educational Objectives (PEO's)

PEO-1: To impart advance theoretical and practical Knowledge, enhance skills to design, test and adapt new computing technologies for attaining professional excellence and leading successful career in industries and academia.

PEO-2: To develop the ability to critically think, analyze and offer techno-commercially feasible and socially acceptable solutions to computational problems, attaining professional excellence and carrying research. & Development (R&D) effectively.

PEO-3: To work collaboratively on development of innovative systems and optimized solutions on multidisciplinary domains and exhibit high levels of professional and ethical values within organization and society globally.

PEO-4: To develop design thinking capabilities for innovation and entrepreneurship development.

Program Specific Objectives (PSOs)

PSO-1: To understand the evolutionary changes in computing, apply standard practices and strategies to promote research and development for innovative career paths and meet future challenges.

PSO-2: The ability to incorporate contemporary and evolving computational problem-solving techniques for lifelong learning support leading to higher studies and entrepreneurship development.

PSO-3: To inculcate knowledge with moral values and professional ethics, to act as a responsible citizen.

Program Outcomes (POs)

PO1: Apply advanced knowledge of mathematics and engineering sciences to conduct independent research and solve practical problems.

PO2: Identify, formulate, and analyze engineering problems through experimentation, data interpretation, and effective technical documentation.

PO3: Demonstrate in-depth expertise in the program's specialization, exceeding undergraduate-level proficiency.

PO4: Design innovative and sustainable solutions for complex real-world problems, considering societal and environmental impact.

PO5: Function effectively as an individual or team leader in multidisciplinary software and engineering projects, applying design thinking and fostering innovation.

Major aspects of the programme

1) Theoretical foundations: This will include the mathematical background required for the subjects.

2) Application of Theory: This will include courses where the fundamentals and advanced concepts (subjects) could be implemented.
3) Thesis/Project Work: Covering the application of the concepts learned or research oriented work.

The programme can:

1. Build mathematical foundations for studying DSC. (Core subjects)
2. Once the foundations are built, it can give options to the students to choose their domain of interest (Computer Vision, Speech, Text etc.) so that they can apply the concepts learned. (Elective subjects).

Career opportunities:

This program would provide students an opportunity to learn both foundational and experimental components of DSE with application of Machine Learning and Deep Learning techniques. A student, on completion of this program, will be able to undertake industry careers involving innovation and problem-solving and join the industry as a Data Scientist/Data Analyst/Data Engineer. Along with courses that provide specialization in DSE, students will also have the option to explore some applied domains such as computer vision, natural language processing, robotics, and software analysis.

Programme Structure

Semester I					
Subject Code	**Subject(s)**	**L**	**T**	**P**	**Credit(s)**
CSE501	Advanced Data Structures and Algorithms	3	0	1	4
CSE502	Advanced Data Mining	3	0	1	4
CSE503	Mathematical Foundations for Data Science	3	1	0	4
	Elective I * *To be chosen from the list of electives.	3	0	1	4
	Elective II * *To be chosen from the list of electives.	3	0	1	4
CSE504	Seminar I	0	0	1	1
Total Credits		**15**	**1**	**5**	**21**

List of Electives (Semester I)		
Sl. No.	**Subject Code**	**Subjects**
Elective I		
1.	CSE505	Soft Computing
2.	CSE506	Natural Language Processing
3.	CSE507	Remote Sensing & Geographic Information System
Elective II		
1.	CSE508	Deep Learning
2.	CSE509	Information Retrieval
3.	CSE510	Computer Vision

Semester II					
Subject Code	**Subject(s)**	**L**	**T**	**P**	**Credit(s)**
CSE511	Machine Learning	3	0	1	4
CSE512	Big Data Analytics	3	0	1	4
CSE513	Research Methodology	3	1	0	4
	Elective III * *To be chosen from the list of electives.	3	0	1	4
	Elective IV* *To be chosen from the list of electives.	3	0	1	4
CSE514	Seminar II	0	0	1	1
Total Credits		**15**	**1**	**5**	**21**

List of Electives (Semester II)		
Sl. No.	**Subject Code**	**Subjects**
Elective III		
1.	CSE515	Reinforcement Learning
2.	CSE516	Stochastic Models and Applications
3.	CSE517	Bioinformatics
Elective IV		
1.	CSE518	Multi-Criteria Decision Making
2.	CSE519	Intrusion Detection System
3.	CSE520	Next Generation Database

Semester III					
Subject Code	**Subject(s)**	**L**	**T**	**P**	**Credit(s)**
CSE521	Statistical Modelling	2	0	2	4
CSE522	Project and Thesis - I *Students may go for industrial or inter institute collaboration, based Project work for 6 months to 1 year	0	0	6	6
CSE523	Seminar III	0	0	1	1
Total Credits		**2**	**0**	**9**	**11**

Semester IV					
Subject Code	**Subject(s)**	**L**	**T**	**P**	**Credit(s)**
CSE524	Project and Thesis - II *Students may go for industrial or inter institute collaboration, based Project work for 6 months to 1 year.	0	0	10	10
Total Credits		**0**	**0**	**10**	**10**

M.Tech CSE	Semester: I	Batch: 2025-27
Course Code: CSE501	L-T-P: 3-0-1	Credit: 4
Course Title: Advanced Data Structures and Algorithms		

Course Objectives:

1. The course is intended to provide the foundations of the practical implementation and usage of Algorithms and Data Structures.
2. One objective is to ensure that the student evolves into a competent programmer capable of designing and analyzing implementations of algorithms and data structures for different kinds of problems.
3. Another objective is to expose the student to the algorithm analysis techniques, to the theory of reductions, and to the classification of problems into complexity classes.

Course Outcomes:

CO1: Analyze and evaluate the correctness, efficiency, and complexity of algorithms using formal methods.

CO2: Implement and utilize advanced data structures and abstract data types for efficient problem-solving.

CO3: Apply various algorithm design paradigms such as brute-force, divide and conquer, and greedy techniques to develop optimal solutions.

CO4: Develop and implement solutions to real-world problems using appropriate data structures, algorithms, and search techniques.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

Course Outcomes ↓ / Outcomes →	PO 1	PO 2	PO 3	PO 4	PO 5	PSO 1	PSO 2	PSO 3
CO1: Analyze and evaluate the correctness, efficiency, and complexity of algorithms using formal methods.	**3**	**3**	**3**	**2**	**1**	**2**	**3**	**1**

CO2: Implement and utilize advanced data structures and abstract data types for efficient problem-solving.	**3**	**2**	**3**	**2**	**2**	**3**	**3**	**1**
CO3: Apply various algorithm design paradigms such as brute-force, divide and conquer, and greedy techniques to develop optimal solutions.	**3**	**2**	**3**	**3**	**2**	**3**	**3**	**1**
CO4: Develop and implement solutions to real-world problems using appropriate data structures, algorithms, and search techniques.	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**2**

3 = Strongly mapped, 2 = Moderately mapped, 1 = Slightly mapped

Detailed Syllabus

Units	Course Outlines	No. of Lectures
Unit-I	Introduction to advanced data structures, Fundamentals of the analysis of algorithms, Algorithms, Performance analysis- time complexity and space complexity, Asymptotic Notation-Big Oh, Omega and Theta notations, Complexity Analysis Examples. Data structures-Linear and nonlinear data structures, ADT concept, Linear List ADT, Recurrences: The substitution method, Recursive tree method, Masters Method, Probabilistic analysis, Amortized analysis, Randomized algorithms, Mathematical aspects, and analysis of algorithms.	11
Unit-II	Divide and Conquer technique, Binary search tree, AVL-trees, red-black trees, B and B+ trees, Finding the minimum and maximum, Merge sort, Quick sort, Strassen's matrix multiplication. Splay Trees, Binomial Heaps, Fibonacci Heaps, Application of k-D tree (k-dimensional tree) in range searches and nearest neighbor searches.	12
Unit-III	Greedy algorithms: Introduction, Knapsack problem, Job sequencing with deadlines, Minimum cost spanning trees, Kruskal's algorithm, Prim's algorithm, Optimal storage on tapes, Optimal merge pattern, Subset cover problem, Container loading or Bin packing problem.	10
Unit-IV	Dynamic algorithms: Introduction Dynamic algorithms, All pair shortest path, 0/1 knapsack, Travelling salesman	12

	problem, Coin Changing Problem, Matrix Chain Multiplication, Flow shop scheduling, Optimal binary search tree (OBST), Analysis of All problems, Introduction to NP-Hard And NP- Complete ProblemsMore algorithms: Dynamic programming, graph algorithms: DFS, BFS, topological sorting, shortest path algorithms, network flow problems.	
	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

List of Practicals:

Experiment 1 (Arrays, Linked List, Stacks, Queues, Binary Trees)

I. WAP to implement a 3 stacks of size 'm' in an array of size 'n' with all the basic operations such as IsEmpty(i), Push(i), Pop(i), IsFull(i) where 'i' denotes the stack number (1,2,3), m ≅ n/3. Stacks are not overlapping each other. Leftmost stack facing the left direction and other two stacks are facing in the right direction.

II. WAP to implement 2 overlapping queues in an array of size 'N'. There are facing in opposite direction to each other. Give IsEmpty(i), Insert(i), Delete(i) and IsFull(i) routines for ith queue

III. WAP to implement Stack ADT using Linked list with the basic operations as Create(), Is Empty(), Push(), Pop(), IsFull() with appropriate prototype to a function.

IV. WAP to implement Queue ADT using Linked list with the basic functions of Create(), IsEmpty(), Insert(), Delete() and IsFull() with suitable prototype to a functions

Experiment 2 (Sorting & Searching Techniques)

Experiment 3 (Hashing)

I. WAP to store k keys into an array of size n at the location computed using a hash function, loc = key % n, where k<=n and k takes values from [1 to m], m>n. To handle the collisions use the following collision resolution techniques, a. Linear, Quadratic, Random probing, Double hashing/rehashing, Chaining.

Experiment 4 (BST and Threaded Trees)

Experiment 5 (AVL Trees and Red,Black Trees)

Experiment 6 (B,Trees)

Experiment 7 (Min,Max Heaps, Binomial Heaps and Fibonacci Heaps)

Experiment 8 (Disjoint Sets) Experiment 9 (Graphs Algorithms)

References:

1. Cormen, Leiserson, Rivest and Stein, Introduction to algorithms (Main textbook)

2. Kleinberg and Tardos , Algorithm Design

3. Mark Weiss, Data structures and algorithm analysis in C++ (Java)

4. Aho, Hopcroft and Ullman, Data structures and algorithms

5. S. Sahni, Data Structures, Algorithms, and Applications in C++, Silicon Press

M.Tech CSE	Semester: I	Batch: 2025-27
Course Code: CSE502	L-T-P: 3-0-1	Credit: 4
Course Title: Advanced Data Mining		

Course Objectives:

The objective of this course is to introduce data warehousing and mining techniques. Application of data mining in web mining, pattern matching and cluster analysis is included to aware students of broad data mining areas.

Course Outcomes:

After completion of course, students would be:

CO1: Study of different sequential pattern algorithms

CO2: To extract patterns from time series data and its application in the real world.

CO3: Can extend the Graph mining algorithms to Web mining

CO4: Help in identifying the computing framework for Big Data

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

Course Outcomes (COs)	PO 1	PO 2	PO 3	PO 4	PO 5	PSO 1	PSO 2	PSO 3
CO1: Study of different sequential pattern algorithms	**2**	**2**	**3**	**2**	**1**	**3**	**2**	**1**
CO2: Extract patterns from time series data and apply in real world	**2**	**3**	**3**	**1**	**2**	**3**	**3**	**2**
CO3: Extend Graph mining algorithms to Web mining	**2**	**2**	**3**	**2**	**2**	**2**	**3**	**1**

CO4: Identify computing framework for Big Data	**3**	**2**	**3**	**2**	**2**	**3**	**3**	**1**

3 = Strongly mapped, 2 = Moderately mapped, 1 = Slightly mapped

Detailed Syllabus

Units	Course Outlines	No. of Lectures
Unit-I	Introduction to Data Warehousing; Introduction to Big Data; Introduction KDD and Data Mining: Mining Knowledge Representation Mining concepts, primitives, scalable methods; Data Mining and Ethics	**10**
Unit-II	Classification: Decision Trees - Decision Tables, Divide and Conquer, Calculating Information, Entropy, Pruning, Estimating Error Rates, The C4.5 Algorithm Evaluation of Learned Results- Training and Testing, Predicting Performance, Cross-Validation, Classification Rules, and prediction; Classification Rules - Inferring Rudimentary Rules, Covering Algorithms for Rule Construction, Probability Measure for Rule Evaluation, Classification of dynamic data streams, Class Imbalance Problem.	**11**
Unit-III	Cluster Analysis – Types of Data in Cluster Analysis, Partitioning methods, Hierarchical Methods; Clustering - Iterative Distance-based Clustering, Incremental Clustering, The EM Algorithm; Association Rules, Item Sets, Rules involving Relations, Rule Efficiency Frequent Patterns, Association and Correlations, Transactional Patterns and other temporal based frequent patterns;	**12**
Unit-IV	Machine Learning and Statistics, Generalization as Search, Neural Networks, Artificial Neural Networks–Perceptron's, Multilayer Networks, The Back Propagation Algorithm; Mining Time series Data, Periodicity Analysis for time related sequence data, Trend analysis, Similarity search in Time-series analysis; Mining Data Streams, Methodologies for stream data processing and stream data systems, Frequent pattern mining in stream data, Sequential Pattern Mining in Data Streams, Graph Mining; Social Network Analysis; Sequential Pattern;	**12**

	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

List of Practicals:

1. Data Warehouse Creation & OLAP Operations: Build a basic data warehouse schema (star/snowflake) and perform OLAP operations (slice, dice, roll-up, drill-down).
2. Introduction to Big Data with Hadoop/Spark: Run basic data operations using HDFS and perform a simple word count using MapReduce or PySpark.
3. Data Preprocessing for KDD: Perform missing value imputation, outlier handling, normalization, and transformation on a raw dataset (e.g., UCI data).
4. Knowledge Representation & Primitives: Apply data selection, transformation, and representation techniques; visualize primitives like tuples, patterns, and rules.
5. Ethical Data Mining: Mini case study on privacy, fairness, and bias in data mining (non-coding task; critical thinking/report writing).
6. Decision Tree Construction (ID3, C4.5 using Python/Weka): Build a decision tree on a dataset and calculate entropy, information gain. Visualize the tree structure.
7. Pruning & Evaluation Techniques: Apply reduced-error pruning and evaluate performance with accuracy, precision, recall, F1-score, ROC-AUC.
8. Classification Rule Mining (e.g., RIPPER or PART algorithms): Generate rules from classification data and evaluate them using probability-based measures.
9. Cross-Validation & Model Evaluation: Apply k-fold cross-validation on multiple classifiers and compare results with/without pruning or class imbalance correction.
10. Handling Class Imbalance: Implement SMOTE or under-sampling and evaluate classifiers with imbalanced datasets like credit card fraud.
11. Classification on Data Streams: Use stream-based classifiers (e.g., Hoeffding Tree with river library) to classify streaming datasets.
12. Partition-based Clustering (k-Means, k-Medoids): Apply and visualize clusters using synthetic and real-world datasets. Use silhouette scores for evaluation.
13. Hierarchical Clustering: Perform agglomerative and divisive clustering; visualize dendrograms.

14. EM Clustering: Apply Expectation-Maximization algorithm and compare with k-Means on the same dataset.
15. Frequent Pattern Mining using Apriori & FP-Growth: Mine frequent itemsets and generate association rules using mlxtend or Weka; evaluate rule support and confidence.
16. Temporal/Sequential Pattern Mining: Perform pattern mining on timestamped transactions (e.g., retail logs) to identify seasonal buying trends.
17. Neural Network Classification (Multilayer Perceptron): Train and evaluate MLP using scikit-learn or Keras on datasets like MNIST, IRIS, or any multi-class problem.
18. Backpropagation Algorithm Implementation: Implement a basic ANN from scratch to understand forward and backward propagation (optional use of numpy only).
19. Time Series Trend and Periodicity Analysis: Use ARIMA/Exponential Smoothing to analyze and predict trends in time-series datasets (e.g., stock or weather data).
20. Similarity Search in Time-Series Data: Perform dynamic time warping (DTW) and similarity measures on time-series data for pattern matching.
21. Data Stream Clustering or Classification: Use frameworks like MOA (Massive Online Analysis) or Python's river to cluster or classify stream data.
22. Graph-Based Pattern Mining: Mine subgraphs, detect communities or perform link prediction using libraries like networkx or igraph.
23. Social Network Analysis: Perform centrality, clustering coefficient, and shortest path analysis on datasets like Twitter, Facebook, or citation graphs.
24. Sequential Pattern Mining: Use GSP or PrefixSpan algorithms to mine frequent sequences in clickstream or purchase data.

References:

1. Data Mining for Scientific and Engineering Applications, edited by R. Grossman, C. Kamath, W. P. Ke gelmeyer, V. Kumar, and R. Namburu, Kluwer Academic Publishers, 2001. ISBN: 1-4020-0033-2.

2. Data Mining Introductory and Advanced Topics, Margaret H.Dunbam, Pearson Education 2003.

3. Data mining: Concepts and techniques (3rd ed.). Waltham: Morgan Kaufmann, Han, J., Kamber, M., & Pei, J. (2011).

4. Data Mining: Practical Machine Learning Tools and Techniques, Ian H. Witten and Eibe Frank,

5. Introduction to Data Mining, 2nd Edition, Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar, Pearson Publishing, January 2018.

6. Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Addison-Wesley ISBN: 0321321367, April 2005. Chinese translation: 2005, Korean Translation: 2007.

7. Introduction to Parallel Computing, (Second Edition) by Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar. Addison-Wesley, 2003. ISBN 0-201-64865-2. International Chinese Edition 2003, Chinese translation, China Machine press, 2004.

8. Introduction to Parallel Computing: Design and Analysis of Algorithms by Vipin Kumar, Ananth Grama, Anshul Gupta and George Karypis, Benjamin-Cummings Publishing Company, November 1993.

9. Managing Cyber Threats: Issues, Approaches and Challenges, edited by V. Kumar, J. Srivastava, and A. Lazarevic, Springer, ISBN 0-387-24226-0, May 2005. April 6, 2019 25 Vipin Kumar

10. NEXT GENERATION OF DATA MINING, edited by Hillol Kargupta, Jiawei Han, Vipin Kumar, Rajeev Motwani, and Philip Yu, Chapman & Hall/CRC - Data Mining and Knowledge Discovery Series, 2008.

11. Parallel Algorithms for Machine Intelligence and Vision edited by V. Kumar, P.S. Gopalkrishnan, and L. Kanal, Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Anuj Karpatne, Springer-Verlag, March 1990.

12. Parallel Processing for Artificial Intelligence, Volume 1, edited by Laveen Kanal, Vipin Kumar, Hiroaki Kitano and Christian B. Suttner, North-Holland, June 1994.

13. Parallel Processing for Artificial Intelligence, Volume 2, edited by Hiroaki Kitano, Vipin Kumar, and Christian B. Suttner, North-Holland, July 1994.

14. Richard O. Duda, Peter E. Hart, David G. Stork - Pattern classification (2001, Wiley)

15. Search in Artificial Intelligence, edited by Laveen Kanal and Vipin Kumar, Springer-Verlag, 1988.

16. Top-10 Algorithms in Data Mining, edited by Xindong Wu and Vipin Kumar, Chapman & Hall/CRC - Data Mining and Knowledge Discovery Series, Spring 2009.

M.Tech CSE	Semester: I	Batch: 2025-27
Course Code: CSE503	L-T-P: 3-1-0	Credit: 4
Course Title: Mathematical Foundations for Data Science		

Course Objectives:

1. To introduce students to the various Mathematical concepts to be used in ML and DS.
2. Learn the concepts of probability and Statistics.
3. Learn how to pose optimization problems.
4. Learn how to solve problems by using different algorithms.

Course Outcomes:

CO1: To acquire knowledge on various Mathematical concepts to be used in Machine Learning and Data Science.

CO2: To apply the concepts of probability and Statistics.

CO3: To solve the various problems using optimization problems.

CO4: To solve various problems in data science using different algorithms.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PSO1	PSO2	PSO3
CO1: To acquire knowledge on various Mathematical concepts to be used in Machine Learning and Data Science.	**3**	**1**	**3**	**2**	**1**	**3**	**2**	**1**
CO2: To apply the concepts of probability and Statistics.	**3**	**2**	**2**	**2**	**1**	**2**	**3**	**2**
CO3: To solve the various problems using optimization problems.	**3**	**2**	**2**	**3**	**2**	**3**	**3**	**1**
CO4: To solve various problems in data science	**2**	**3**	**3**	**3**	**3**	**3**	**3**	**1**

using different algorithms.								

3 = Strongly mapped, 2 = Moderately mapped, 1 = Slightly mapped

Detailed Syllabus

Units	Course Outlines	No. of Lectures
Unit-I	Basics of Linear Algebra: Representation of vectors; Linear dependence and independence; vector space and subspaces (definition, examples, and concepts of basis); linear transformations; range and null space; matrices associated with linear transformations; special matrices; eigenvalues and eigenvectors with applications to data problems; Least square and minimum normed solutions. Matrices in Machine Learning Algorithms: projection transformation; orthogonal decomposition; singular value decomposition; principal component analysis and linear discriminant analysis.	**11**
Unit-II	Gradient Calculus: Basic concepts of calculus: partial derivatives, gradient, directional derivatives, Jacobian, Hessian matrix. Optimization: Convex sets, convex function, and their properties, Unconstrained and Constrained Optimization, Numerical Optimization Techniques for Constrained/Unconstrained Optimization, Derivative-Free methods (Golden Section, Fibonacci Search Method, Bisecting Method), Methods using Derivatives (Newton's Method, Steepest Descent Method), Penalty Function Methods for Constrained Optimization.	**15**
Unit-III	Probability: Basic concepts of probability, conditional probability, total probability, independent events, Bayes' theorem, random variable, Moments, moment generating functions, some useful distributions, Joint distribution, conditional distribution, transformation of random variables, covariance, correlation.	**10**
Unit-IV	Statistics: Random sample, sampling techniques, statistics, sampling distributions, mixture models.	**9**
	Total Theory:	**45+30=75**

References:

1. M. P. Deisenroth, A. A. Faisal, C. S. Ong, Mathematics for Machine Learning, Cambridge University Press (1st edition)

2. S. Axler, Linear Algebra Done Right. Springer International Publishing (3rd edition) 3. J. Nocedal and S. J. Wright, Numerical Optimization. New York: Springer Science+Business Media
4. E. Kreyszig, Advanced Engineering Mathematics, John Wiley and Sons, Inc., U.K. (10th Edition)
5. R. A. Johnson, I. Miller, and J. E.Freund, "Miller & Freund's Probability and Statistics for Engineers", Prentice Hall PTR, (8th edition)
6. C. Mohan and K. Deep: "Optimization Techniques", New Age Publishers, New Delhi.

M.Tech CSE	Semester: I	Batch: 2025-27
Course Code: CSE505	L-T-P: 3-0-1	Credit: 4
Course Title: Soft Computing		

Course Objectives:

The objective of this course is to introduce the fundamental concepts and techniques of soft computing, including fuzzy logic, neural networks, genetic algorithms, and hybrid systems. It aims to equip students with the ability to model and solve complex, uncertain, and nonlinear problems using intelligent systems.

Course Outcomes: Through this course students should be able to

CO1: define soft computing techniques and their roles in building intelligent machines

CO2: compare different neural network algorithms for classification and clustering problems

CO3: apply fuzzy logic to handle uncertainty in data

CO4: examine genetic algorithm, and swarm intelligence for optimization problems

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

Course Outcomes (COs)	PO 1	PO 2	PO 3	PO 4	PO 5	PSO 1	PSO 2	PSO 3
CO1: Define soft computing techniques and their roles in building intelligent machines	**3**	**2**	**3**	**2**	**1**	**3**	**2**	**1**
CO2: Compare different neural network algorithms for classification and clustering	**3**	**3**	**3**	**2**	**2**	**3**	**3**	**1**
CO3: Apply fuzzy logic to handle uncertainty in data	**2**	**2**	**3**	**2**	**1**	**2**	**3**	**2**
CO4: Examine genetic algorithm, and swarm intelligence for optimization problems	**3**	**3**	**3**	**3**	**2**	**3**	**3**	**1**

3 = Strongly mapped, 2 = Moderately mapped, 1 = Slightly mapped

Detailed Syllabus

Units	Course Outlines	No. of Lectures
Unit-I	Introduction: Artificial intelligence, Artificial neural networks, Genetic algorithm, Swarm intelligent systems, Expert systems Neural Network Concepts: Introduction to neural networks, biological neural networks to artificial neural networks, Classification of ANNs, McCulloch-Pitts neuron model, learning rules-Hebbian, Delta, Perceptron network, Support Vector Machines	**14**
Unit-II	Neural Networks: Backpropagation neural networks, Kohonen neural network, Learning vector quantization, Adaptive resonance theory neural networks, Radial basis function neural networks	**10**
Unit-III	Fuzzy Systems: Basic definition and terminology, set-theoretic operations, Fuzzy Sets, Operations on Fuzzy Sets, Fuzzy Relations, Fuzzy Rules and Fuzzy Reasoning, Fuzzy Inference system, Fuzzy Expert Systems Genetic Algorithms: Introduction to Genetic algorithms(GA), Representation, Genetic algorithm operators-Methods of selection, crossover and mutation, Working of GA, Application of GA	**12**
Unit-IV	Hybrid Systems: Hybrid systems, Genetic algorithm based Backpropagation network, Fuzzy Backpropagation network, Neuro-fuzzy systems, Fuzzy genetic algorithms Swarm Intelligence: Swarm intelligence, Cuckoo search, flocks of birds, ant colony optimization, swarm intelligence in bees, shoals of fish	**9**
	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

List of Practicals:

Neural Networks:

Neural Networks using Python Libraries.

Creation of single and multi-layer perceptrons for pattern recognition.

Creation of Back Propagation Neural Networks for Pattern Recognition.

Creation of Radial basis function Neural Networks for pattern Recognition.

Creation of SOM network for pattern classification.

Fuzzy Systems

Fuzzy Logic using Python.

Fuzzy set, operations on fuzzy set, Fuzzification.

Fuzzy System Implementation.

ANFIS Genetic Algorithms and Optimizations

Genetic Algorithm and Optimization using Python

Implementation of GA operators

Problem solving and optimization using GA

Swarm intelligence techniques like ACO and BCO

Cuckoo Search Implementation

References:

1. PRINCIPLES OF SOFT COMPUTING by S. N. SIVANANDAM, S.N. DEEPA, WILEY

2. NEURAL NETWORKS,FUZZY LOGIC, AND GENETIC ALGORITHM SYNTHESIS AND APPLICATION BY RAJASEKARAN, S., PAI, G. A. VIJAYALAKSHMI, PRENTICE HALL

3. SOFT COMPUTING WITH MATLAB PROGRAMMING by N.P. PADHY , S. P. SIMON, OXFORD UNIVERSITY PRESS

M.Tech CSE	Semester: I	Batch: 2025-27
Course Code: CSE506	L-T-P: 3-0-1	Credit: 4
Course Title: Natural Language Processing		

Course Objectives:

To equip students with foundational and practical knowledge of natural language processing techniques, enabling them to develop applications that analyze, understand, and generate human language using computational methods.

Course Outcomes: Through this course students should be able to

CO1: Define the fundamental concepts and components of Natural Language Processing (NLP).

CO2: Illustrate grammar-based and statistical models for analyzing linguistic structures in text.

CO3: Apply syntactic and semantic analysis techniques to process and interpret natural language data.

CO4: Examine discourse processing, natural language generation, and machine translation with a focus on Indian languages.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PSO1	PSO2	PSO3
CO1: Define the fundamental concepts and components of Natural Language Processing (NLP).	**3**	**2**	**3**	**1**	**2**	**3**	**2**	**1**
CO2: Illustrate grammar-based and statistical models for analyzing linguistic structures in text.	**2**	**3**	**3**	**2**	**1**	**2**	**3**	**2**
CO3: Apply syntactic and semantic analysis techniques to process and interpret natural language data.	**3**	**3**	**3**	**3**	**2**	**3**	**3**	**1**

CO4: Examine discourse processing, natural language generation, and machine translation with a focus on Indian languages.	**2**	**2**	**3**	**3**	**2**	**2**	**2**	**1**

3 = Strongly mapped, 2 = Moderately mapped, 1 = Slightly mapped

Detailed Syllabus

Units	Course Outlines	No. of Lectures
Unit-I	Introduction: NLP introduction, origins of NLP, Language and Knowledge, The challenges of NLP, Language and Grammar, Processing Indian Languages, NLP applications, Some successful Early NLP systems, Information Retrieval Programming in Python: An introduction to programming in Python, Why Python?, The NLTK (Natural Language Toolkit)	**10**
Unit-II	Language Modeling: Introduction to Language modeling, Various Grammars-based language models, Statistical Language Model Word Level Analysis: Introduction to word level analysis, Regular Expressions, Finite State Automata, Morphological Parsing, Spelling Error Detection and Correction, Words and Word Classes, Part-of-Speech Tagging Programming in Python: Regular expression, spelling error, word level analysis	**12**
Unit-III	Syntactic Analysis: Introduction to syntactic analysis, Context-Free Grammar, Constituency, Parsing, Probabilistic Parsing, Indian Languages Semantic Analysis: Introduction to semantic analysis, Meaning Representation, Lexical Semantics, Ambiguity, Word Sense Disambiguation Programming in Python: Parsing, Syntactic Analyzer	**13**
Unit-IV	Discourse Processing: Introduction to discourse processing, Cohesion, Reference Resolution, Discourse Coherence and Structure, Programs based on Discourse processing Natural Language Generation: Introduction, Architecture of NLG Systems, Generation Tasks and Representations, Application of NLG Machine Translation: Introduction to machine translation, Problems in Machine Translation, Characteristics of Indian Languages, Machine Translation Approaches: Direct, Rule-based, Corpus-based, Semantic or Knowledge-based	**10**

	MT Systems, Translation involving Indian Languages Other Applications: Information Extraction, Automatic Text Summarization, Question-Answering System Lexical Resources: Word Net, Frame Net, Stemmers, Part-of-Speech Tagger Programming in Python: Discourse Processing	
	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

List of Practicals

1. Python Installation and NLTK Kit
2. WAP to implement Regular Expressions
3. WAP to check spelling errors.
4. WAP for word level analysis.
5. WAP a program to implement parsing.
6. WAP to implement syntactic analyzer.
7. WAP to check ambiguity in sentences.
8. WAP to implement discourse processing.
9. WAP to translate one language to another language.
10. WAP to implement rule-based machine translation.
11. WAP to implement corpus based machine translation.
12. WAP to implement stemming.
13. WAP to implement part of speech tagging.

References:

1. NLP: A PANINIAN PERSPECTIVE by AKSHAR BHARATI, VINEET CHAITANYA AND RAJEEV SANGAL, PHI Learning

2. NATURAL LANGUAGE UNDERSTANDING by JAMES ALLEN, Pearson Education India

3. NATURAL LANGUAGE PROCESSING AND INFORMATION RETRIEVAL by TANVEER SIDDIQUI, U.S. TIWARY, OXFORD UNIVERSITY PRESS

M.Tech CSE	Semester: I	Batch: 2025-27
Course Code: CSE507	L-T-P: 3-0-1	Credit: 4
Course Title: Remote Sensing & Geographic Information System		

Course Objectives:

The course aims to provide a comprehensive understanding of remote sensing principles, image processing, GIS, GNSS, and geospatial tool customization. It emphasizes theoretical concepts and hands-on applications to equip students with the skills required to analyze, process, and apply geospatial data in diverse real-world scenarios.

Course Outcomes: Through this course students should be able to
CO1: Understand the principles of remote sensing, including microwave, LiDAR, thermal, and hyperspectral technologies, and their data acquisition mechanisms.
CO2: Apply digital image processing techniques such as enhancement, classification, segmentation, and change detection for extracting meaningful information from satellite imagery.
CO3: Analyze spatial data using GIS software, spatial databases, and data processing tools for effective geospatial analysis and decision-making.
CO4: Evaluate the role of GNSS, GPS positioning, and geospatial tool customization in real-world applications including natural resource management, disaster mitigation, and planetary exploration.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PSO1	PSO2	PSO3
CO1: Understand the principles of remote sensing, including microwave, LiDAR, thermal, and hyperspectral technologies, and their data acquisition mechanisms.	**3**	**2**	**3**	**2**	**1**	**3**	**2**	**1**

CO2: Apply digital image processing techniques such as enhancement, classification, segmentation, and change detection for extracting meaningful information from satellite imagery.	**3**	**3**	**3**	**2**	**2**	**3**	**2**	**2**
CO3: Analyze spatial data using GIS software, spatial databases, and data processing tools for effective geospatial analysis and decision-making.	**3**	**3**	**3**	**3**	**2**	**3**	**3**	**1**
CO4: Evaluate the role of GNSS, GPS positioning, and geospatial tool customization in real-world applications including natural resource management, disaster mitigation, and planetary exploration.	**2**	**2**	**2**	**3**	**3**	**2**	**3**	**1**

3 = Strongly mapped, 2 = Moderately mapped, 1 = Slightly mapped

Detailed Syllabus

Units	Course Outlines	No. of Lectures
Unit-I	Image Statistics: Introduction to Image Statistics, Statistics for Image Processing, Basics of Remote Sensing: Physics of Remote Sensing, RS Data Acquisition Mechanism, Microwave and LiDAR Remote Sensing,	**12**
Unit-II	Thermal Remote Sensing, Hyperspectral Remote Sensing; Photogrammetry and Cartography: Aerial Photogrammetry, Satellite Photogrammetry, Image Matching, Terrain Analysis, Cartography Digital Image Processing: Introduction to Digital Image Processing, Image Preprocessing, Image Enhancement, Image Classification, Image Fusion and Change Detection, Image Texture and Segmentation, Hyperspectral Image Analysis,	**11**
Unit-III	Microwave & LiDAR Data Processing Geographical Information System: Introduction to Geographic Information Systems (GIS), GIS Software and Application, Geospatial Database Generation and Organisation, Spatial	**12**

	Data Analysis with GIS, Technology trends in GIS	
Unit-IV	Global Navigation Satellite System: Introduction to GNSS, Introduction to GPS Error, Sources and Positioning Types of GPS, Elements of GPS Global Navigation, Satellite System Customization of Geospatial Tools: Customisation of Geospatial Tools, GIS-Development Environment Applications of Geospatial Technologies: Operational Remote Sensing Natural Resource Management, Disaster Management, Planetary Missions	**10**
	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

List of Practicals:

1. Image Histogram and Statistical Analysis: Calculate mean, variance, standard deviation, histogram, and entropy of grayscale and multispectral images using Python (OpenCV/GDAL) or ERDAS.
2. Radiometric and Atmospheric Corrections: Apply radiometric calibration and atmospheric correction (DOS method or FLAASH) on satellite imagery.
3. Satellite Image Metadata Interpretation: Interpret metadata from Landsat/Sentinel images and compute solar elevation, acquisition time, spectral resolution, etc.
4. Introduction to LiDAR and Microwave Data: Visualize and interpret LiDAR point cloud data (LAS format) and SAR images using software like LAStools, SNAP, or QGIS.
5. Thermal Remote Sensing Analysis: Extract land surface temperature (LST) from thermal bands of Landsat/Sentinel-3 using Python or ArcGIS.
6. Hyperspectral Image Analysis: Perform band selection and vegetation index computation (e.g., NDVI, SAVI) from hyperspectral data using ENVI or Python (Spectral library).
7. Photogrammetry – 3D Terrain Mapping: Generate 3D terrain models using aerial stereo images or UAV data with software like Agisoft Metashape or Pix4D.
8. Image Enhancement Techniques: Apply contrast stretching, histogram equalization, and spatial filtering (mean, median, high-pass filters).

9. Supervised and Unsupervised Classification: Classify multispectral images using Maximum Likelihood, SVM, or k-means clustering.
10. Image Fusion and Change Detection: Perform pixel-based or feature-based image fusion and multi-temporal change detection (pre/post flood or deforestation).
11. Texture and Segmentation: Apply GLCM-based texture extraction and segmentation (thresholding, edge-based, region-growing methods).
12. SAR Image Processing: Preprocess Sentinel-1 SAR data using ESA SNAP Toolbox (orbit correction, speckle filtering, terrain correction).
13. LiDAR DEM Generation and Analysis: Create Digital Elevation Model (DEM), DSM, and extract terrain features (slope, aspect) from LiDAR data.
14. GIS Software Hands-on: Perform digitization, layer overlay, and attribute editing in QGIS or ArcGIS.
15. Geospatial Database Creation: Create a geodatabase and perform spatial queries (e.g., buffer, spatial join, clipping).
16. Spatial Analysis with GIS:Conduct proximity, network, or overlay analysis (e.g., shortest path, land suitability).
17. GPS Data Collection and Mapping: Collect ground control points (GCPs) or waypoints using handheld GPS/mobile app and import into GIS.
18. Error Analysis in GPS Positioning: Perform static vs. dynamic GPS data collection and analyze errors due to multipath, satellite geometry, etc.
19. GNSS-based Navigation Simulation: Simulate GNSS satellite positions and signal paths using online tools or MATLAB/Google Earth Pro.
20. Customization of Geospatial Tools: Build a simple GIS plugin/tool using Python in QGIS (e.g., automated buffer tool or map calculator).
21. Remote Sensing for Disaster Monitoring: Use pre/post-event satellite imagery to analyze landslides, floods, or forest fire impact.
22. Geospatial Analysis for Natural Resource Management: Conduct land use/land cover (LULC) classification and mapping for water, forest, or agriculture resource assessment.

23. Planetary Remote Sensing Application: Analyze planetary datasets (e.g., Mars Orbiter images from ISRO/NASA) using JMARS or similar open-source tools.

References:

1. A TEXT BOOK OF REMOTE SENSING, GIS AND GNSS by Sailesh Samanta:
2. Advances in Geoinformatics, Remote Sensing, and GIS by G. S. Bhunia:
3. Concepts of Cartography, Remote Sensing, and GIS by various authors:
4. Designed maps: a sourcebook for GIS users by Cynthia A. Brewer
5. Essentials of Geographic Information Systems by Open Textbook Library:
6. GIS & Remote Sensing: Recent Trends and Applications by various authors:
7. Integration of GIS and remote sensing by Victor Mesev, ed.
8. Mastering QGIS: go beyond the basics and unleash the full power of QGIS with practical step-by-step examples by Kurt Menke; Richard Smith Jr.; Luigi Pirelli; John Van Hoesen; Phillip Davis
9. QGIS: becoming a GIS power user by Anita Graser; Ben Mearns; Alex Mandel; Victor Olaya Ferrero; Alexander Bruy
10. Qualitative GIS: a mixed methods approach by Sarah Elwood; Meghan S. Cope, eds
11. Remote Sensing and Geographical Information Systems by Basudeb Bhatta:
12. Remote Sensing and GIS Applications: A Starter Guide by Scientific Publishers:
13. Remote Sensing and GIS for Ecologists by Martin Wegmann, Benjamin Leutner, and Stefan Dech:
14. Textbook of Remote Sensing and Geographical Information Systems by Kali Charan Sahu

M.Tech CSE	Semester: I	Batch: 2025-27
Course Code: CSE508	L-T-P: 3-0-1	Credit: 4
Course Title: Deep Learning		

Course Objectives:

The course aims to introduce the fundamental concepts, architectures, and algorithms of deep learning. It focuses on building, training, and evaluating neural networks for tasks such as classification, regression, and feature extraction, enabling students to apply deep learning techniques to solve real-world problems across domains.

Course Outcomes: Through this course students should be able to

CO1: Define and implement deep learning models using TensorFlow and Keras frameworks.

CO2: Apply convolutional neural networks (CNNs) for image classification and related tasks.

CO3: Analyze and optimize deep learning models using techniques like autoencoders and hyperparameter tuning.

CO4: Develop and evaluate recurrent neural networks (RNNs) for modeling and solving real-world sequential data problems.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

Course Outcomes (COs)	PO 1	PO 2	PO 3	PO 4	PO 5	PSO 1	PSO 2	PSO 3
CO1: Define and implement deep learning models using TensorFlow and Keras frameworks.	**3**	**1**	**3**	**2**	**2**	**3**	**2**	**2**
CO2: Apply convolutional neural networks (CNNs) for image classification and related tasks.	**3**	**2**	**3**	**3**	**2**	**3**	**2**	**1**
CO3: Analyze and optimize deep learning models using techniques like autoencoders and hyperparameter tuning.	**3**	**3**	**3**	**2**	**2**	**3**	**3**	**1**

CO4: Develop and evaluate recurrent neural networks (RNNs) for modeling and solving real-world sequential data problems.	**3**	**3**	**3**	**3**	**2**	**3**	**3**	**1**

3 = Strongly mapped, 2 = Moderately mapped, 1 = Slightly mapped, – = No significant mapping

Detailed Syllabus

Units	Course Outlines	No. of Lectures
Unit-I	Deep Learning, Deep Learning Frameworks and Software Libraries: Introduction to TensorFlow, Installation of TensorFlow, TensorFlow ranks and tensors, TensorFlow's computation graphs, variables in TensorFlow, tensorflow optimizers, transforming tensors as multidimensional data arrays, visualization with Tensorboard, Introduction to deep learning, Reasons to go deep, An old problem: The Vanishing Gradient, Deep Learning Platforms, Deep Learning Libraries Introduction to Keras: Keras installation, keras layers and models, implementing linear regression model, overfit and underfit, save and load model, hyperparameter tuning.	**12**
Unit-II	Deep Learning models using Keras: Image classification with keras, multi-layer Perceptron learning for classification, building text classification model Convolutional Neural Networks: Building blocks of convolutional neural networks, determining the size of the convolution output, performing a discrete convolution in 2D, subsampling, putting everything together to build a CNN, Implementing a deep convolutional neural network using TensorFlow, Transfer learning with pre-trained CNN, Data Augmentation, Image segmentation, evaluation metrics for image classification.	**11**
Unit-III	Autoencoders: Introduction to autoencoders, need for autoencoders, architecture of autoencoder, denoising autoencoders, data compression using autoencoders, variational autoencoders, sparse autoencoders Recurrent Neural Networks: Modeling sequential data, understanding the structure and flow of an RNN, computing activation in an RNN, challenges of learning long-range interactions, implementing a multilayer RNN for sequence modeling in TensorFlow, text classification with an RNN, text generation with an RNN, time series forecasting.	**10**

Unit-IV	Advanced RNN Models: LSTM units, sequence classification with LSTM, stacked LSTM for sequence classification Generative Adversarial Networks: Neural style transfer, Introduction to generative models, overview of GAN structure, discriminator, generator, building GAN, problems with GANs, CycleGAN, Adversarial FGSM	**12**
	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

List of Practicals

Program to perform different operations on tensors in TensorFlow.

1. WAP to perform text classification using keras.
2. WAP to perform regression using Keras.
3. WAP to save and load trained models in keras.
4. WAP to perform image classification using dense layers.
5. WAP to identify and avoid underfitting and overfitting in DNN and improving model performance using hyperparameter tuning.
6. WAP to implement image classification using CNN and evaluate the performance of the model.
7. WAP to perform transfer learning and fine tuning.
8. WAP to perform data augmentation.
9. WAP to perform image denoising using autoencoders.
10. WAP for anomaly detection using autoencoders.
11. WAP to perform text classification using RNN.
12. WAP to implement Generative Adversarial Networks (GANS).

References

1. DEEP LEARNING by AMIT KUMAR DAS, Pearson Education India
2. ADVANCED DEEP LEARNING WITH TENSORFLOW 2 AND KERAS - SECOND EDITION by ROWEL ATIENZA, PACKT PUBLISHING

M.Tech CSE	Semester: I	Batch: 2025-27
Course Code: CSE509	L-T-P: 3-0-1	Credit: 4
Course Title: Information Retrieval		

Course Objectives:

The course aims to provide foundational and advanced knowledge of information retrieval systems, covering indexing, ranking, retrieval models, evaluation metrics, and web search. It equips students with the skills to design, implement, and evaluate search engines and understand the challenges in retrieving relevant information from large-scale datasets.

Course Outcomes: Through this course students should be able to

CO1: Explain the fundamental principles of information retrieval systems and the architecture of modern search engines.

CO2: Apply various indexing, retrieval, and ranking algorithms across classical and probabilistic IR models.

CO3: Evaluate retrieval system performance using standard metrics such as precision, recall, and F-measure.

CO4: Design and implement IR-based applications incorporating techniques like relevance feedback, web crawling, link analysis, and text mining using real-world datasets.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

Course Outcomes (COs)	PO 1	PO 2	PO 3	PO 4	PO 5	PSO 1	PSO 2	PSO 3
CO1: Explain the fundamental principles of IR systems and search engine architecture.	**3**	**2**	**3**	**2**	**1**	**2**	**2**	**1**
CO2: Apply various indexing, retrieval, and ranking algorithms.	**3**	**3**	**3**	**3**	**2**	**3**	**3**	**1**
CO3: Evaluate retrieval system performance using standard metrics.	**3**	**3**	**3**	**2**	**1**	**3**	**2**	**2**
CO4: Design and implement IR-based applications with real-world data.	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**1**

3 = Strongly mapped, 2 = Moderately mapped, 1 = Slightly mapped

Detailed Syllabus

Units	Course Outlines	No. of Lectures
Unit-I	Introduction to Information Retrieval: Definition, IR System Architecture and Workflow, History and Evolution of IR, Information vs. Data Retrieval, Boolean Retrieval Model,Term Vocabulary and Document Statistics,Zipf's Law, Heaps' Law, Preprocessing: Tokenization, Stopword Removal, Stemming and Lemmatization, Indexing: Inverted Index, Positional Index	**11**
Unit-II	IR Models and Ranking: Vector Space Model (VSM): Term Frequency and tf-idf, Cosine Similarity, Probabilistic IR Model: Binary Independence Model, Language Models for IR (LMIR), Query Likelihood Model, BM25 Ranking Algorithm, Evaluation of IR Systems: Precision, Recall, F-Measure, MAP, NDCG, Relevance Feedback and Query Expansion	**12**
Unit-III	Text Categorization, Clustering & Web Search: Text Classification using Naive Bayes, Rocchio, kNN, SVM Clustering: Flat (k-Means) and Hierarchical Agglomerative and Divisive Methods, Introduction to Web Search , Web Crawling: Architecture, Frontier Management, URL Normalization, Duplicate Detection, Link Analysis: PageRank, HITS Algorithm	**12**
Unit-IV	Advanced IR Topics: Personalization and Recommender Systems (Collaborative and Content-based), Latent Semantic Indexing (LSI) and SVD, Word Embeddings in IR: Word2Vec, BERT-based Retrieval, Search over Social Media and Microblogs, Semantic Search and Knowledge Graphs, Multilingual and Cross-lingual IR, IR Applications in Question Answering, Digital Libraries, and E-commerce	**10**
	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

List of Practicals:

1. Text Preprocessing: Tokenization, stopword removal, stemming, lemmatization using Python/NLTK.

2. Building an Inverted Index: Implement indexing for a collection of plain text documents.
3. Boolean and Vector Space Retrieval: Search documents using Boolean and tf-idf models with cosine similarity.
4. Implementing Document Ranking: Use BM25 or VSM to rank documents for a given query.
5. Evaluating Retrieval Models: Calculate Precision, Recall, F1-score, MAP, and plot Precision-Recall curves.
6. Text Classification: Implement Naive Bayes and k-NN for document categorization using a dataset (e.g., 20 Newsgroups).
7. Document Clustering: Apply k-Means or Hierarchical clustering to group similar documents.
8. Web Crawling: Build a simple focused crawler using Python and extract metadata from web pages.
9. PageRank Algorithm: Compute PageRank scores for a small web graph using matrix operations.
10. Recommender System: Create a basic collaborative filtering or content-based recommender (e.g., movie/book dataset).
11. Semantic Search: Use word embeddings (e.g., Word2Vec/BERT) to perform query-document similarity matching.

References:

1. Introduction to Information Retrieval, Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, Cambridge University Press
2. Modern Information Retrieval: The Concepts and Technology behind Search, Ricardo Baeza-Yates and Berthier Ribeiro-Neto, Addison-Wesley

M.Tech CSE	Semester: I	Batch: 2025-27
Course Code: CSE510	L-T-P: 3-0-1	Credit: 4
Course Title: Computer Vision		

Course Objectives:

The course aims to introduce the fundamental principles and techniques of computer vision, enabling students to understand, analyze, and implement algorithms for interpreting visual data. Emphasis is placed on image formation, feature extraction, object detection, motion analysis, and real-world applications such as face recognition and autonomous systems.

Course Outcomes: Through this course students should be able to

CO1: Explain the fundamentals of computational image formation and visual perception.

CO2: Apply core computer vision techniques for feature extraction, segmentation, and object recognition.

CO3: Implement geometric transformations and 3D reconstruction in visual systems.

CO4: Utilize machine learning and deep learning methods to develop and deploy real-world computer vision applications.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

Course Outcomes (COs)	PO 1	PO 2	PO 3	PO 4	PO 5	PSO 1	PSO 2	PSO 3
CO1: Explain the fundamentals of computational image formation and visual perception.	**3**	**2**	**3**	**2**	**1**	**2**	**2**	**1**
CO2: Apply core computer vision techniques for feature extraction, segmentation, and object recognition.	**3**	**3**	**3**	**3**	**2**	**3**	**2**	**1**
CO3: Implement geometric transformations and 3D reconstruction in visual systems.	**3**	**3**	**3**	**3**	**2**	**3**	**2**	**1**
CO4: Utilize machine learning and deep learning methods to develop and	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**2**

deploy real-world computer vision applications.								

3 = Strongly mapped, 2 = Moderately mapped, 1 = Slightly mapped

Detailed Syllabus

Units	Course Outlines	No. of Lectures
Unit-I	Introduction and Image Formation Human vision vs. Computer vision, Applications of computer vision, Image formation: Pinhole camera model, radiometry, image sensing, and acquisition, Color spaces: RGB, HSV, YCbCr, Image transformations: Translation, Rotation, Scaling, Affine and Perspective transformations, Image Filtering: Smoothing, Sharpening, Edge Detection (Sobel, Laplacian, Canny), Image Histograms and Thresholding	**11**
Unit-II	Features and Segmentation Interest point detection: Harris and FAST corner detectors, Blob detection: LoG, DoG, Feature descriptors: SIFT, SURF, ORB, Image segmentation: Region growing, Watershed, Graph-based methods, Contour detection and shape analysis, Feature matching and tracking: Optical flow (Lucas-Kanade), Kalman Filter, Stereo Vision and Epipolar Geometry	**11**
Unit-III	Object Recognition and 3D Vision Object recognition pipelines, Bag of visual words, Template matching, HOG (Histogram of Oriented Gradients), Classification using SVM/k-NN, Structure from motion, 3D reconstruction from multiple views, Depth estimation: Stereo correspondence, Camera calibration and pose estimation (PnP problem)	**12**
Unit-IV	Deep Learning for Vision & Applications Introduction to CNNs for image recognition, CNN architectures: LeNet, AlexNet, VGG, ResNet, Object Detection: R-CNN, Fast R-CNN, YOLO, SSD, Semantic and Instance Segmentation (U-Net, Mask R-CNN), Real-time face detection & recognition (Haar cascades, Dlib), Applications: Autonomous vehicles, medical imaging, surveillance, augmented reality	**12**
	Practicals:	**30**

	Total Theory+Practicals:	**45+30=75**

List of Practicals:

1. Image Preprocessing: Apply filters (blur, sharpen), color conversions, and histogram equalization using OpenCV.
2. Edge and Corner Detection: Implement Sobel, Canny edge detectors and Harris corner detection.
3. Feature Extraction and Matching: Extract and match features using SIFT, ORB; draw keypoints and matched pairs.
4. Image Segmentation: Apply watershed, k-means clustering, and graph-based segmentation methods.
5. Object Detection Using Traditional Methods: Use template matching, HOG + SVM, and Haar cascades.
6. Camera Calibration: Calibrate a camera using a chessboard pattern and compute intrinsic/extrinsic parameters.
7. Stereo Matching and Depth Estimation: Use stereo images to compute disparity maps and estimate depth.
8. Optical Flow and Object Tracking: Implement Lucas-Kanade optical flow and track motion across frames.
9. 3D Reconstruction: Reconstruct simple 3D scenes from two or more 2D images using structure from motion.
10. CNN for Image Classification: Build and train a CNN using TensorFlow or PyTorch on datasets like MNIST or CIFAR-10.
11. YOLO or SSD for Real-time Object Detection: Implement pre-trained object detection models and test on custom images or webcam feed.
12. Face Recognition System: Build a face detection and recognition pipeline using face_recognition or Dlib libraries.

References:

1. Computer Vision: Algorithms and Applications, Richard Szeliski, Springer
2. Digital Image Processing, Rafael C. Gonzalez and Richard E. Woods, Pearson

M.Tech CSE	Semester: II	Batch: 2025-27
Course Code: CSE511	L-T-P: 3-0-1	Credit: 4
Course Title: Machine Learning		

Course Objectives:

The course aims to introduce the foundational concepts, algorithms, and techniques of Machine Learning. It equips students with the ability to design, implement, and evaluate models for classification, regression, clustering, and dimensionality reduction using real-world data and modern tools.

Course Outcomes: Through this course students should be able to

CO1: Explain the mathematical foundations and key concepts underlying various machine learning algorithms.

CO2: Apply appropriate supervised and unsupervised learning techniques to solve practical problems.

CO3: Evaluate and validate machine learning models using relevant metrics and cross-validation methods.

CO4: Optimize and deploy machine learning solutions using model selection, hyperparameter tuning, and open-source tools.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

Course Outcomes (COs)	PO 1	PO 2	PO 3	PO 4	PO 5	PSO 1	PSO 2	PSO 3
CO1: Explain the mathematical foundations and key concepts underlying various machine learning algorithms.	**3**	**2**	**3**	**2**	**1**	**2**	**2**	**1**
CO2: Apply appropriate supervised and unsupervised learning techniques to solve practical problems.	**2**	**3**	**3**	**3**	**2**	**3**	**2**	**1**

CO3: Evaluate and validate machine learning models using relevant metrics and cross-validation methods.	**3**	**3**	**2**	**2**	**1**	**2**	**3**	**2**
CO4: Optimize and deploy machine learning solutions using model selection, hyperparameter tuning, and open-source tools.	**2**	**2**	**3**	**3**	**3**	**3**	**3**	**1**

3 = Strongly mapped, 2 = Moderately mapped, 1 = Slightly mapped

Detailed Syllabus

Units	Course Outlines	No. of Lectures
Unit-I	Foundations of Machine Learning: Introduction to ML: Definitions, types of ML (supervised, unsupervised, reinforcement), ML applications in industry, research, and society, Mathematical foundations: Linear algebra, probability, statistics, optimization basics, Data preprocessing: Cleaning, normalization, encoding, feature engineering, Model evaluation: Cross-validation, bias-variance trade-off, overfitting vs underfitting	**10**
Unit-II	Supervised Learning: Linear Regression: Least squares, gradient descent, Logistic Regression and classification, Decision Trees and Random Forests, Support Vector Machines (SVMs): Hard/soft margin, kernel trick, k-Nearest Neighbors (k-NN), Model evaluation: Confusion matrix, accuracy, precision, recall, F1-score, ROC, AUC	**10**
Unit-III	Unsupervised Learning and Clustering: Clustering: k-Means, Hierarchical clustering, DBSCAN, Dimensionality reduction: PCA, t-SNE, LDA, Association Rule Mining: Apriori, FP-Growth, Gaussian Mixture Models (GMMs) and Expectation-Maximization, Anomaly Detection Techniques, Introduction to Ensemble Methods: Bagging, Boosting, Stacking	**12**
Unit-IV	Advanced Topics and Recent Trends: Neural Networks: Perceptron, Multilayer Perceptron, Backpropagation, Deep Learning Basics: CNNs, RNNs, Autoencoders (basic intro only), Reinforcement Learning: Markov Decision Process, Q-learning (conceptual overview) Model Selection and Hyperparameter Tuning: Grid Search, Random Search, Bayesian Optimization, Explainable AI (XAI) and Ethical Aspects of ML, ML Tools and Libraries: Scikit-learn, TensorFlow, PyTorch, Keras	**13**

	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

List of Practicals:

1. Data Preprocessing: Handle missing data, categorical features, outlier detection, and feature scaling.
2. Implement Linear and Logistic Regression: From scratch and using scikit-learn on real-world datasets.
3. Classification Using Decision Trees and SVM: Train and compare models using accuracy, confusion matrix, and ROC curves.
4. k-NN for Classification: Implement k-NN and tune the value of k using cross-validation.
5. Clustering Using k-Means and Hierarchical Clustering: Apply clustering to unlabelled datasets and visualize results.
6. Dimensionality Reduction with PCA: Use PCA for visualization and noise reduction.
7. Market Basket Analysis: Generate association rules using Apriori algorithm (e.g., groceries dataset).
8. Ensemble Methods: Implement Random Forests, AdaBoost, or Gradient Boosting and compare them with single models.
9. ANN for Classification: Build a basic neural network using Keras or PyTorch.
10. Hyperparameter Tuning: Perform grid search or random search for tuning model parameters.
11. Model Evaluation: Apply cross-validation and compute various performance metrics.
12. Mini Project: Build and present an end-to-end ML solution (e.g., predictive modeling, sentiment analysis, fraud detection).

References:

1. Pattern Recognition and Machine Learning, Christopher M. Bishop, Springer
2. Machine Learning: A Probabilistic Perspective, Kevin P. Murphy, MIT Press

M.Tech CSE	Semester: II	Batch: 2025-27
Course Code: CSE512	L-T-P: 3-0-1	Credit: 4
Course Title: Big Data Analytics		

Course Objectives:

The course aims to introduce students to the fundamentals and advanced techniques of big data analytics. It focuses on data storage, processing, and analysis frameworks such as Hadoop and Spark, enabling students to derive meaningful insights from large-scale datasets for informed decision-making across various domains.

Course Outcomes: Through this course students should be able to

CO1: Describe the architecture, components, and challenges of big data ecosystems and technologies.

CO2: Utilize Hadoop, Spark, and related frameworks for distributed data processing and analytics.

CO3: Design and implement algorithms for batch and real-time data analytics on large-scale datasets.

CO4: Integrate and manage diverse data sources using NoSQL databases and real-time streaming tools to build complete big data applications.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

Course Outcomes (COs)	PO 1	PO 2	PO 3	PO 4	PO 5	PSO 1	PSO 2	PSO 3
CO1: Describe the architecture, components, and challenges of big data ecosystems and technologies.	**2**	**2**	**3**	**1**	**1**	**3**	**2**	**1**
CO2: Utilize Hadoop, Spark, and related frameworks for distributed data processing and analytics.	**2**	**2**	**3**	**2**	**2**	**3**	**3**	**2**
CO3: Design and implement algorithms for batch and real-time data analytics on large-scale datasets.	**3**	**3**	**3**	**3**	**2**	**3**	**3**	**2**

CO4: Integrate and manage diverse data sources using NoSQL databases and real-time streaming tools to build complete big data applications.	**3**	**3**	**3**	**2**	**2**	**3**	**3**	**1**

3 = Strongly mapped, 2 = Moderately mapped, 1 = Slightly mapped

Detailed Syllabus

Units	Course Outlines	No. of Lectures
Unit-I	Introduction to Big Data and Hadoop Ecosystem Definition and characteristics of Big Data (Volume, Variety, Velocity, Veracity, Value), Challenges of big data processing Big Data vs. traditional data, Introduction to Hadoop ecosystem: HDFS, MapReduce, YARN, Hadoop architecture and components, Data ingestion tools: Flume, Sqoop, File formats: Avro, Parquet, ORC	**11**
Unit-II	Distributed Data Processing and NoSQL Databases MapReduce programming model: Word count, sorting, joins Advanced MapReduce concepts: Combiner, Partitioner, Counters, Introduction to NoSQL databases: Characteristics, CAP theorem, Key-value stores: HBase, Cassandra, Document stores: MongoDB, Graph databases: Neo4j, Querying NoSQL data	**10**
Unit-III	Apache Spark and Real-Time Processing Spark architecture and RDD fundamentals, Spark transformations and actions, Spark SQL and DataFrames, Machine learning with Spark MLlib, Introduction to Spark Streaming, Kafka-based real-time data processing, Lambda and Kappa architecture	**12**
Unit-IV	Big Data Analytics & Applications Data analytics lifecycle, Text analytics and sentiment analysis, Social media analytics, Recommender systems (Collaborative filtering, Content-based filtering), Big data in healthcare, finance, and smart cities, Data visualization with Tableau / Power BI / matplotlib, Big data privacy and ethics	**12**
	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

List of Practicals:

1. Setup and Run Hadoop in Pseudo-Distributed Mode: Install Hadoop, configure HDFS, and run HDFS file operations.
2. Word Count using MapReduce: Implement MapReduce jobs for word frequency analysis.
3. Data Ingestion Using Sqoop and Flume: Import data from MySQL to HDFS using Sqoop and stream data with Flume.
4. Working with HBase and MongoDB: Store, retrieve, and query data using NoSQL databases.
5. Spark RDD and DataFrame Operations: Perform transformations and actions using PySpark.
6. Spark SQL: Run SQL queries on structured data using Spark SQL.
7. Spark MLlib: Build and evaluate a machine learning model (e.g., classification or clustering).
8. Real-time Data Processing with Kafka and Spark Streaming: Consume streaming data from Kafka and process it using Spark Streaming.
9. Social Media Data Analytics: Collect tweets using Twitter API and analyze sentiment using text analytics.
10. Data Visualization: Visualize data insights using tools like Tableau, matplotlib, or Power BI.
11. Mini Project: End-to-end big data application involving data ingestion, processing, analytics, and visualization.

References:

1. Big Data: Principles and Best Practices of Scalable Real-Time Data Systems, Nathan Marz and James Warren, Manning Publications
2. Mining of Massive Datasets, Jure Leskovec, Anand Rajaraman, Jeffrey Ullman, Cambridge University Press

M.Tech CSE	Semester: II	Batch: 2025-27
Course Code: CSE513	L-T-P: 3-0-1	Credit: 4
Course Title: Research Methodology		

Course Objectives:

The course aims to equip students with the fundamental principles and techniques of scientific research. It focuses on research design, data collection methods, statistical analysis, and ethical practices to enable students to conduct independent research, critically evaluate literature, and effectively communicate their findings.

Course Outcomes: Through this course students should be able to

CO1: Formulate relevant research problems and objectives within their domain of expertise.

CO2: Apply suitable research methodologies and experimental designs to address engineering and scientific challenges.

CO3: Critically analyze and synthesize existing literature to build a strong foundation for research.

CO4: Develop well-structured technical documents, including research proposals and scholarly articles, adhering to ethical standards and intellectual property guidelines.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

Course Outcomes (COs)	PO 1	PO 2	PO 3	PO 4	PO 5	PSO 1	PSO 2	PSO 3
CO1: Formulate relevant research problems and objectives within their domain.	**3**	**2**	**3**	**2**	**1**	**3**	**2**	**2**
CO2: Apply suitable research methodologies and experimental designs.	**3**	**3**	**2**	**3**	**2**	**2**	**3**	**1**

CO3: Critically analyze and synthesize literature to build a strong research base.	**3**	**2**	**3**	**2**	**1**	**3**	**2**	**1**
CO4: Develop technical documents adhering to ethical and IPR guidelines.	**2**	**3**	**2**	**2**	**3**	**2**	**2**	**3**

3 = Strongly mapped, 2 = Moderately mapped, 1 = Slightly mapped

Detailed Syllabus

Units	Course Outlines	No. of Lectures
Unit-I	Fundamentals of Research Meaning, objectives, and importance of research, Types of research: Fundamental, applied, qualitative, quantitative, exploratory, empirical, Research process and research cycle, Research problem formulation: Criteria, sources, and scope, Hypothesis: Types, formulation, and testing	**11**
Unit-II	Research Design and Data Collection Research design: Descriptive, experimental, exploratory, Sampling techniques: Probability and non-probability, Data collection methods: Observation, questionnaire, interview, survey, Scaling and measurement techniques, Reliability and validity of instruments, Case studies in computing and engineering research	**12**
Unit-III	Data Analysis and Interpretation Statistical techniques for data analysis (mean, median, mode, standard deviation, correlation), Hypothesis testing: t-test, chi-square test, ANOVA, Regression and correlation analysis, Use of statistical tools: SPSS, R, MATLAB, Python (NumPy/Pandas), Interpretation of results and decision making, Visualization techniques: Graphs, charts, and plots	**12**
Unit-IV	Technical Writing, Ethics and Research Tools Technical writing: Thesis, dissertation, journal papers, and research proposals, Structure and formatting styles (IEEE, ACM, APA, MLA), Plagiarism and tools for detection (Turnitin, Grammarly, PlagScan), Research ethics, copyright, patents, and IPR, Literature review and citation management tools (Zotero, Mendeley, EndNote), Identifying suitable journals and conferences, Overview of research metrics: h-index, impact factor, citation indices	**11**

	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

List of Practicals:

1. Problem Identification & Problem Statement Preparation: Identify a real-world research problem and draft a clear problem statement.
2. Literature Survey using Scopus/IEEE Xplore/Google Scholar: Perform a systematic review and summarize at least 5–10 recent papers.
3. Bibliography & Citation Management using Mendeley/Zotero: Create a reference library and generate in-text citations and bibliography.
4. Research Paper Review: Critically review one peer-reviewed research paper and evaluate its contributions.
5. Hypothesis Formulation: Frame null and alternative hypotheses for a selected research question.
6. Design a Questionnaire/Survey: Develop a survey using Google Forms or similar tools and collect sample responses.
7. Data Analysis using MS Excel / Python / R: Perform statistical analysis such as mean, standard deviation, t-test, ANOVA.
8. Plagiarism Checking using Turnitin or Similar Tool: Submit a report and check for plagiarism, then correct and resubmit.
9. Writing a Research Proposal: Prepare a mini-proposal with objectives, methodology, and expected outcomes.
10. Presentation on Research Ethics and IPR: Deliver a group seminar on ethical issues and IPR in research.

References:

1. Research Methodology: Methods and Techniques, C.R. Kothari and Gaurav Garg, New Age International Publishers
2. Research Methodology: A Step-by-Step Guide for Beginners, Ranjit Kumar, SAGE Publications

M.Tech CSE	Semester: II	Batch: 2025-27
Course Code: CSE515	L-T-P: 3-0-1	Credit: 4
Course Title: Reinforcement Learning		

Course Objectives:

This course aims to introduce the foundational concepts and algorithms of Reinforcement Learning (RL), focusing on how intelligent agents learn optimal behavior through interaction with an environment. It equips students with theoretical understanding and practical skills to model, implement, and evaluate RL solutions for real-world decision-making problems.

Course Outcomes: Through this course students should be able to

CO1: Explain the foundational principles of reinforcement learning, including Markov Decision Processes and exploration-exploitation strategies.

CO2: Implement and apply value-based learning algorithms such as Dynamic Programming, Monte Carlo, and Temporal Difference methods.

CO3: Utilize policy-based and actor-critic methods for solving reinforcement learning problems in continuous and complex environments.

CO4: Design, develop, and evaluate RL agents using frameworks like OpenAI Gym and PyTorch for both simulated and practical applications.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

Course Outcomes (COs)	PO 1	PO 2	PO 3	PO 4	PO 5	PSO 1	PSO 2	PSO 3
CO1: Explain the foundational principles of reinforcement learning, including Markov Decision Processes and exploration-exploitation strategies.	**3**	**2**	**3**	**2**	**1**	**3**	**2**	**1**
CO2: Implement and apply value-based learning algorithms such as Dynamic Programming, Monte Carlo, and Temporal Difference methods.	**3**	**3**	**3**	**2**	**1**	**3**	**2**	**1**

CO3: Utilize policy-based and actor-critic methods for solving reinforcement learning problems in continuous and complex environments.	**3**	**3**	**3**	**3**	**2**	**3**	**3**	**2**
CO4: Design, develop, and evaluate RL agents using frameworks like OpenAI Gym and PyTorch for both simulated and practical applications.	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**1**

3 = Strongly mapped, 2 = Moderately mapped, 1 = Slightly mapped

Detailed Syllabus

Units	Course Outlines	No. of Lectures
Unit-I	Foundations of Reinforcement Learning Introduction to RL and its relation to supervised/unsupervised learning, Elements of RL: Agent, Environment, State, Action, Reward, Goal of RL: Maximize cumulative reward, Markov Decision Process (MDP): States, actions, transitions, rewards, policies, Finite MDPs and Bellman equations, Types of RL tasks: Episodic vs. Continuing, Exploration vs. Exploitation	**13**
Unit-II	Dynamic Programming, Monte Carlo and TD Learning Dynamic Programming (DP): Policy evaluation, policy improvement, policy iteration, value iteration, Monte Carlo (MC) Methods: First-visit and every-visit MC prediction, control, Temporal-Difference (TD) Learning: TD(0), SARSA, Q-learning, n-step bootstrapping, Eligibility traces, TD(λ), Comparison of DP, MC, and TD methods	**11**
Unit-III	Function Approximation and Deep RL Need for function approximation in large state spaces, Linear and non-linear approximators, Deep Q-Networks (DQN): Experience replay, target networks, Double DQN, Dueling DQN, Introduction to deep RL frameworks: PyTorch, TensorFlow, Keras, Training and debugging deep RL agents	**12**
Unit-IV	Policy Gradient Methods and Advanced Topics Policy gradient theorem, REINFORCE algorithm, Actor-Critic methods, Proximal Policy Optimization (PPO), Trust Region Policy Optimization (TRPO), A3C,	**10**

	Model-based RL: Planning, world models, Applications: Game playing (Atari, AlphaGo), robotics, recommendation systems, Challenges: Sample efficiency, credit assignment, safety in RL	
	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

List of Practicals:

1. Introduction to OpenAI Gym: Setting up and interacting with Gym environments.
2. Solving MDP using Dynamic Programming: Implement policy/value iteration on a grid world.
3. Monte Carlo Control: First-visit Monte Carlo control algorithm in an episodic task.
4. TD(0) and SARSA: Implement TD learning and SARSA for a FrozenLake environment.
5. Q-Learning: Implement Q-learning for a navigation or grid-based environment.
6. Function Approximation with Linear Models: Use linear value function approximators for continuous state spaces.
7. Deep Q-Networks (DQN): Build and train a DQN to solve CartPole-v1.
8. Double DQN and Dueling DQN: Extend DQN implementation with double Q-learning and dueling architecture.
9. Policy Gradient (REINFORCE): Implement REINFORCE on CartPole or MountainCar environments.
10. Actor-Critic and PPO: Use Actor-Critic or PPO algorithms to train agents in continuous environments like Pendulum-v0.
11. Mini-Project: Design and develop an RL agent to solve a real-world task (e.g., inventory management, dynamic pricing, game AI).

References:

1. Reinforcement Learning: An Introduction, Richard S. Sutton and Andrew G. Barto, MIT Press
2. Deep Reinforcement Learning Hands-On, Maxim Lapan, Packt Publishing

3. Algorithms for Reinforcement Learning, Csaba Szepesvári, Morgan & Claypool

M.Tech CSE	Semester: II	Batch: 2025-27
Course Code: CSE516	L-T-P: 3-0-1	Credit: 4
Course Title: Stochastic Models and Applications		

Course Objectives:

To develop a strong foundation in stochastic processes and probabilistic modeling, enabling students to analyze, simulate, and apply models like Markov chains, Poisson processes, and queueing theory in real-world systems such as networking, algorithms, and distributed computing.

Course Outcomes: Through this course students should be able to

CO1: Explain core concepts of probability theory and stochastic processes for modeling uncertainty in dynamic systems.

CO2: Construct and analyze discrete and continuous-time Markov chains for system modeling.

CO3: Apply Poisson processes, renewal theory, and queueing models to evaluate performance in real-world computing systems.

CO4: Utilize simulation tools and numerical methods to assess reliability and performance of stochastic models.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

Course Outcomes (COs)	PO 1	PO 2	PO 3	PO 4	PO 5	PSO 1	PSO 2	PSO 3
CO1: Explain core concepts of probability theory and stochastic processes for modeling uncertainty in dynamic systems.	**3**	**2**	**3**	**2**	**1**	**3**	**2**	**1**
CO2: Construct and analyze discrete and continuous-time Markov chains for system modeling.	**3**	**3**	**3**	**2**	**2**	**3**	**2**	**1**

CO3: Apply Poisson processes, renewal theory, and queueing models to evaluate performance in real-world computing systems.	**3**	**3**	**2**	**3**	**1**	**3**	**3**	**2**
CO4: Utilize simulation tools and numerical methods to assess reliability and performance of stochastic models.	**3**	**3**	**2**	**3**	**2**	**2**	**3**	**1**

3 = Strongly mapped, 2 = Moderately mapped, 1 = Slightly mapped

Detailed Syllabus

Units	Course Outlines	No. of Lectures
Unit-I	Probability and Random Processes: Review of probability theory: Random variables, probability distributions, Expectation, variance, moment generating functions, Conditional probability, independence, Bayes' theorem, Random processes: Definitions and classification, Stationarity, ergodicity, autocorrelation, and autocovariance, Applications in computer systems (packet delays, load balancing)	**12**
Unit-II	Markov Chains and Applications: Discrete-Time Markov Chains (DTMC): Transition probability matrix, n-step transition, Classification of states: recurrent, transient, absorbing, Limiting distributions, steady-state behavior, Continuous-Time Markov Chains (CTMC): Birth-death processes, Applications in web modeling (PageRank), memory management, job scheduling	**12**
Unit-III	Poisson Processes and Renewal Theory: Poisson process: Inter-arrival times, memoryless property, Non-homogeneous Poisson processes, Renewal processes: Renewal function, elementary renewal theorem, Applications: System reliability, failure-repair models, Renewal reward process, regenerative processes	**11**
Unit-IV	Queueing Theory and Stochastic Applications: Basic Queueing Models: M/M/1, M/M/c, M/G/1, Little's Law, Kendall's notation, Queueing networks and product-form solutions, Reliability theory: Reliability functions, hazard rate, MTBF, Applications in cloud computing, traffic modeling, and parallel systems, Simulation of queueing systems and Monte Carlo methods	**11**

	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

List of Practicals:

1. Probability Distribution Simulation: Simulate and visualize discrete and continuous probability distributions (Binomial, Poisson, Normal).
2. Markov Chain Analysis: Implement DTMC and analyze long-run behavior using transition matrices.
3. Birth-Death Process Simulation: Simulate CTMC for M/M/1 and M/M/c queues.
4. Poisson Process Simulation: Simulate event arrivals using exponential inter-arrival times.
5. Queueing Models with Little's Law: Validate Little's Law with simulation of M/M/1 queues.
6. Monte Carlo Simulation: Estimate probability distributions or integrals using Monte Carlo methods.
7. Renewal Process Modeling: Simulate component failure and repair processes.
8. System Reliability Analysis: Compute and visualize reliability metrics (hazard rate, MTBF) of series and parallel systems.
9. Stochastic Modeling in Network Traffic: Model bursty traffic using Poisson/Markov Modulated models.
10. Project: Stochastic Modeling of a Real-World System: E.g., Elevator system, traffic lights, cloud resource management, load balancing in distributed systems.

References:

1. Introduction to Probability Models, Sheldon M. Ross, Academic Press
2. Stochastic Processes, J. Medhi, New Age International

M.Tech CSE	Semester: II	Batch: 2025-27
Course Code: CSE517	L-T-P: 3-0-1	Credit: 4
Course Title: Bioinformatics		

Course Objectives:

This course introduces the fundamentals of bioinformatics and computational biology, covering biological databases, sequence analysis, genome and protein structure prediction, and functional genomics. It emphasizes the application of machine learning, data mining, and statistical tools to design, simulate, and develop computational solutions for biological systems.

Course Outcomes: Through this course students should be able to

CO1: Explain the fundamental concepts and interdisciplinary role of bioinformatics in biology, genetics, and medicine.

CO2: Retrieve, manage, and interpret biological data using genomic and proteomic databases.

CO3: Apply sequence alignment algorithms and computational tools to analyze DNA, RNA, and protein sequences.

CO4: Design and evaluate bioinformatics workflows incorporating statistical and machine learning techniques for biological data analysis.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

Course Outcomes (COs)	PO 1	PO 2	PO 3	PO 4	PO 5	PSO 1	PSO 2	PSO 3
CO1: Explain the fundamental concepts and interdisciplinary role of bioinformatics in biology, genetics, and medicine.	**3**	**2**	**3**	**1**	**1**	**3**	**2**	**2**
CO2: Retrieve, manage, and interpret biological data using genomic and proteomic databases.	**3**	**3**	**3**	**2**	**1**	**3**	**3**	**1**
CO3: Apply sequence alignment algorithms and computational tools to analyze DNA, RNA, and protein sequences.	**3**	**3**	**3**	**3**	**2**	**3**	**3**	**1**

CO4: Design and evaluate bioinformatics workflows incorporating statistical and machine learning techniques for biological data analysis.	**3**	**3**	**3**	**3**	**2**	**3**	**3**	**1**

3 = Strongly mapped, 2 = Moderately mapped, 1 = Slightly mapped

Detailed Syllabus

Units	Course Outlines	No. of Lectures
Unit-I	Introduction DNA Structure, Sequence and Processing of DNA Introduction to molecular computing, Mathematical and Statistical Theory of Biological Sequences Computing Sequence Database Searching,	**11**
Unit-II	Introduction to language theory, Sticker systems Representing and Finding Sequence Features Machine Learning Techniques in Computational Biology, Parallelism of DNA strands; Watson Crick Automata; Insertion-Deletion Systems; Splicing circular strings, Distributed H Systems, Splicing revisited, Protein Coding Regions , Gene Finding Phylogenetic Tree	**12**
Unit-III	Protein-Protein Interaction Networks, Computational Identification of Protein-Protein Interactions in Model Plant Proteomes, Recent trends and applications of DNA Computing in computer networks Clustering of Protein-Protein Interaction Network	**12**
Unit-IV	Biological Imaging Analysis I-live cell images:, II-Automated Interpretation of Sub-cellular Patterns in Microscope Images :Protein Location Localization motifs: & III-Supervised learning of patterns ; Virtual Cell	**11**
	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

List of Practicals:

1. Retrieval and Analysis of DNA Sequences: Use NCBI/Ensembl databases to retrieve DNA sequences, Perform GC content analysis, motif detection, and ORF identification.

2. Sequence Alignment: Perform pairwise and multiple sequence alignment using tools like BLAST, CLUSTALW.
3. Sequence Pattern Matching: Implement algorithms like Knuth-Morris-Pratt (KMP) or regular expressions to detect motifs.
4. DNA Sequence Visualization: Visualize DNA sequences using tools like SnapGene, BioPython, or IGV (Integrative Genomics Viewer).
5. Simulation of Sticker Systems and Watson-Crick Automata: Use Python to simulate basic models of DNA strand parallelism or sticker systems.
6. Gene Prediction Using Hidden Markov Models (HMMs): Predict gene regions using tools like GENSCAN or implement a simple HMM.
7. Constructing and Analyzing Phylogenetic Trees: Use MEGA or PHYLIP software to build and interpret phylogenetic trees.
8. Application of ML Algorithms to Biological Sequences: Train classifiers (SVM, Random Forest) to distinguish between coding and non-coding regions.
9. Visualization of Protein-Protein Interaction (PPI) Networks: Use Cytoscape to build and analyze a PPI network from STRING or BioGRID data.
10. Clustering of PPI Networks: Perform community detection or k-means clustering on PPI data.
11. Prediction of PPI Using Computational Tools: Use tools like PIPE or iPPI-DB to predict novel interactions.
12. Automated Analysis of Subcellular Localization Images: Use open-source image analysis tools like ImageJ or CellProfiler.
13. Supervised Learning for Protein Localization: Train and test ML models (e.g., SVM, Decision Trees) to classify protein localization based on extracted features from images.
14. Feature Extraction from Cell Images: Extract morphological or texture features (e.g., Haralick, Zernike) using Python libraries (OpenCV/skimage).
15. Virtual Cell Simulation: Use Virtual Cell software to simulate cellular biochemical pathways and visualize subcellular processes.

References:

1. Atwood, T., & Parry, D. J. (2003). Introduction to Bioinformatics (1st ed.). Pearson Smith Publication.

2. Baxevanis, A. D., & Francis Ouellette, B. F. (2004). Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins (3rd ed.). Wiley-Interscience.

3. DNA Computing by Paun, Gheorghe, Rozenberg, Grzegorz, Salomaa, Arto; Springer publication

4. de Vries, A., & Meys, J. (2021). R für Dummies. John Wiley & Sons.

5. Krane, D. E., & Raymer, M. L. (2004). Fundamental Concepts of Bioinformatics (1st ed.). Pearson Education.

6. Lesk, A. (2019). Introduction to bioinformatics. Oxford University Press.

7. M.S. Waterman, Introduction to Computational Biology Maps, Sequences and Genomes, Chapman and Hall – CRC Press, 1995.

8. Mount, D. W. (2004). Sequence and Genome Analysis (1st ed.). Cold Spring Harbor Laboratory

9. R. Durbin, S. Eddy, A. Krogh, G. Mitchison, Biological Sequence Analysis Cambridge University Press, 1998.

10. Schmuller, J. (2018). R Projects for Dummies. John Wiley & Sons.

11. Wickham, H., Çetinkaya-Rundel, M., & Grolemund, G. (2023). R for data science. O'Reilly Media, Inc..

12. Xiong, J. (2006). Essential Bioinformatics. Cambridge University Press

M.Tech CSE	Semester: II	Batch: 2025-27
Course Code: CSE518	L-T-P: 3-0-1	Credit: 4
Course Title: Multi-Criteria Decision Making		

Course Objectives:

This course aims to provide a comprehensive understanding of decision theory and MCDM techniques, including classical, modern, and fuzzy approaches, with a focus on real-world applications in AI, ML, optimization, and hands-on computational decision modeling.

Course Outcomes: Through this course students should be able to

CO1: Formulate complex decision-making problems involving multiple and conflicting criteria in structured formats.

CO2: Apply classical MCDM techniques such as AHP, TOPSIS, and ELECTRE to evaluate and rank alternatives.

CO3: Implement fuzzy and intuitionistic approaches to MCDM for handling uncertainty in decision-making environments.

CO4: Design and analyze decision-support systems using MCDM tools to derive rational and defensible conclusions.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

Course Outcomes (COs)	PO 1	PO 2	PO 3	PO 4	PO 5	PSO 1	PSO 2	PSO 3
CO1: Formulate complex decision-making problems involving multiple and conflicting criteria in structured formats.	**3**	**2**	**2**	**3**	**2**	**3**	**2**	**1**
CO2: Apply classical MCDM techniques such as AHP, TOPSIS, and ELECTRE to evaluate and rank alternatives.	**3**	**3**	**3**	**3**	**2**	**3**	**2**	**–**

CO3: Implement fuzzy and intuitionistic approaches to MCDM for handling uncertainty in decision-making environments.	**3**	**2**	**3**	**3**	**1**	**3**	**3**	**2**
CO4: Design and analyze decision-support systems using MCDM tools to derive rational and defensible conclusions.	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**1**

3 = Strongly mapped, 2 = Moderately mapped, 1 = Slightly mapped

Detailed Syllabus

Units	Course Outlines	No. of Lectures
Unit-I	Introduction to Decision Making and MCDM Fundamentals of decision making, Types of decision environments: certainty, risk, and uncertainty, Structure of multi-criteria decision problems, Classification of MCDM methods: multi-objective and multi-attribute, Value function approach vs. outranking approach, Applications in engineering, business, software evaluation, and environmental systems	**11**
Unit-II	Classical MCDM Methods Weighted Sum Model (WSM), Weighted Product Model (WPM), Analytic Hierarchy Process (AHP): pairwise comparison, consistency index, Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), ELimination Et Choix Traduisant la REalité (ELECTRE I, II), Preference Ranking Organization Method for Enrichment Evaluations (PROMETHEE I, II), Strengths and weaknesses of each method	**12**
Unit-III	Fuzzy and Advanced MCDM Techniques Fuzzy logic fundamentals: membership functions, fuzzy numbers, Fuzzy AHP, Fuzzy TOPSIS, Intuitionistic fuzzy sets and interval-valued fuzzy sets, Grey Relational Analysis (GRA), Hybrid MCDM models and ensemble approaches, Case studies in uncertain environments	**12**
Unit-IV	Implementation and Applications Decision support systems (DSS) architecture using MCDM, Use of MCDM in software project selection, cloud service selection, supply chain, etc, Tools and libraries: MATLAB, Python (Scikit-MCDM, PyMCDA), Excel Solver, Ethical	**10**

	considerations in decision making, Recent trends in MCDM: AI-integrated, ML-driven decision systems	
	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

List of Practicals:

1. Modeling Decision Problems: Identify alternatives, criteria, and construct a decision matrix.
2. Implementation of AHP: Compute pairwise comparison matrix, weights, and consistency ratio using Excel or Python.
3. TOPSIS Method Application: Rank alternatives based on closeness to ideal solution using a real dataset.
4. ELECTRE Method: Develop outranking relations and rank alternatives using ELECTRE I.
5. Fuzzy AHP using Triangular Fuzzy Numbers: Apply fuzzy pairwise comparison and defuzzification for weight calculation.
6. Fuzzy TOPSIS Implementation: Implement fuzzy distance and ranking logic using Python or MATLAB.
7. Grey Relational Analysis: Perform normalization, deviation sequences, and GRC computation.
8. Hybrid MCDM Implementation: Combine AHP with TOPSIS or GRA and analyze results.
9. MCDM-based Decision Support System: Design a simple GUI or script-based tool for service/technology selection.

References:

1. Multiple Criteria Decision Analysis: State of the Art Surveys, José Figueira, Salvatore Greco, Matthias Ehrgott, Springer
2. Multicriteria Decision Making: An Operations Research Approach, V. Kreinovich, H.T. Nguyen, Springer

M.Tech CSE	Semester: II	Batch: 2025-27
Course Code: CSE519	L-T-P: 3-0-1	Credit: 4
Course Title: Intrusion Detection System		

Course Objectives:

To equip students with comprehensive knowledge of intrusion detection and prevention systems, including their types, architectures, and detection techniques, while integrating machine learning approaches and practical experience with IDS tools and datasets for effective cybersecurity analysis.

Course Outcomes: Through this course students should be able to

CO1: Explain the fundamental principles and components of intrusion detection systems.

CO2: Analyze different types of network attacks and system vulnerabilities.

CO3: Implement signature-based and anomaly-based intrusion detection models.

CO4: Use machine learning and deep learning techniques for advanced IDS solutions and evaluate IDS performance and deploy open-source IDS tools like Snort or Suricata.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

Course Outcomes (COs)	PO 1	PO 2	PO 3	PO 4	PO 5	PSO 1	PSO 2	PSO 3
CO1: Explain the fundamental principles and components of intrusion detection systems.	**2**	**2**	**3**	**1**	**1**	**2**	**1**	**2**
CO2: Analyze different types of network attacks and system vulnerabilities.	**3**	**3**	**3**	**2**	**1**	**2**	**2**	**2**
CO3: Implement signature-based and anomaly-based intrusion detection models.	**3**	**3**	**3**	**2**	**2**	**3**	**2**	**2**
CO4: Use machine learning and deep learning techniques for advanced IDS solutions and evaluate IDS performance and	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**2**

deploy open-source IDS tools like Snort or Suricata.								

3 = Strongly mapped, 2 = Moderately mapped, 1 = Slightly mapped

Detailed Syllabus

Units	Course Outlines	No. of Lectures
Unit-I	Fundamentals of Intrusion Detection Systems: Introduction to IDS and Intrusion Prevention Systems (IPS), Categories of IDS: Host-based IDS (HIDS), Network-based IDS (NIDS), Hybrid IDS, IDS vs. Firewalls, Components of IDS: Sensors, Analyzers, User Interface, Threats and vulnerabilities: DoS, R2L, U2R, Probe, IDS architecture and deployment models	**8**
Unit-II	Detection Techniques and Models: Signature-based detection: patterns, rule engines (e.g., Snort rules), Anomaly-based detection: statistical approaches, threshold models, Specification-based detection, Alert generation and correlation, Comparative evaluation of detection models, Datasets: KDDCup99, NSL-KDD, CICIDS2017	**12**
Unit-III	Machine Learning and AI in IDS: Feature extraction and selection in network data, Supervised learning models: SVM, Random Forest, Decision Trees, Unsupervised learning: Clustering, Outlier detection (k-Means, Isolation Forest), Ensemble learning for IDS, Deep learning for IDS: DNNs, CNNs, RNNs, Evaluation metrics: Precision, Recall, F1-score, ROC curve	**13**
Unit-IV	IDS Tools, Implementation and Challenges: Overview of popular IDS tools: Snort, Suricata, Zeek (Bro), IDS rule writing and testing, Integrating IDS with SIEM (Security Information and Event Management) tools, Limitations of IDS: False Positives/Negatives, Evasion Techniques, Emerging topics: Adversarial ML in IDS, IDS in IoT and cloud environments, Case studies and real-world deployment scenarios	**12**
	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

List of Practicals:

1. Installation and Configuration of Snort: Setup and run Snort in NIDS mode.
2. Packet Capture and Traffic Analysis: Analyze network traffic using Wireshark and identify potential threats.
3. Writing and Testing Snort Rules: Create custom rules to detect ICMP, TCP scans, and DoS attacks.
4. Feature Extraction from IDS Datasets: Preprocess NSL-KDD or CICIDS datasets for use in ML models.
5. Anomaly Detection using Statistical Thresholding: Implement basic threshold-based anomaly detection in Python.
6. Intrusion Detection using Machine Learning: Train and test ML classifiers (SVM, Random Forest) on NSL-KDD dataset.
7. Unsupervised Anomaly Detection: Apply k-Means clustering or Isolation Forest to detect novel intrusions.
8. Deep Learning for IDS: Use TensorFlow/Keras to implement a DNN for attack classification.
9. Evaluate IDS Performance: Calculate detection rate, false positive rate, and plot ROC/AUC.

References:

1. Intrusion Detection and Prevention Systems, Ali A. Ghorbani, Wei Lu, Mahbod Tavallaee, Springer
2. Machine Learning and Security: Protecting Systems with Data and Algorithms, Clarence Chio and David Freeman, O'Reilly Media

M.Tech CSE	Semester: II	Batch: 2025-27
Course Code: CSE520	L-T-P: 3-0-1	Credit: 4
Course Title: Next Generation Database		

Course Objectives:

To introduce emerging database paradigms beyond traditional RDBMS, focusing on NoSQL, distributed, cloud-based, and specialized databases, while providing practical exposure to modern data technologies and performance analysis.

Course Outcomes: Through this course students should be able to

CO1: Compare traditional and next-generation database architectures and models.

CO2: Apply NoSQL and NewSQL databases for scalable and high-performance applications.

CO3: Implement graph and object-based databases for complex relationships and domain-specific applications.

CO4: Analyze the role of consistency, partitioning, and replication in distributed databases; Develop, deploy, and evaluate cloud-based and non-relational database systems using real-world datasets.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

Course Outcomes (COs)	PO 1	PO 2	PO 3	PO 4	PO 5	PSO 1	PSO 2	PSO 3
CO1: Compare traditional and next-generation database architectures and models.	**2**	**2**	**3**	**2**	**1**	**3**	**2**	**1**
CO2: Apply NoSQL and NewSQL databases for scalable and high-performance applications.	**3**	**3**	**3**	**3**	**2**	**3**	**3**	**2**
CO3: Implement graph and object-based databases for complex relationships and domain-specific applications.	**2**	**3**	**3**	**2**	**2**	**2**	**3**	**1**

CO4: Analyze the role of consistency, partitioning, and replication in distributed databases; Develop, deploy, and evaluate cloud-based and non-relational database systems using real-world datasets.	**3**	**3**	**3**	**3**	**2**	**3**	**3**	**1**

3 = Strongly mapped, 2 = Moderately mapped, 1 = Slightly mapped

Detailed Syllabus

Units	Course Outlines	No. of Lectures
Unit-I	Introduction and Limitations of RDBMS: Review of relational databases and limitations in modern applications, Requirements of next-generation databases, Introduction to ACID vs. BASE properties, Overview of next-generation paradigms: NoSQL, NewSQL, CAP theorem, Big Data and its impact on database systems	**9**
Unit-II	NoSQL and NewSQL Databases: Key-Value Stores: Redis, Amazon DynamoDB, Document Stores: MongoDB, CouchDB, Column Family Stores: Apache Cassandra, HBase, Graph Databases: Neo4j, OrientDB, NewSQL systems: Google Spanner, VoltDB, CockroachDB, Query languages for NoSQL systems, Trade-offs: Consistency, Availability, Partition Tolerance	**12**
Unit-III	Advanced Data Models: Object-oriented databases: DB4O, ObjectDB, Time-series databases: InfluxDB, TimescaleDB, Graph data models and queries: Cypher, Gremlin, Multi-model databases: ArangoDB, OrientDB, Schema-less designs and versioning, Data modeling in NoSQL vs. RDBMS	**11**
Unit-IV	Distributed and Cloud Databases: Distributed database architecture and design, Sharding and replication strategies, CAP theorem in practice, Database-as-a-Service (DBaaS): Firebase, Google Bigtable, Amazon Aurora, Data security, backup, and disaster recovery in distributed systems, Performance tuning, scalability, and benchmarking, Case studies: Netflix, Facebook, Google Cloud Spanner	**13**
	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

List of Practicals:

1. Installation and Basic Operations in MongoDB: CRUD operations, indexing, and aggregation framework.
2. Working with Redis or DynamoDB: Key-value data manipulation and real-time analytics.
3. Column Store Operations in Cassandra: Creating keyspaces, column families, and executing CQL queries.
4. Graph Database Modeling in Neo4j: Use Cypher queries to analyze social network or recommendation data.
5. Time-Series Data Storage and Querying: Use InfluxDB or TimescaleDB for IoT sensor data analysis.
6. Object-Oriented Database Implementation: Use DB4O or ObjectDB to persist Java or .NET objects.
7. Designing a Sharded and Replicated MongoDB Cluster: Demonstrate scaling and failover recovery.
8. Cloud Database Integration: Connect and interact with Firebase/Google Bigtable/Aurora from a web or mobile application.
9. Comparative Performance Analysis: Compare MongoDB vs. MySQL for structured and semi-structured data.

References:

1. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Pramod J. Sadalage and Martin Fowler, Addison-Wesley
2. Designing Data-Intensive Applications, Martin Kleppmann, O'Reilly Media