VIDYASAGAR UNIVERSITY

Proposed PG Syllabus (NEP)

M.Sc. in Computer Science

Under the Department of Computer Science

From Session 2025-2026

MIDNAPORE-721102

PG Syllabus 2025

PG 1^{st} Year				
Sem	**Course Code**	**Course Title**	**Credit**	**Marks**
I	DSC 1 (T+P)	Software Engineering	3 + 1	35 + 15
	DSC 2 (T+P)	Compiler Design	3 + 1	35 + 15
	DSC 3 (T+P)	Research Methodology and Ethics	3 + 1	35 + 15
	DSE 1 (T+P)	1. Cloud Computing 2. Real Time System 3. Mobile Computing	3 + 1 3 + 1 3 + 1	35 + 15 35 + 15 35 + 15
	DSE 2 (T+P)	1. Data Science 2. Pattern Recognition & Image Processing 3. Natural Language Processing	3 + 1	35 + 15
	Indian Knowledge System (IKS)		2	25
	Vidyasagar: Life and Philosophy		Compulsory non-Credit course	
	Total		**22**	**275**
II	DSC 4 (T+P)	Artificial Intelligence	3 + 1	35 + 15
	DSE 3 (T+P)	1. Internet of Things 2. Signal Processing 3. Cryptography and Steganography	3 + 1	35 + 15
	DSE 4 (T+P)	1. Green Computing 2. Soft Computing 3. Quantum Computing	3 + 1	35 + 15
	DSE 5 (T+P)	1. Cyber Security 2. Computer Vision 3. Big Data Analytics	3 + 1	35 + 15
	DSC 5 (T+P)	Machine Learning	3 + 1	35 + 15
	Field Visit / Industry Visit /Case Study / Hands-on Practical/ Skill Enhanced Course		2	25
	Total		**22**	**275**
Total: 1^{st} Year of PG				

III	DSC 6 (T+P)	Advanced Data Structure and Algorithm	3 + 1	35 + 15
	DSC 7 (T+P)	Advanced Database Management System	3 + 1	35 + 15
	DSC 8 (T+P)	Deep Learning	3 + 1	35 + 15
	DSC 9 (T+P)	Advanced Network	3 + 1	35 + 15
	MOOCs		4	50
	Social Service / Community Engagement		2	25
	Total		**22**	**275**
IV	DSC 10 (T+P)	Gen AI	3 + 1	35 + 15
	DSC 11 (T+P)	High Performance Computing	3 + 1	35 + 15
	Research Project/Dissertation		8	100
	Internship / Capstone Project / Applied Field or Industry Project/ Innovation & Incubation/ Entrepreneurship/ Start-up Proposal or Practice		4	50
	Intellectual Property Right (IPR) / Skill Enhanced Course		2	25
	Total		**22**	**275**

Program Outcome

M.Sc. in Computer Science students will be able to

PO1. Computational Knowledge: Apply knowledge of computing fundamentals, computing specialisation, mathematics, and domain knowledge appropriate for the computing specialisation to the abstraction and conceptualisation of computing models from defined problems and requirements.

PO2. Problem Analysis: Identify, formulate, research literature, and solve complex computing problems reaching substantiated conclusions using fundamental principles of mathematics, computing sciences, and relevant domain disciplines.

PO3. Design /Development of Solutions: Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.

PO4. Conduct investigations of complex Computing problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Modern Tool Usage: Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations.

PO6. Professional Ethics: Understand and commit to professional ethics and cyber regulations, responsibilities, and norms of professional computing practices.

PO7. Life-long Learning: Recognise the need, and have the ability, to engage in independent learning for continual development as a computing professional.

PO8. Project management and finance: Demonstrate knowledge and understanding of the computing and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO9. Communication Efficacy: Communicate effectively with the computing community, and with society at large, about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.

PO10. Societal and Environmental Concern: Understand and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practices.

PO11. Individual and Team Work: Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary environments.

PO12. Innovation and Entrepreneurship: Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.

Semester - I

DSC 1T: Software Engineering **3 credits**

Course Outcome (CO):

Students able to:

CO1: Learn software engineering process models and metrics to plan and manage software development effectively**.**

CO2: Learn coding best practices, inspections, and walkthroughs to produce maintainable and reliable code.

CO3: Learn and execute effective test strategies using white-box, black-box, and specialized testing techniques.

CO4: Manage software maintenance processes, including cost estimation and reverse engineering, for long-term sustainability.

Detailed Syllabus:

Unit 1: Introduction :

Software, Software Myths, Software types, Software engineering : A Layered Technology, Different Software Process Models, Fourth Generation Techniques, Software project management. **[6L]**

Unit 2: Software Design :

Problem Partitioning, Top-Down And Bottom-Up design, Decision tree, decision table, Cohesion, Coupling, Design approaches : Functional and Object- Oriented approach. Function- Oriented Software Design : SA/SD methodology, Structure analysis, DFD, Structure chart, Design review Object-Oriented Software Design : Concept of OO Software – Design and Analysis, Overview of various UML diagrams and UML analysis modeling, analysis case studies, analysis tools, analysis patterns. **[8L]**

Unit 3: Coding :

Coding standards, code walk- throughs, code inspection, clean room testing and documentation. **[4L]**

Unit 4: Software Quality Assurance :

Quality Concepts, The Quality Movement, Software Quality Assurance, Software Reviews, Formal Technical Reviews, Formal Approaches to SQA, Statistical Software Quality Assurance, Software Reliability, Mistake Proofing for Software, Introduction to ISO standard. **[7L]**

Unit 5: Software Testing Technique :

Software testing fundamentals, Test case design, White-box Testing, Basis path testing, Control structure testing, Black-box testing, Testing for specialized environments, architectures and applications. **[6L]**

Unit 6: Software Maintenance:

Characteristics, reverse engineering, maintenance process models, estimation of cost. **[4L]**

DSC 1P: Software Engineering Lab **1 credits**

[20L]

Program Details:

1. Analysis report of project-
   a. Abstract of Project
   b. Model Description (Best Suitable for Project)
   c. Technology Used
   d. References used for project (Similarities, Differences, Webpages)
2. To perform the system analysis: Requirement analysis, SRS. (According Format)
3. To perform the function oriented diagram: DFD or Structured chart.
4. To perform UML diagrams using UML tool
   a. Introduction of UML
   b. Introduction UML Tool.
5. To perform the user's view analysis: Use case diagram. (Based on given project)
6. To draw the structural view diagram: Class diagram, object diagram. (Based on given project)
7. To draw the behavioral view diagram: Sequence diagram, Collaboration diagram. (Based on given Project)
8. To draw the behavioral view diagram: State-chart diagram, Activity diagram. (Based on given Project)
9. To draw the implementation view diagram: Computer diagram. (Based on given project)

Sample Projects:

1. Passport automation System
2. Book Bank
3. Online Exam Registration
4. Stock Maintenance System
5. Online course reservation system
6. E-ticketing
7. Software Personnel Management System
8. Credit Card Processing
9. E-book management System.
10. Recruitment system

References Books:

1. Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill.
2. Sommerville, I. (2016). *Software Engineering* (10th ed.). Addison-Wesley.
3. Bruegge, B., & Dutoit, A. H. (2010). *Object-Oriented Software Engineering: Using UML,*

Patterns, and Java (3rd ed.). Pearson Prentice Hall.
4. Jalote, P. (2006). *An Integrated Approach to Software Engineering* (3rd ed.). Springer.
5. Mall, R. (2018). *Fundamentals of Software Engineering* (5th ed.). PHI Learning.
6. Aggarwal, K. K., & Singh, Y. (2016). *Software Engineering* (3rd rev. ed.). New Age International Publishers.
7. Braude, E. J. (2001). *Software Engineering: An Object-Oriented Perspective*. Wiley.
8. Bell, D. (2005). *Software Engineering for Students* (4th ed.). Addison-Wesley.
9. Schmuller, J. (2004). *SAMS Teach Yourself UML in 24 Hours* (3rd ed.). Sams Publishing.

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	CO1	1	2	2	2	2	3	2	2	3	1	2	2
	CO2	-	2	3	2	2	2	2	2	2	1	2	2
	CO3	-	2	2	1	2	2	2	2	1	1	2	3
	CO4	-	2	1	1	2	2	2	2	1	1	1	2

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSC 2T: Compiler Design **3 credits**

Course Outcome (CO):

Students able to:

CO1: Learn the structure and phases of a compiler (lexical analysis, syntax analysis, semantic analysis, code generation, and optimization). Also learn regular expressions for lexical analysis and implement a lexical analyzer.

CO2: Learn parsing techniques (Top-down LL(1), Bottom-up LR(0), and Shift-Reduce parsing) to construct parse trees. Apply LEX and YACC tools to perform lexical analysis and syntax analysis for various programming and natural language inputs.

CO3: Learn to generate intermediate code (Three-address code, quadruples, triples) and optimize code using DAGs and peephole optimization.

CO4: Analyze runtime environments, including storage allocation, parameter passing, and symbol table management and design a simple compiler module by integrating lexical, syntax, and semantic analysis phases.

Details Syllabus:

Unit 1: Introduction **[8L]**

Introduction to Compiler, Different phases and passes of compiler. Lexical Analysis: Role of Lexical Analyzer, Input Buffering, Specification of Tokens, Finite state machines and regular expressions and their applications to lexical analysis, Implementation of lexical analyzers

Unit 2: Syntex Analysis **[8L]**

Syntax Analysis: Role of the parser, Formal grammars and their application to syntax analysis,Context free grammars, Derivation and parse trees, Top Down parsing, LL(1) grammars, Predictive Parsing, Bottom-up-parsing, Shift Reduce Parsing, LR(0) grammars, LR parsing algorithms.

Unit 3: Semantic Analysis **[4L]**

Syntax Directed Translation: Syntax directed definitions, Construction of syntax trees, Bottom-up evaluation of S-attributed definitions, L-attributed definitions.

Unit 4: Runtime Environments **[4L]**

Source Language issues, Storage Organization, Storage Allocation strategies, Access to non-local names, Parameter passing mechanism.

Unit 5: Intermediate Code Generation **[5L]**

Intermediate languages, Graphical representation, Three address code, Implementation of three address statements (Quadruples, Triples, Indirect triples).

Unit 6: Code Optimization and Code Generation **[6L]**

Introduction, Basic blocks and flow graphs, Transformation of basic blocks, DAG representation of basic blocks, Principle sources of optimization, Loops in flow graph, Peephole optimization. Issues in the design of code generator, Register allocation and assignment.

DSC 2P: Compiler Design Lab: **1 credit**
[20L]

List of Program

1. Write a Program in LEX/YACC to check whether a given string is a valid ID (Identifier), Keyword, RELOP (Relational Operator) or others.
2. Write a program in LEX/YACC to check whether a given expression (relational or assignment or bitwise operator) is valid or not and it gives the type of expression as output.
3. Write a program in LEX /YACC which takes standard input as output of system date and time and give either of the following messages “ Good Morning”, “Good Afternoon”, “Good Evening” .
4. Construct a syntax directed translation scheme that translates integers into roman numerals. Implement translator from integers to roman numerals based on above syntax directed translation using LEX/YACC.
5. Write a program using FLEX/YACC, which recognize regular expression.
6. Write a C code analyzer in LEX/YACC: comments, code, white space, count braces, keywords etc. Try to identify function definition and declaration, which are names followed by ‘(‘outside of any braces.
7. Write programs in LEX/YACC, which replaces all the occurrences of “rama” with “RAMA” and “sita” with “ SITA”.

8. Write a program in LEX/YACC to check whether a sentence of English language is grammatically correct or not.
9. Write a program in LEX/YACC which takes a English sentence as input and gives the output as the parts of speech.
10. Write a program in LEX/YACC which takes a C program as inputs and delete the comment, white space and Count the no of lines.
11. Write a program in LEX/YACC which counts the no of lines, total no of characters, total no of vowels and total no of punctuation marks in a paragraph.
12. Write a program in LEX/YACC to check the parts of Speech of a Sentence.
13. Write a program in LEX/YACC to count all occurrences of “rama” and “sita” in a given file and eliminate them.
14. Write programs in LEX/YACC that eliminate multiple spaces and tabs and replace with a single space and remove empty lines.
15. Write a lex program to count the number of comment lines in a given C program. Also eliminate them and copy that program into separate file
16. Program to recognize whether a given sentence is simple or compound.
17. Program to recognize a valid arithmetic expression and to recognize the identifiers and operators present. Print them separately.

Reference Books

1. Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2007). *Compilers: Principles, Techniques, and Tools* (2nd ed.). Pearson Education.
   (Commonly referred to as the "Dragon Book")
2. Beck, L. L. (1997). *System Software: An Introduction to Systems Programming* (3rd ed.). Addison-Wesley.
3. Donovan, J. J. (1997). *Systems Programming*. McGraw-Hill.
4. Louden, K. C. (1997). *Compiler Construction: Principles and Practice*. Thomson Learning.
5. Fischer, C. N., & LeBlanc, R. J. (1991). *Crafting a Compiler with C*. Benjamin/Cummings.
6. Bennett, J. P. (1996). *Introduction to Compiler Techniques* (2nd ed.). Tata McGraw-Hill.
7. Holub, A. I. (1993). *Compiler Design in C*. Prentice Hall.
8. Alblas, H., & Nymeyer, A. (2001). *Practice and Principles of Compiler Building with C*. PHI Learning.
9. Dhamdhere, D. M. (1996). *Systems Programming and Operating Systems* (2nd ed.). Tata McGraw-Hill.

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	CO1	1	1	2	1	2	-	1	-	-	-	1	-
	CO2	2	3	2	1	2	-	1	-	-	-	1	-
	CO3	2	2	1	3	2	-	1	-	-	-	1	-
	CO4	2	2	1	2	2	-	1	-	-	-	1	-

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSC 3T: Research Methodology and Ethics **3 credits**

Course Outcome:

Students able to:

CO1: Learn to formulate research problems and design appropriate methodologies.
CO2: Learn to select and apply suitable data collection and sampling techniques.
CO3: Analyze data using descriptive and inferential statistics, create research instruments, and document them in LaTeX.
CO4: Interpret results ethically and present findings in a structured report in LaTex..

Details Syllabus:

Unit 1: Introduction to Research **[6 L]**

Meaning, objectives, and motivation for research, Types of research (Basic vs. Applied, Qualitative vs. Quantitative), Research approaches and scientific method, Criteria of good research

Unit 2: Research Design & Sampling **[8 L]**

Research problem formulation and hypothesis development, Research design: Need, features, and types (Experimental, Descriptive), Sampling techniques: Probability vs. Non-probability, sample size determination

Unit 3: Data Collection & Measurement **[8 L]**

Primary vs. secondary data, Tools: Questionnaires, interviews, observations, Measurement scales (Nominal, Ordinal, Interval, Ratio), Scaling techniques (Likert, Semantic Differential)

Unit 4: Data Analysis & Hypothesis Testing **[8 L]**

Descriptive statistics (Mean, SD, Skewness), Inferential statistics: t-test, ANOVA, Chi-square, Correlation and regression analysis, Non-parametric tests (Mann-Whitney, Kruskal-Wallis)

Unit 5: Research Ethics & Report Writing **[5 L]**

Ethical issues in research (Plagiarism, Confidentiality), Structure of a research report, Oral presentation techniques, Use of software (SPSS/R) for data analysis

DSC 3P: Research Methodology and Ethics Lab **1 credits**
[20L]

List of Programs:

1. LaTeX Installation and First Document
2. Formatting a Research Problem Statement
3. Creating a Research Proposal Template
4. Designing a Sampling Plan with Lists
5. Building a Questionnaire with Tables
6. Creating a Data Codebook
7. Data Import and Cleaning with R/Python
8. Generating Descriptive Statistics
9. Importing Data into LaTeX with csvsimple
10. Formatting Statistical Tables with booktabs
11. Performing and Reporting a T-Test
12. Creating a Correlation Matrix Table
13. Simple Linear Regression Analysis
14. Non-Parametric Mann-Whitney U Test
15. Managing References with BibTeX
16. Creating a Bar Chart with pgfplots
17. Creating a Scatter Plot with pgfplots
18. Plagiarism Check and Ethical Writing
19. Compiling a Full Research Report
20. Designing a Research Poster with beamerposter

Reference Books

1. Kothari, C. R. (2004). *Research Methodology: Methods and Techniques* (2nd ed.). New Age International Publishers.
2. Creswell, J. W., & Creswell, J. D. (2023). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches* (6th ed.). SAGE Publications.
3. Lamport, L. (1994). *LaTeX: A Document Preparation System* (2nd ed.). Addison-Wesley.
4. Mittelbach, F., & Goossens, M. (2004). *The LaTeX Companion* (2nd ed.). Addison-Wesley.

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1	CO1	3	2	2	2	3	3	3	-	1	1	3	2

(T+P)	**CO2**	**3**	**2**	**3**	**2**	**2**	**2**	**2**	**-**	**1**	**1**	**2**	**2**
	CO3	**3**	**3**	**3**	**2**	**3**	**2**	**3**	**-**	**1**	**1**	**2**	**2**
	CO4	**3**	**3**	**3**	**1**	**3**	**2**	**3**	**-**	**1**	**1**	**2**	**2**

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSE 1.1T: Cloud Computing **3 credits**

Course Outcome:

Students able to:

CO1: Learn fundamental concepts of cloud infrastructure to evaluate trade-offs in power, efficiency, and cost, and manage data centers for resilient, elastic, and cost-efficient cloud applications.

CO2: Learn to describe system, network, and storage virtualization techniques and explain their role in enabling cloud computing.

CO3: Learn to illustrate the principles of cloud storage and demonstrate their implementation in systems such as Amazon S3 and HDFS.

CO4: Analyze various cloud programming models and apply them to develop and deploy solutions on the cloud.

Details Syllabus:

Unit 1: Introduction: **[3L]**

Essentials, Benefits and need for Cloud Computing - Business and IT Perspective - Cloud and Virtualization - Cloud Services Requirements - Cloud and Dynamic Infrastructure - Cloud Computing Characteristics Cloud Adoption.

Unit 2: Cloud Models: **[4L]**

Cloud Characteristics - Measured Service - Cloud Models - Security in a Public Cloud Public versus Private Clouds - Cloud Infrastructure Self Service

Unit 3: Cloud as a Service: **[3L]**

Gamut of Cloud Solutions - Principal Technologies - Cloud Strategy Cloud Design and Implementation using SOA - Conceptual Cloud Model - Cloud Service Defined

Unit 4: Cloud Solutions: **[3L]**

Cloud Ecosystem - Cloud Business Process Management - Cloud Service Management - Cloud Stack - Computing on Demand (CoD) – Cloud sourcing.

Unit 5: Cloud Offerings: **[3L]**

Information Storage, Retrieval, Archive and Protection - Cloud Analytics Testing under Cloud - Information Security - Virtual Desktop Infrastructure - Storage Cloud.

Unit 6: Cloud Management: **[3L]**

Resiliency – Provisioning - Asset Management - Cloud Governance - High Availability and Disaster Recovery - Charging Models, Usage Reporting, Billing and Metering.

Unit 7: Cloud Virtualization Technology: **[5L]**

Virtualization Defined - Virtualization Benefits - Server Virtualization - Virtualization for x86 Architecture - Hypervisor Management Software - Logical Partitioning (LPAR) - VIO Server - Virtual Infrastructure Requirements.

Unit 8: Cloud Storage Virtualization: **[5L]**

Storage virtualization - Storage Area Networks - Network-Attached storage - Cloud Server Virtualization - Virtualized Data Center.

Unit 9: Cloud and SOA: **[3L]**

SOA Journey to Infrastructure - SOA and Cloud - SOA Defined - SOA and IaaS - SOA-based Cloud Infrastructure Steps - SOA Business and IT Services.

Unit 10: Cloud Infrastructure Benchmarking: **[3L]**

OLTP Benchmark - Business Intelligence Benchmark - e-Business Benchmark - ISV Benchmarks - Cloud Performance Data Collection and Performance Monitoring Commands - Benchmark Tools.

DSE 1.1P: Cloud Computing Lab **1 credits**

[20L]

List of Program:

1) Install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.

2) Install a C compiler in the virtual machine created using virtual box and execute Simple Programs

3) Install Google App Engine. Create “hello world app” and other simple web applications using python/java.

4) Use GAE launcher to launch the web applications.

5) Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.

6) Find a procedure to transfer the files from one virtual machine to another virtual machine.

7) Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)

8) Install Hadoop single node cluster and run simple applications like wordcount.

Reference Books

1. Buyya, R., Broberg, J., & Goscinski, A. M. (Eds.). (2011). *Cloud Computing: Principles and Paradigms*. John Wiley & Sons.
2. Hurwitz, J., Bloor, R., Kaufman, M., & Halper, F. (2010). *Cloud Computing For Dummies*. Wiley Publishing.

3. Rhoton, J. (2009). *Cloud Computing Explained: Implementation Handbook for Enterprises*. Recursive Press.
4. Rosenberg, J., & Matheos, A. (2010). *The Cloud at Your Service: The When, How, and Why of Enterprise Cloud Computing*. Manning Publications.
5. Saurabh, K. (2011). *Cloud Computing: Insight Into New-Era Infrastructure*. Wiley India.
6. Sosinsky, B. (2011). *Cloud Computing Bible*. John Wiley & Sons.

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	CO1	2	2	1	1	2	2	2	-	-	-	2	2
	CO2	3	2	2	2	2	2	2	1	-	-	2	2
	CO3	3	3	2	2	2	3	2	2	-	2	2	2
	CO4	3	3	2	2	2	3	2	2	-	2	2	2

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSE 1.2T: Real Time System **3 credits**

Course Outcome:

Students able to:

CO1: Learn to design and manage real-time systems by applying concepts of task scheduling, process management, and synchronization.
CO2: Students will be able to analyze and optimize real-time systems using various scheduling algorithms and fault-tolerant techniques.
CO3: Students will develop expertise in creating real-time applications and managing inter-process communication.
CO4: Students will acquire experience with real-time system modeling and case studies like RTLinux and Windows.

Details Syllabus:

Unit 1: Introduction to Real Time System **[3L]**

Historical background, Elements of a Computer Control System, RTS- Definition, Issues in Real-Time Computation, Structure of a Real-time System, Characterizing RTS and Tasks, Time Constraints, Performance measure for RTS.

Unit 2: Real Time Task Assignment and Scheduling **[8L]**

Classical uniprocessor scheduling of IRIS tasks, Common approaches to RT scheduling: clock driven, weighted Round Robin, Priority driven, Earliest Deadline First, Least Slack Time First and Rate Monitoring algorithm. Offline vs Online Scheduling, Scheduling Aperiodic and Sporadic jobs in Priority Driven and Clock driven Systems.

Unit 3: Handling Resource Sharing in RTS **[8L]**

Effect of Resource Contention and Resource Access Control (RAC), Non-preemptive critical sections, Basic priority Inheritance and Priority Ceiling protocols (PCP), Stack based PCP, Use of PCP in Dynamic priority Systems. Preemption Ceiling Protocol, Access Control in Multiple-Module resources. Controlling concurrent access to data objects.

Unit 4: Scheduling RT tasks in Multiprocessor and Distributed Systems **[4L]**

Multiprocessor and Dynamic task allocation, Fault tolerant Scheduling, Clock Synchronizations.

Unit 5: Real-Time Databases **[4L]**

Real-time vs General purpose Databases, Main memory Databases, Transaction priorities, Transaction aborts, Concurrency control issues, Disk Scheduling algorithms, Two-phase approach to improve predictability, Maintaining Serialization Consistency, Databases for Hard RTS.

Unit 6: Real-Time Communication **[4L]**

Basic concepts of RT communication, Soft and Hard RT communication systems, Model of RT communication, RT communications in LAN, Boundary Access Protocols for LANs, RT communications over Packet Switched networks, QoS, Rate Control, Internet and Resource Reservation Protocols.

Unit 7: Real-Time OS **[4L]**

Time Services, Features of RTOS, Time services, UNIX and Windows as RTOS, POSIX, Survey of contemporary RTOS viz. PSOS, VRTX, RT Linux, Lynx, Windows CE etc., Benchmarking RTS, Rhealstone Metric, Interrupt processing overhead, Tri-dimensional measure, Determining kernel preemptibility.

DSE 1.2P: Real Time System Lab **1 credits**
[20L]

Program Details:

Lab 1: Introduction to RTOS & Task Creation

Setup FreeRTOS/RTLinux. Write a periodic and an aperiodic task with deadlines.

Lab 2: Real-Time Scheduling Basics

Implement **Round Robin** and **Priority Scheduling**. Compare results with theoretical expectations.

Lab 3: Advanced Scheduling Simulation

Simulate **EDF (Earliest Deadline First)** and **Rate Monotonic (RM)** scheduling using SimSo.

Lab 4: Resource Sharing & Priority Inversion

Demonstration of **priority inversion**. Implement **priority inheritance** protocol.

Lab 5: Real-Time Communication (Mini Simulation)

Use NS-3 or socket programming to simulate soft vs. hard real-time message delivery.

Lab 6: RTOS Benchmarking & Case Study

Measure task latency and context switch time on RTLinux/FreeRTOS. Short case study of a real RTOS (RTLinux or FreeRTOS features).

References Books

1. Liu, J. W. S. (2000). *Real-Time Systems*. Prentice Hall.
2. Mall, R. (2007). *Real-Time Systems: Theory and Practice*. Pearson Education.
3. Krishna, C. M., & Shin, K. G. (2010). *Real-Time Systems*. McGraw-Hill.
4. Laplante, P. A., & Ovaska, S. J. (2012). *Real-Time Systems Design and Analysis: Tools for the Practitioner*. Wiley.
5. Laplante, P. A. (2004). *Real-Time Systems Design and Analysis* (3rd ed.). Wiley-IEEE Press.

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	CO1	3	2	3	-	2	1	2	-	-	-	1	-
	CO2	2	3	2	3	2	1	2	-	-	1	-	-
	CO3	3	2	3	-	3	1	2	-	-	-	2	2
	CO4	2	2	2	2	3	1	2	-	-	-	1	2

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSE 1.3T: Mobile Computing **3 credits**

Course Outcome:

Students able to:

CO1: Learn the principles and theories of mobile computing technologies.

CO2: Describe infrastructures and technologies of mobile computing technologies.

CO3: Learn the applications in different domains that mobile computing offers to the public, employees, and businesses.

CO4: Describe the possible future of mobile computing technologies and applications.

Details Syllabus:

Unit 1: Introduction **[4L]**

Introduction to wireless networks and mobile computing – Characteristics, Issues and challenges.

Unit 2: Wireless Transmission **[10L]**

Fundamentals of wireless transmission - Medium Access Control Protocols, Different types of multiple access techniques and their characteristics.

Unit 3: Cellular and Mobile Communication **[12L]**

Cellular concept, Overview of different Generations. Mobile IP, Mobile transport layer - Mechanisms for improving TCP performances on wireless links, Overview of Security in mobile environments, GSM architecture, GPRS architecture, UMTS Network architecture, CDMA, ISDN.

Unit 5: Wireless **[5L]**

Overview of Wireless LAN IEEE 802.11 series, Overview of Bluetooth, Overview of Wireless Sensor Networks.

Unit 6: Wireless application Environments **[4L]**

WAP, WML, Push Architecture, Push/Pull Services Mobile Adhoc Networks – Characteristics, Routing protocols.

DSE 1.3P: Mobile Computing Lab **1 credits**
[20L]

List of Lab Programs

1. Lab Setup and Emulator Configuration: Install and configure Android Studio, set up an Android Virtual Device (AVD), and run a "Hello World" application.
2. Network Simulation Basics: Install a network simulator (NS3 or OMNeT++) and create a simple wired network scenario to understand the simulation environment.
3. WLAN Setup and Analysis: Configure a basic Wireless LAN (WLAN) using access points and clients. Capture and analyze beacon frames, probe requests, and associations using Wireshark.
4. Multiple Access Techniques - CSMA/CA: Simulate a wireless network with multiple nodes to demonstrate and analyze the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol.
5. Multiple Access Techniques - TDMA/FDMA: Create a simulation to model and compare Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA) techniques in a shared medium.
6. Cellular Network Handover Simulation: Simulate a mobile node moving between two base stations to demonstrate and analyze a hard handover (break-before-make) mechanism.
7. Generational Technology Comparison (4G vs. 5G): Use a simulation tool to model and compare key performance metrics (e.g., latency, throughput) of 4G LTE and 5G NR networks under similar conditions.
8. Mobile IP Configuration: Configure Mobile IP on a network simulator to demonstrate how a mobile node can maintain its IP address while moving between different networks.
9. TCP for Wireless Networks: Simulate a TCP file transfer over a wireless link to analyze performance issues. Implement and test a TCP variant like TCP Vegas or Westwood to observe improvements.

10. Basic Wireless Security: Configure and test different security protocols on a WLAN (WEP, WPA2, WPA3) and use Wireshark to analyze the handshake process and encrypted traffic.
11. Bluetooth PAN Setup: Establish a Bluetooth Personal Area Network (PAN) between two devices and demonstrate a simple data transfer application.
12. Wireless Sensor Network (WSN) Simulation: Simulate a simple WSN with sensor nodes and a sink to collect and visualize environmental data like temperature.
13. WAP/WML Application: Develop a basic Wireless Application Protocol (WAP) page using Wireless Markup Language (WML) to display information on a mobile emulator.
14. MANET Routing - AODV: Simulate a Mobile Ad-hoc Network (MANET) and implement the Ad-hoc On-Demand Distance Vector (AODV) routing protocol. Analyze route discovery and maintenance.
15. MANET Routing - DSR: Implement the Dynamic Source Routing (DSR) protocol in a MANET simulation and compare its routing overhead and performance with AODV.
16. MANET Routing - OLSR: Configure the Optimized Link State Routing (OLSR) protocol in a simulator and analyze its proactive nature in maintaining routes compared to reactive protocols.
17. Push Service Demonstration: Develop a simple mobile application that receives and displays push notifications from a server (e.g., using Firebase Cloud Messaging).
18. Pull Service Demonstration: Develop a simple mobile application that periodically pulls data from a remote server (REST API) to update its content.
19. Location-Based Service App: Develop a basic Android application that uses the device's GPS to display its current latitude and longitude.
20. Comprehensive Project: Design and simulate a complete mobile computing scenario (e.g., a VANET for traffic updates, a IoT-based smart environment) integrating concepts from multiple units.

Reference Books

1. Hansmann, H., Merk, L., Nicklous, M. S., & Stober, T. (2003). Principles of Mobile Computing (2nd ed.). Springer.
2. Kamal, R. (2007). Mobile Computing. Oxford University Press.
3. Pahlavan, K., & Krishnamurthy, P. (2002). Principles of Wireless Networks: A Unified Approach. Prentice Hall.
4. Rappaport, T. S. (2002). Wireless Communications: Principles and Practice (2nd ed.). Prentice Hall.
5. Schiller, J. (2003). Mobile Communications (2nd ed.). Pearson Education.
6. Stallings, W. (2005). Wireless Communications & Networks (2nd ed.). Pearson Education.
7. Stojmenovic, I., & Cacute, A. (Eds.). (2002). Handbook of Wireless Networks and Mobile Computing. Wiley.

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	**CO1**	**3**	**2**	**2**	**1**	**2**	**1**	**2**	**-**	**1**	**-**	**2**	**1**
	CO2	**2**	**3**	**2**	**1**	**2**	**1**	**2**	**-**	**1**	**1**	**2**	**2**
	CO3	**2**	**2**	**3**	**1**	**2**	**1**	**2**	**1**	**2**	**2**	**2**	**2**
	CO4	**2**	**2**	**2**	**2**	**1**	**1**	**2**	**-**	**2**	**1**	**3**	**3**

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSE 2.1T: Data Science **3 credits**

Course Outcome:

Students able to:

CO1: Learn, analyze and interpret data using appropriate tools and techniques to derive actionable insights from structured and unstructured datasets.

CO2: Apply programming skills, especially in Python and libraries like Pandas, NumPy, and Scikit-learn, to solve real-world data-driven problems.

CO3: Build and evaluate predictive models using machine learning algorithms, while understanding the mathematical foundations behind them.

CO4: Demonstrate awareness of ethical, legal, and societal implications of data use, including privacy.

Detailed Syllabus:

Unit 1: Introduction to Data Science **[3L]**

What is Data Science? Applications and Use Cases, Data Science Workflow, Roles: Data Scientist vs Analyst vs Engineer

Unit 2: Data Wrangling and Exploration **[3L]**

Data Types and Structures (CSV, JSON, SQL, etc.), Pandas and NumPy for data manipulation, Handling Missing Data and Outliers, Exploratory Data Analysis (EDA), Data Visualization (Matplotlib, Seaborn)

Unit 3: Statistics and Probability **[3L]**

Descriptive Statistics (mean, median, std. dev), Probability Distributions (normal, binomial, Poisson), Bayes' Theorem, Inferential Statistics: Hypothesis Testing, Confidence Intervals, Correlation and Causation

Unit 4: Programming for Data Science **[4L]**

Python Basics for Data Science, Functions, Loops, List Comprehensions, Writing Clean, Modular Code, Jupyter Notebooks & IDEs, Version Control with Git

Unit 5: Machine Learning Foundations **[5L]**

Supervised vs Unsupervised Learning, Model Evaluation (Train/Test Split, Cross-Validation), Metrics (Accuracy, Precision, Recall, F1 Score), Linear Regression, Logistic Regression, Decision Trees and Random Forests

Unit 6: Advanced Machine Learning **[5L]**

Support Vector Machines (SVM), Clustering (K-Means, DBSCAN), Ensemble Methods, (Boosting, Bagging), Dimensionality Reduction (PCA, t-SNE), Model Tuning and Hyperparameter Optimization (Grid Search, Random Search)

Unit 7: Feature Engineering and Model Deployment **[4L]**

Feature Scaling, Encoding Categorical Variables, Handling Imbalanced Data, Pipelines and Model Persistence (Pickle, Joblib), Model Deployment Basics (Flask, FastAPI), Introduction to MLOps Concepts

Unit 8: Time Series and Forecasting **[4L]**

Time Series Components and Decomposition, ARIMA, SARIMA Models, Seasonality and Trend Analysis, Forecasting Accuracy Metrics (MAE, RMSE), Handling Temporal Data in Pandas

Unit 9: Deep Learning Basics **[4L]**

Neural Networks Overview, Backpropagation and Activation Functions, Introduction to TensorFlow/Keras or PyTorch, CNNs, RNNs (very high-level intro), Use Cases (Image Classification, NLP)

DSE 2.1P: Data Science Lab **1 credit [20L]**

1. Anaconda Environment Setup and Jupyter Notebook Basics
2. Data Import from CSV and JSON Files
3. Connecting to and Querying a SQL Database
4. Data Cleaning: Handling Missing Values and Duplicates
5. Data Transformation with Pandas
6. Exploratory Data Analysis (EDA) with Descriptive Statistics
7. Data Visualization with Matplotlib: Histograms and Box Plots
8. Advanced Visualization with Seaborn: Heatmaps and Pair Plots
9. Calculating and Interpreting Correlation Matrices
10. Implementing and Sampling from Probability Distributions
11. Performing a Statistical Hypothesis Test (T-Test)
12. Building and Evaluating a Linear Regression Model
13. Implementing and Tuning a Decision Tree Classifier
14. Building a Random Forest Model
15. Unsupervised Learning with K-Means Clustering
16. Feature Engineering: Scaling and Encoding
17. Saving and Loading Models with Joblib
18. Creating a Simple Flask API for Model Deployment
19. Time Series Analysis and Forecasting with ARIMA
20. Building a Basic Neural Network with TensorFlow/Keras

References Books:

1. Wes McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, O'Reilly Media.
2. Jake VanderPlas, Python Data Science Handbook: Essential Tools for Working with Data, O'Reilly Media.
3. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media.
4. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer.
5. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, An Introduction to Statistical Learning with Applications in R and Python, Springer.
6. Wes McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, O'Reilly Media.
7. Jake VanderPlas, Python Data Science Handbook: Essential Tools for Working with Data, O'Reilly Media.
8. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media.
9. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer.
10. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, An Introduction to Statistical Learning with Applications in R and Python, Springer.

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	CO1	2	3	3	2	3	-	-	1	-	1	3	-
	CO2	2	3	2	3	3	-	-	1	-	-	2	-
	CO3	3	2	3	3	2	-	1	-	-	-	2	3
	CO4	-	-	-	-	-	2	2	-	-	2	-	-

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSE 2.2T: Pattern Recognition & Image Processing **3 credits**

Course Outcome:

Students able to:

CO1: Learn to design and implement classifiers using methods like Bayes, K-NN, and neural networks.

CO2: Explore modern advancements including SVMs, fuzzy clustering, and neuro-fuzzy systems.

CO3: Understand and review image transforms and Analyze the basic algorithms used for image processing & image compression with morphological image processing.

CO4: Contrast Image Segmentation and Representation and Design & Synthesize Color image processing and its real world applications.

Detailed Syllabus:

Unit 1: Introduction to Pattern Recognition [2L]

Definition and scope; real-world applications (OCR, speech recognition, fingerprint, signature); mathematical preliminaries.

Unit 2: Classification Techniques [4L]

Bayes decision theory, error probability and rate, minimum distance classifiers, Mahalanobis distance; K-NN classifier; Linear discriminant functions, non-linear decision boundaries; Fisher's LDA; perceptrons (single and multilayer); training and test sets; standardization and normalization.

Unit 3: Clustering Techniques [4L]

Distance functions and similarity measures; minimum within-cluster distance criterion; K-means clustering; single and complete linkage clustering; minimum spanning tree (MST), medoids, DBSCAN; data visualization; uniqueness and existence of clusters.

Unit 4: Feature Selection and Extraction [5L]

Problem statement and significance; probabilistic separability-based criterion functions; inter-class distance-based criterion functions; branch and bound algorithm; sequential forward/backward selection algorithms; (l,r) algorithm. Dimensionality reduction and transformation; Principal Component Analysis (PCA), Kernel PCA.

Unit 5: Introduction to Image Processing [4L]

Fundamentals of Digital Image Processing, Image representation, Basic Image transforms, image file format.

Unit 6: Image Enhancement and Filtering in Spatial domain: [4L]

Contrast stretching, Histogram Equalization, Binarization, Mean filter, Order Statistics filters. Filtering in Frequency domain Butterworth filter, Gaussian filter.

Unit 7: Image Restoration and textures [4L]

Image degradation models, Weiner filter. Run Length Coding, Gray-level co-occurrence matrix

Unit 8: Image Segmentation [6L]

Edge detection: Gradient operators, Compass operator, Laplacian operators. LoG operator. Region growing, region splitting and merging. Least Mean Square error line fitting, Eigenvector line fitting, Straight line Hough Transform, Generalized Hough Transform.

Unit 9: Morphological Operators [2L]

Dilation, Erosion, Opening , Closing, Hit-and-Miss transforms, Applications. Image

Compression.

DSE 2.2P: Pattern Recognition & Image Processing Lab **1 credit [20L]**

List of Program:

1. **Bayes Classifier Implementation** — Implement a Bayes classifier and evaluate performance on a simple dataset.
2. **K-Nearest Neighbour (K-NN) Classifier** — Write a program to classify data points using K-NN with varying values of *k*.
3. **Perceptron Classifier** — Implement a single-layer perceptron for binary classification.
4. **Multilayer Perceptron (MLP)** — Use a feedforward neural network for multi-class classification.
5. **K-Means Clustering** — Implement K-means and visualize clusters for a given dataset.
6. **Hierarchical Clustering** — Apply single-linkage and complete-linkage clustering to group patterns.
7. **DBSCAN Clustering** — Implement DBSCAN and compare results with K-means.
8. **Feature Selection** — Apply Sequential Forward Selection (SFS) or Sequential Backward Selection (SBS) on a dataset.
9. **Principal Component Analysis (PCA)** — Implement PCA for dimensionality reduction and visualize transformed data.
10. **Fisher's Linear Discriminant Analysis (LDA)** — Implement LDA for classification and compare results with PCA.
11. **Support Vector Machine (SVM)** — Use SVM for classification on a real-world dataset (e.g., handwritten digits).
12. **Fuzzy C-Means Clustering** — Implement FCM and visualize fuzzy membership of data points.
13. **Neuro-Fuzzy System** — Implement a simple neuro-fuzzy classifier using a given dataset.
14. MATLAB program to extract different Attributes of an Image.
15. MATLAB program for Image Negation.
16. MATLAB program for Power Law Transformation.
17. MATLAB program for Histogram Mapping and Equalization.
18. MATLAB program for Image Smoothening and Sharpening.
19. MATLAB program for Edge Detection using Sobel, Prewitt and Roberts Operators.
20. MATLAB program for Morphological Operations on Binary Images.
21. MATLAB program for Pseudo Coloring.
22. MATLAB program for Chain Coding.
23. MATLAB program for DCT/IDCT Computation.

References Books:

1. Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification* (2nd ed.). John Wiley & Sons.
2. Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition* (2nd ed.). Academic Press.
3. Theodoridis, S., & Koutroumbas, K. (2009). *Pattern Recognition* (4th ed.). Academic Press.
4. Chanda, B., & Majumder, D. D. (2001). *Digital Image Processing and Analysis*. PHI Learning.
5. Gonzales, R. C., & Woods, R. E. (2018). *Digital Image Processing* (4th ed.). Pearson Education.
6. Jahne, B. (2002). *Digital Image Processing* (5th ed.). Springer-Verlag.
7. Jain, A. K. (1989). *Fundamentals of Digital Image Processing*. Prentice Hall.

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	**CO1**	**3**	**2**	**2**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
	CO2	**2**	**3**	**2**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
	CO3	**2**	**2**	**3**	**2**	**2**	**-**	**-**	**-**	**-**	**1**	**-**	**-**
	CO4	**2**	**2**	**3**	**2**	**2**	**-**	**-**	**-**	**-**	**2**	**-**	**1**

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSE 2.3T: Natural Language Processing **4 credits**

Course Outcome:

Students able to:

CO1: Learn to create CORPUS linguistics based on digestive approach (Text Corpus method)

CO2: Learn method for statistical approaches to machine translation.

CO3: Demonstrate the state-of-the-art algorithms and techniques for text-based processing of natural language with respect to morphology.

CO4: Develop Statistical Methods for Real World Applications and explore deep learning based NLP.

Detailed Syllabus:

Unit 1: Introduction to NLP **[3L]**

Introduction to NLP - Various stages of NLP –The Ambiguity of Language: Why NLP Is Difficult Parts of Speech: Nouns and Pronouns, Words: Determiners and adjectives, verbs, Phrase Structure. Statistics Essential Information Theory: Entropy, perplexity, the relation to language, Cross entropy.

Unit 2: Text Pre-processing and Morphology **[5L]**

Character Encoding, Word Segmentation, Sentence Segmentation, Introduction to Corpora, Corpora Analysis. Inflectional and Derivation Morphology, Morphological analysis and generation using Finite State Automata and Finite State transducer.

Unit 3: Language Modelling **[6L]**

Words: Collocations- Frequency-Mean and Variance – Hypothesis testing: The t test, Hypothesis testing of differences, Pearson's chi-square test, Likelihood ratios. Statistical Inference: n -gram Models over Sparse Data: Bins: Forming Equivalence Classes- N gram model - Statistical Estimators- Combining Estimators.

Unit 4: Word Sense Disambiguation **[5L]**

Methodological Preliminaries, Supervised Disambiguation: Bayesian classification, An information theoretic approach, Dictionary-Based Disambiguation: Disambiguation based on sense, Thesaurus based disambiguation, Disambiguation based on translations in a second-language corpus.

Unit 5: Markov Model and POS Tagging **[5L]**

Markov Model: Hidden Markov model, Fundamentals, Probability of properties, Parameter estimation, Variants, Multiple input observation. The Information Sources in Tagging: Markov model taggers, Viterbi algorithm, Applying HMMs to POS tagging, Applications of Tagging.

Unit 6: Probabilistic Context Free Grammars and Probabilistic parsing **[5L]**

The Probability of a String, Problems with the Inside-Outside Algorithm, Parsing for disambiguation, Tree banks, Parsing models vs. language models, Phrase structure grammars and dependency, Lexicalized models using derivational histories, Dependency-based models.

Unit 7: Syntax and Semantics **[6L]**

Shallow Parsing and Chunking, Shallow Parsing with Conditional Random Fields (CRF), Lexical Semantics, Word Net, Thematic Roles, Semantic Role Labelling with CRFs. Statistical Alignment and Machine Translation, Text alignment, Word alignment, Information extraction, Text mining, Information Retrieval, NL interfaces, Sentimental Analysis, Question Answering Systems, Social network analysis.

DSE 2.3P: Natural Language Processing Lab **1 credit**
[20L]

Program Details:

1. **Text Preprocessing** — Implement tokenization, stemming, lemmatization, and stopword removal using Python (NLTK/spaCy).
2. **Corpus Creation and Analysis** — Build a small text corpus and perform frequency analysis, collocation extraction, and concordance generation.
3. **Morphological Analysis** — Implement finite state automata for basic morphological analysis.
4. **Statistical Language Modelling** — Build unigram, bigram, and trigram models and evaluate perplexity.
5. **POS Tagging with Rule-Based Method** — Create a simple POS tagger using a lexicon and rules.
6. **POS Tagging with HMMs** — Implement a Hidden Markov Model POS tagger and apply the Viterbi algorithm.
7. **Word Sense Disambiguation (Supervised)** — Train a classifier (e.g., Naive Bayes) for word sense disambiguation.
8. **Word Sense Disambiguation (Dictionary-Based)** — Implement Lesk's algorithm for WSD.
9. **Probabilistic Parsing** — Implement a basic probabilistic context-free grammar (PCFG) parser.
10. **Named Entity Recognition (NER)** — Perform NER using spaCy or CRF-based models.
11. **Sentiment Analysis** — Implement sentiment classification using logistic regression or an LSTM-based deep learning model.
12. **Machine Translation (Basic)** — Implement a statistical alignment model for word/phrase alignment (IBM Model 1 or basic translation memory).
13. **Question Answering System** — Develop a simple fact-based QA system using text retrieval and extraction techniques.

References Books:

1. Christopher D. Manning and Hinrich Schutze, " Foundations of Natural Language Processing" , 6th Edition, The MIT Press Cambridge, Massachusetts London, England, 2003
2. Daniel Jurafsky and James H. Martin "Speech and Language Processing", 3rd edition, Prentice Hall, 2009.

3. Nitin Indurkhya, Fred J. Damerau "Handbook of Natural Language Processing", Second Edition, CRC Press, 2010.
4. James Allen "Natural Language Understanding", Pearson Publication 8th Edition. 2012.
5. Chris Manning and Hinrich Schütze, "Foundations of Statistical Natural Language Processing", 2nd edition, MIT Press Cambridge, MA, 2003.
6. Hobson lane, Cole Howard, Hannes Hapke, "Natural language processing in action" MANNING Publications, 2019.
7. Alexander Clark, Chris Fox, Shalom Lappin, "The Handbook of Computational Linguistics and Natural Language Processing", Wiley-Blackwell, 2012
8. Rajesh Arumugam, Rajalingappa Shanmugamani "Hands-on natural language processing with python: A practical guide to applying deep learning architectures to your NLP application". PACKT publisher, 2018.
9. Christopher D. Manning and Hinrich Schütze – *Foundations of Statistical Natural Language Processing*, MIT Press, 2003.
10. Daniel Jurafsky and James H. Martin – *Speech and Language Processing*, 3rd Edition, Prentice Hall, 2009.
11. Nitin Indurkhya, Fred J. Damerau – *Handbook of Natural Language Processing*, 2nd Edition, CRC Press, 2010.
12. Hobson Lane, Cole Howard, Hannes Hapke – *Natural Language Processing in Action*, Manning Publications, 2019.
13. Rajesh Arumugam, Rajalingappa Shanmugamani – *Hands-On Natural Language Processing with Python*, Packt Publishing, 2018.

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	**CO1**	**3**	**2**	**1**	**2**	**2**	**1**	**2**	**-**	**1**	**-**	**2**	**1**
	CO2	**2**	**3**	**2**	**2**	**2**	**1**	**2**	**-**	**1**	**-**	**2**	**2**
	CO3	**2**	**2**	**3**	**2**	**3**	**1**	**2**	**-**	**1**	**1**	**2**	**2**
	CO4	**2**	**2**	**3**	**2**	**3**	**1**	**3**	**-**	**1**	**2**	**2**	**2**

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Indian Knowledge System (IKS) **2 credits**

Course Outcome:

Students able to:

CO1: Learn to analyze India's characteristic knowledge traditions, epistemology, and frameworks of classification of knowledge.

CO2: Demonstrate an understanding of ancient scriptures, educational practices, and their relevance in shaping sciences and society.

CO3: Evaluate scientific, literary, and artistic contributions of IKS torch-bearers across diverse disciplines such as astronomy, mathematics, medicine, architecture, language, and performing arts.

CO4: Apply insights from IKS in areas of governance, cognitive science, logic, and sustainable practices to envision future directions for knowledge and societal development.

Details Syllabus:

Unit 1: History of Indian Knowledge System

Genesis of Bhartiya Knowledge System, History of IKS

Unit 2: India's characteristic knowledge & India's epistemology

Nature, Philosophy and Character, India's Epistemology, Knowledge Frameworks & Classification

Unit 3: Ancient Scriptures

Ancient Scriptures

Unit 4: Ancient Education System

Ancient Education, Educating Sciences

Unit 5: Scientific approaches of IKS & Torch-bearers

Khagol Vijnana (Astronomy), Vastukala (Architecture), Ayurveda, Krishi Vijnana (Agricultural) Practices,

Unit 6: Scientific approaches of IKS & Torch-bearers

Dhatu Vijnana (Metallurgy), Ganita: Mathematics in India, Yuddha Vidhya (Military Sciences), Niyuddha Kala (Martial Arts), Environmental Sciences

Unit 7: Literary Aspects of IKS & Torch-bearers.

Chandashastra (Prosody), Bhasa Va Vyakarana (Language and Grammar), Bharata's Natyashastra (Science of Drama, Dance and Music)

Unit 8: Governance in IKS & Way Forward

Science of Consciousness in Ancient India (Cognitive Science), Anviksiki (Logic and Disputation), Governance & Public Administration, IKS way forward

Reference Books:

1. Introduction to Indian Knowledge System: Concepts and Applications, Archak, K.B. (2012). Kaveri Books, New Delhi.ISBN-13:978-9391818203
2. Introduction To Indian Knowledge System: Concepts and Applications, Mahadevan, B.Bhat, Vinayak Rajat,Nagendra Pavana R.N.PHI, ISBN: 9789391818203
3. Glimpse into Kautilya's Arthashastra Ramachandrudu P. (2010), Sanskrit Academy, Hyderabad ISBN:9788380171074
4. "Introduction" in Studies in Epics and Purāṇas, (Eds.), KM Munshi and N Chandrashekara Aiyer Bhartiya Vidya Bhavan

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12

DSC 1 (T+P)	**CO1**	**2**	**2**	**1**	**1**	**-**	**2**	**2**	**-**	**1**	**2**	**-**	**1**
	CO2	**2**	**1**	**1**	**-**	**-**	**2**	**2**	**-**	**1**	**2**	**-**	**1**
	CO3	**2**	**2**	**2**	**1**	**1**	**2**	**2**	**-**	**2**	**3**	**-**	**1**
	CO4	**1**	**2**	**2**	**1**	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Vidyasagar : Life and philosophy **Compulsory non- credits**

Classes will be held centrally.

Semester - II

DSC 4T: Artificial Intelligence **3 credits**

Course Outcome:

Students able to:

CO1: Explain the fundamentals of Artificial Intelligence, its applications, intelligent agents, and their behavior in different environments.
CO2: Apply various problem-solving and search techniques, including uninformed, heuristic, and metaheuristic algorithms, to AI-related problems.
CO3: Represent knowledge using predicate logic, semantic networks, frames, production rules, conceptual graphs, and ontologies.

CO4: Analyze and apply planning concepts and reasoning techniques under certainty and uncertainty, including probability, Bayes' rule, fuzzy logic, and partial-order planning.

Details Syllabus:

Unit 1: **[4 L]**

Introduction: Introduction to Artificial Intelligence, various definitions of AI, AI Applications and Techniques, Turing Test and Reasoning - forward & backward chaining.

Unit 2: **[3 L]**

Intelligent Agents: Introduction to Intelligent Agents, Rational Agent, their structure, reflex, model-based, goal-based, and utility-based agents, behaviour and environment in which a particular agent operates.

Unit 3: **[10 L]**

Problem Solving and Search Techniques: Problem Characteristics, Production Systems, Control Strategies, Breadth First Search, Depth First Search, iterative deepening, uniform cost search, Hill climbing and its Variations, simulated annealing, genetic algorithm search; Heuristics Search Techniques: Best First Search, A* algorithm, AO* algorithm, Minimax & game trees, refining minimax, Alpha – Beta pruning, Constraint Satisfaction Problem, Means-End Analysis.

Unit 4: **[7 L]**

Knowledge Representation: Introduction to First Order Predicate Calculus, Resolution Principle, Unification, Semantic Nets, Conceptual Dependencies, semantic networks, Frames system, Production Rules, Conceptual Graphs, Ontologies.

Unit 5: **[5 L]**

Planning: Basic representation for planning, symbolic-centralized vs. reactive distributed, partial order planning algorithm

Unit 6: **[6 L]**

Reasoning with Uncertain Knowledge: Different types of uncertainty - degree of belief and degree of truth, various probability constructs - prior probability, conditional probability, probability axioms, probability distributions, and joint probability distributions, Bayes' rule, other approaches to modelling uncertainty such as Dempster-Shafer theory and fuzzy sets/logic.

DSC 4P: Artificial Intelligence Lab **1 credits** **[20L]**

Program Details:

List of Experiment:

1. Study of PROLOG Programming language and its Functions. Write simple facts for the statements using PROLOG.
2. Write a program in PROLOG to implement simple arithmetic.
3. Write a program to implement Factorial, Fibonacci of a given number.
4. Write predicates One converts centigrade temperatures to Fahrenheit, the other checks if a temperature is below freezing.

5. Write a program to Implement Depth First Search using PROLOG.
6. Implementation of Breadth First Search for Tic-Tac-Toe problem.
7. Write a program to solve N-Queen problem
8. Write a program to solve water jug problem using PROLOG.
9. Write a program to solve the Monkey Banana problem.
10. Implementation of Traveling Salesman Problem.
11. Solve 8-puzzle problem using Best First Search.

Reference Books:

1. John L. Hennessy and David A. Patterson
   Title: Computer Architecture: A Quantitative Approach, Morgan Kaufmann
2. A foundational and widely used book covering advanced architecture, performance analysis, ILP, memory systems, and multiprocessors.
3. Kai Hwang
   Title: Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill
4. A comprehensive text focused on parallel processing, memory hierarchies, vector and multiprocessor systems.
5. David A. Patterson and John L. Hennessy
   Title: Computer Organization and Design: The Hardware/Software Interface, Morgan Kaufmann

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	**CO1**	**3**	**2**	**1**	**1**	**1**	**2**	**2**	**-**	**1**	**-**	**1**	**2**
	CO2	**3**	**3**	**2**	**2**	**2**	**2**	**2**	**-**	**1**	**-**	**1**	**2**
	CO3	**3**	**2**	**2**	**1**	**1**	**2**	**3**	**-**	**1**	**-**	**2**	**2**
	CO4	**2**	**3**	**2**	**3**	**2**	**2**	**3**	**1**	**-**	**1**	**2**	**3**

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSE 3.1T: Internet of Things **3 credits**

Course Outcomes:

Students able to:

CO1: Design and develop IoT-based prototypes using sensing, actuation, processing, and communication modules, and explain IoT sensors, wireless communication methods, along with technological challenges related to energy, power, and sensing. **CO2:** Describe the IoT value chain, including devices, cloud data management, application areas, and related technologies, and implement IoT applications using platforms such as

Raspberry Pi.

CO3: Explain the fundamentals of embedded systems, including microcontrollers, devices, buses, and networking in embedded environments, and apply programming and system control concepts to perform specific tasks in embedded systems.

CO4: Develop embedded system programs for various applications and use circuit emulators for testing and development.

Details Syllabus:

Unit 1: Introduction of Internet of Things (IoT): [6L]

Vision, Definition, Conceptual Framework,Architectural view, technology behind IoT,Sources of the IoT, M2M Communication, IoT Examples. Design Principles for Connected devices:IoT/M2M systems layers and design standardization, communication technologies.

Unit 2: Hardware for IoT: [6L]

Sensors, Digital sensors, actuators, radio frequency identification (RFID) technology, wireless sensor networks, participatory sensing technology.

Unit 3: Embedded Platforms for IoT: [6L]

Embedded computing basics, Overview of IOT supported Hardware platforms such as Arduino, Netduino, Raspberry pi, Beagle Bone, Intel Galileo boards and ARM cortex.

Unit 4: Network & Communication aspects in IoT: [6L]

Wireless Medium access issues, MAC protocol survey, Survey routing protocols, Sensor deployment & Node discovery, Data aggregation &dissemination.

Unit 5: Programming the Arduino: [6L]

Arduino Platform Boards Anatomy, Arduino IDE, coding, using emulator, using libraries, additions in Arduino, programming the Arduino for IoT.

Unit 6: Challenges in IoT Design challenges: [5L]

Development Challenges, Security Challenges, Other challenges IoT Applications: Smart Metering, E-health, City Automation, Automotive Applications, home automation, smart cards, communicating data with H/W units, mobiles, tablets, Designing of smart street lights in smart city.

DSE 3.1P: Internet of Things Lab **1 credits [20L]**

Program Details:

1. Introduction of Arduino IDE, demonstrate setup (), loop (),serial and serial.begin(), serial.print(), serial.available(),serial.read(),serial.write(),serial.analogRead() ,user defined functions.
2. Write an Arduino program to demonstrate data types, variables, constants, and operators.

3. Write an Arduino program to demonstrate if statements, switch case, loops, arrays.
4. Write an Arduino program to demonstrate strings, string object, time based functions, random numbers generation
5. Write an Arduino program to demonstrate digital I/O functions, analog I/O functions
6. Write an Arduino program to demonstrate light and LED, the 7-segment display, button, switch. Write an Arduino program to demonstrate interrupts, UART communication protocol .
7. Write an Arduino program to demonstrate I2C communication protocol .
8. Write an Arduino program to demonstrate SPI communication protocol .
9. Write an Arduino program for interfacing with potentiometer, temperature sensor, PIR sensor.Write an Arduino program for interfacing with infrared and ultrasonic sensor, accelerometer, PWM.
10. Write an Arduino program for interfacing with servo motor, stepper motor, DC motor.

Reference Books:

1. Raspberry Pi Cookbook, Software and Hardware Problems and solutions, Simon Monk, O'Reilly (SPD), 2016, ISBN 7989352133895
2. Internet of Things - A Hands-on Approach, ArshdeepBahga and Vijay Madisetti, Universities Press, 2015, ISBN: 9788173719547
3. Getting Started with Raspberry Pi, Matt Richardson & Shawn Wallace, O'Reilly (SPD), 2014, ISBN: 9789350239759
4. Programming Embedded Systems in C and C++, First Edition January, Michael Barr, O' Reilly Introduction to embedded systems, Shibu K V Tata McGraw-Hill.
5. Fundamentals of Microcontrollers and Applications In Embedded Systems" by ameshGaonkar 6. Embedded System Design by Peter Marwedel

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	CO1	3	2	3	2	3	1	1	1	1	1	1	1
	CO2	2	2	2	1	2	1	1	1	1	-	-	1
	CO3	3	2	2	1	2	-	-	-	-	-	-	-
	CO4	2	2	3	2	3	-	-	-	-	-	-	-

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSE 3.2T: Signal Processing **4 credits**

Course Outcomes:

Students able to:

CO1: After the completion of the course the student will be able to : Illustrate digital signals, systems and their significance.

CO2: Analyse the digital signals using various digital transforms DFT, FFT etc.

CO3: Design and develop the basic digital system.

CO4: Interpret the finite word length effects on functioning of digital filters.

Program Details:

Unit 1: Basic elements of digital signal Processing:

Concept of frequency in continuous time and discrete time signals –Sampling theorem Discrete time signals. Discrete time systems –Analysis of Linear time invariant systems –Z transform –Convolution and correlation.

Unit 2: Introduction to DFT:

Efficient computation of DFT Properties of DFT – FFT algorithms – Radix-2 and Radix-4 FFT algorithms – Decimation in Time – Decimation in Frequency algorithms – Use of FFT algorithms in Linear Filtering and correlation.

Unit 3: Structure of IIR:

System Design of Discrete time IIR filter from continuous time filter – IIR filter design by Impulse Invariance. Bilinear transformation – Approximation derivatives – Design of IIR filter in the Frequency domain.

Unit 4: Symmetric & Anti-symmetric FIR filters:

Linear phase filter – Windowing techniques – rectangular, triangular, Blackman and Kaiser windows – Frequency sampling techniques – Structure for FIR systems.

Unit 5: Finite word length effects in FIR and IIR digital filters:

Quantization, round off errors and overflow errors. Multi rate digital signal processing: Concepts, design of practical sampling rate converters, Decimators, interpolators. Polyphasedecompositions.Application of DSP – Model of Speech Wave Form – Vocoder.

Reference Books:

1.Oppenheim A V and Sehafer R W, "Discrete Time Signal Processing", Prentice Hall (1989).

2. Proakis J G and Manolakis D G, "Digital Signal Processing", Pearson Education India.

3.Oppenheim A V, Willsky A S and Young I T, "Signal & Systems", Prentice Hall, (1983).

4. Ifeachor and Jervis, "Digital Signal Processing", Pearson Education India.

5.DeFatta D J, Lucas J G and Hodgkiss W S, "Digital Signal Processing", J Wiley and

Sons, Singapore, 1988

6. Sanjit K Mitra "Digital Signal Processing" TMH

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	CO1	3	2	2	1	2	-	1	-	-	-	-	-
	CO2	3	3	2	2	2	-	1	-	-	-	-	-
	CO3	2	2	3	2	3	-	-	-	-	-	-	-
	CO4	2	2	2	2	2	-	-	-	-	-	-	-

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSE 3.3T: Cryptography and Steganography **3 credits**

Course outcome:

Students able to:

CO1: Ability to analyze and determine for any organization the security requirements and appropriate solutions., Ability to protect system from different types of threats, malicious software's vulnerabilities and attacks.

CO2: Ability to describe symmetric and public key encryption algorithms like DES, AES, RSA etc.

CO3: Ability to identify ethical, professional responsibilities, risks and liabilities in computer and network environment, and best practices to write security policy. Ability to narrate the Authentication of digital certificates.

CO4: Ability to differentiate MAC and hashing techniques needed for authentication. Ability to protect information from different types of threats, malicious software's vulnerabilities and attacks through Data hiding and Watermarking scheme.

Details Syllabus:

Unit 1: Introduction

Introduction to security attacks, services and mechanism, introduction to cryptography. Conventional Encryption: Conventional encryption model, classical encryption techniques-substitution ciphers and transposition ciphers, cryptanalysis, stereography, [6L]

Unit 2: Ciphers

Stream and block ciphers. Modern Block Ciphers: Block ciphers principals, Shannon's theory of confusion and diffusion, feistal structure, data encryption standard (DES), strength of DES, differential and linear crypt analysis of DES. AES. [6L]

Unit 3: Digital Signature

Digital Signature, Authentication, Key Management [5]

Unit 4: Introduction to Information Hiding

Technical Steganography, Linguistic Steganography, Copy Right Enforcement, Wisdom from Cryptography Principles of Steganography: Framework for Secret Communication, Security of Steganography System, Information Hiding in Noisy Data , Adaptive versus non-Adaptive Algorithms, Active and Malicious Attackers, Information hiding in Written Text. [5L]

Unit 5: Steganographic Techniques

Substitution systems and Bit Plane Tools, Transform Domain Techniques: - Spread Spectrum and Information hiding, Statistical Steganography, Distortion Techniques, Cover Generation Techniques.

UNit 6: Steganalysis:

Looking for Signatures: - Extracting hidden Information, Disabling Hidden Information. [10L]

Unit 7: Watermarking and Copyright Protection

Basic Watermarking, Watermarking Applications, Requirements and Algorithmic Design Issues, Evaluation and Benchmarking of Watermarking system. [8L]

Reference Books:

1. William Stallings, "Cryptography and Network Security: Principals and Practice", Prentice Hall, New Jersy.

2. Johannes A. Buchmann, "Introduction to Cryptography", Springer-Verlag.

3. Bruce Schiener, "Applied Cryptography".

4. Katzendbisser, Petitcolas, " Information Hiding Techniques for Steganography and Digital Watermarking", Artech House.

5. Peter Wayner, "Disappearing Cryptography: Information Hiding, Steganography and Watermarking 2/e", Elsevier

6. Bolle, Connell et. al., "Guide to Biometrics", Springer

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	**CO1**	**2**	**3**	**2**	**2**	**2**	**-**	**1**	**-**	**-**	**2**	**-**	**-**
	CO2	**3**	**2**	**2**	**1**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
	CO3	**1**	**2**	**2**	**1**	**1**	**3**	**1**	**-**	**2**	**2**	**-**	**-**
	CO4	**2**	**2**	**2**	**2**	**2**	**-**	**-**	**-**	**-**	**2**	**-**	**-**

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSE 4.1T: Green Computing **3 credit**

Course Outcome:

Students able to:

CO1: Discuss Green IT concepts, its dimensions, strategies, and sustainability trends.
CO2: Describe green devices, hardware, and software methodologies for energy-efficient computing.
CO3: Explain green enterprise activities, functions, and management practices along with necessary components of Green IT.
CO4: Discuss laws, standards, and protocols regulating Green IT and their role in sustainable IT practices.

Detailed Syllabus:

Unit 1: Fundamentals : **[10L]**

Green IT Fundamentals: Business, IT, and the Environment – Green computing: carbon foot print, Green IT Strategies: Drivers, Dimensions, and Goals – Environmentally Responsible Business: Policies, Practices, and Metrics. Environmental Concerns and Sustainable Development, Environmental Impacts of IT, Holistic Approach to Greening IT, Applying IT for enhancing environmental sustainability, Green IT Standards and Eco-Labelling of IT.

Unit 2: Green assets and modelling : **[6L]**

Green Assets: Buildings, Data Centres, Networks, and Devices – Green Business Process Management: Modeling, Optimization, and Collaboration – Green Enterprise Architecture – Environmental Intelligence – Green Supply Chains – Green Information Systems: Design and Development Models.

Unit 3: Green Hardware and Software: **[6L]**

Introduction, Life Cycle of a device or hardware, Reuse, Recycle and Dispose.
Green Software: Introduction, Energy-saving software techniques, Evaluating and Measuring software Impact to platform power.

Unit 4: Grid Framework: **[8L]**

Virtualization of IT systems – Role of electric utilities, Telecommuting, teleconferencing and teleporting – Materials recycling – Best ways for Green PC – Green Data centre – Green Grid framework.

Unit 5: Green Compliance : **[5L]**

Socio-cultural aspects of Green IT – Green Enterprise Transformation Roadmap – Green Compliance: Protocols, Standards, and Audits – Emergent Carbon Issues: Technologies and Future.

1.

DSE 4.1P: Green Computing Lab **1 credits [20L]**

Program Details:

1. Working as Motivators under the Swatch Bharat Campaign of the Government.
2. Literacy drive : Teaching in the Charitable School
3. Enroll as NSS Volunteers for various projects (Cleanliness, Women/child health awareness).
4. Counseling camps in villages
5. Tree plantation
6. Enroll in the Gandhian Studies Centre as student Volunteer for surveys in villages.
7. Women Empowerment Programmes.
8. Environment Awareness (Reduce Pollution)
9. Disaster Management/Relief Work.

References Books:

2. Harnessing Green IT Principles and Practices , San Murugesan, G.R. Gangadharan Wiley Publication, ISBN:9788126539680
3. Bhuvan Unhelkar, ―Green IT Strategies and Applications-Using Environmental Intelligence, CRC Press, June 2014.
4. Woody Leonhard, Katherine Murray, ―Green Home computing for dummies, August 2012.

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	CO1	1	1	-	-	-	-	2	-	-	3	-	-
	CO2	2	2	2	-	2	-	1	-	-	3	-	-
	CO3	1	2	2	1	1	1	2	1	1	3	1	-
	CO4	-	1	1	-	-	2	1	-	-	3	-	-

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSE 4.2T: Soft Computing **3 credits**

Course Outcome:

Students able to:

CO1: Apply neural networks, bidirectional associative memories, and adaptive resonance theory to solve different engineering problems.
CO2: Identify and describe soft computing techniques and build supervised and unsupervised learning networks.
CO3: Apply fuzzy logic, reasoning, and genetic algorithms to handle uncertainty and solve optimization problems.
CO4: Evaluate and compare solutions for engineering problems using various soft computing approaches

Detailed Syllabus:

Unit 1: Introduction to Soft Computing **[4 L]**

Soft computing vs. hard computing, evolution of soft computing, features and types of soft computing, applications of soft computing, basics of machine learning.

Unit 2: Neural Networks and Back Propagation networks **[8 L]**

Basic concepts of Neural Networks, Model of Artificial Neuron, Neural Network Architectures, Characteristics of neural networks, Learning Methods, Early neural network architectures, Application domains. Back propagation network (BPN), Back propagation Learning, Applications of BPN, Parameter selection, Variations of Back propagation Algorithms.

Unit 3: Unsupervised learning: Adaptive Resonance **[7 L]**

Adaptive Resonance Theory (ART), Classical ART Networks, Simplifies ART Architecture, Features, algorithms and Illustration of ART1 and ART2 model, Related Applications.

Unit 4: Fuzzy Sets and Fuzzy Relations **[5 L]**

Fuzzy versus Crisp, Crisp Sets, Fuzzy sets, Membership functions, fuzzy set operations, properties of Fuzzy sets, Crisp Relations, Fuzzy relations – Fuzzy Cartesian product, Operations of Fuzzy Relations.

Unit 5: Fuzzy Logic and Inference **[4 L]**

Crisp Logic, Predicate Logic, Fuzzy Logic, Fuzzy Quantifiers, Fuzzy Inference, Fuzzy knowledge and rule-based system, fuzzy decision making, Defuzzification, and Application of fuzzy logic.

Unit 6: Genetic Algorithms **[7 L]**

History of Genetic Algorithm, Basic concepts, Creation of offspring, working principles, encoding, fitness function, reproduction, Genetic modelling: Inheritance operator, crossover, inversion & deletion, mutation operator, Bitwise operator, Generational Cycle, Convergence of GA, Applications & advances in GA, Differences & similarities between GA & other traditional method, Hybrid systems, evolutionary computing, Genetic

Algorithm based on Back propagation networks- Implementation and comparison on performance of traditional algorithms with Genetic Algorithm.

DSE 4.2P: Soft Computing Lab **1 credit [20L]**

Program Details:

1. **Perceptron Implementation** — Design and train a single-layer perceptron for binary classification.
2. **Multi-Layer Perceptron (MLP)** — Implement an MLP with Back Propagation for function approximation.
3. **BPN for Classification** — Apply Back Propagation Network for a multi-class classification problem (e.g., digit recognition).
4. **Parameter Tuning in BPN** — Study the effect of learning rate, momentum, and number of hidden layers on model performance.
5. **Adaptive Resonance Theory (ART1)** — Implement ART1 for binary input pattern clustering.
6. **Adaptive Resonance Theory (ART2)** — Implement ART2 for real-valued input pattern clustering.
7. **Fuzzy Set Operations** — Implement fuzzy union, intersection, complement, and Cartesian product.
8. **Fuzzy Inference System (FIS)** — Design a Mamdani-type FIS for a decision-making application.
9. **Defuzzification Methods** — Implement and compare defuzzification techniques (Centroid, Mean of Maxima, etc.).
10. **Genetic Algorithm Basics** — Implement GA to solve a simple optimization problem (e.g., function maximization).
11. **GA for Combinatorial Problems** — Apply GA to solve the Travelling Salesman Problem (TSP).
12. **Hybrid GA–BPN System** — Train a neural network using GA for weight optimization and compare with standard backpropagation.

References Books:

1. S, Rajasekaran & G.A. Vijayalakshmi Pai, "Neural Networks, Fuzzy systems and evolutionary algorithms: Synthesis and Applications", PHI Publication, 2nd Ed. 2017.
2. Timothy J. Ross, "Fuzzy Logic with Engineering Applications", John Wiley and Sons, 3rd ed, 2011.
3. 3. S.N. Sivanandam & S.N. Deepa, "Principles of Soft Computing", Wiley Publications, 3rd ed, 2018.
4. Jang, Jyh-Shing Roger, Chuen-Tsai Sun, and Eiji Mizutani. "Neuro-fuzzy and soft computing a computational approach to learning and machine intelligence" Pearson, 1997.

5. Kosko, B., Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence, PHI Publication, 1994.
6. George J. Klir,Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice Hall, 2015
7. Rich E andKnight K, Artificial Intelligence, McGraw Hill Education; 3rd ed, 2017.
8. S. Haykin, "Neural Networks and Learning Machines", Pearson Education Inc., 3rd Ed 2008.
9. Goodfellow, Ian, YoshuaBengio, and Aaron Courville. Deep learning: Adaptive Computation and Machine Learning series, MIT press, 2016.

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	CO1	3	2	2	1	2	-	1	-	-	-	-	-
	CO2	2	2	2	1	2	-	2	-	-	-	-	-
	CO3	2	3	2	1	2	-	2	-	-	-	-	-
	CO4	1	2	2	2	2	-	2	-	-	-	-	-

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSE 4.3T: Quantum Computing **3 credits**

Course Outcome (CO):

Students able to:

CO1: Explain quantum mechanics foundations (qubits, superposition, entanglement) and contrast classical vs. quantum computational models.

CO2: Design and implement quantum circuits using gates (Hadamard, CNOT, Toffoli) and simulate algorithms (Deutsch-Jozsa, Grover's) on quantum SDKs (Qiskit/Cirq).

CO3: Analyze quantum algorithms (Shor's, QFT, VQE) and evaluate their speedup over classical counterparts for cryptography, optimization, and chemistry.

CO4: Assess hardware challenges (decoherence, error correction) and emerging technologies (superconducting, photonic, trapped-ion qubits).

Detailed Syllabus:

Unit 1: Introduction **[6L]**

Quantum Foundations: Qubits, Bloch sphere, Dirac notation, Postulates of quantum mechanics

Entanglement and Bell inequalities

Unit 2: Quantum Circuits **[10L]**

Quantum Circuits & Gates: Universal gate sets (Pauli, Hadamard, CNOT), Quantum teleportation and superdense coding

Unit 3: Algorithms **[7L]**

Quantum Algorithms: Deutsch-Jozsa, Bernstein-Vazirani, Grover's search, Shor's factoring

Unit 4: Quantum Techniques **[7L]**

Variational Quantum Eigensolver (VQE), Quantum Machine Learning (QML) basics
Lab: VQE for molecular ground states

Unit 5: Advanced Quantum **[5L]**

Decoherence, T1/T2 times, Surface codes, fault-tolerant QC

DSE 4.3P: Quantum Computing Lab **1 credit [20L]**

Program Details:

Unit 1: Quantum Basics in Practice

Qubit representation and initialization in Qiskit/Cirq, Applying single-qubit gates (X, Y, Z, H, S, T) and visualizing on Bloch sphere, Multi-qubit operations: CNOT, SWAP, Toffoli gates

Unit 2: Quantum States & Measurement

Creating superposition states and entangled states (Bell states, GHZ states), Measurement in different bases, Verification of entanglement via Bell inequality tests

Unit 3: Algorithm Implementations

Deutsch-Jozsa algorithm implementation and verification, Grover's search algorithm for small database problems, Bernstein–Vazirani algorithm

Unit 4: Variational & Learning Approaches

Implementing Variational Quantum Eigensolver (VQE) for simple molecular Hamiltonians, Intro to Quantum Machine Learning (QML) with basic classification example,

Unit 5: Hardware & Noise Considerations

Running circuits on IBM Quantum Experience or other cloud-based QPUs, Exploring noise models and simple error mitigation techniques, Comparing simulated vs real hardware results

References Books:

1. "Quantum Computation and Quantum Information" – Michael Nielsen & Isaac Chuang
2. "Quantum Computing for Computer Scientists" – *Noson S. Yanofsky & Mirco A. Mannucci*
3. Michael A. Nielsen & Isaac L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press.
4. Noson S. Yanofsky & Mirco A. Mannucci, *Quantum Computing for Computer Scientists*, Cambridge University Press.
5. Chris Bernhardt, *Quantum Computing for Everyone*, MIT Press.
6. Jack D. Hidary, *Quantum Computing: An Applied Approach*, Springer.
7. Robert S. Sutor, *Dancing with Qubits*, Packt Publishing.

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12

DSC 1 (T+P)	**CO1**	**3**	**2**	**1**	**1**	**1**	**-**	**2**	**-**	**-**	**-**	**-**	**-**
	CO2	**3**	**2**	**2**	**1**	**3**	**-**	**2**	**-**	**-**	**-**	**-**	**-**
	CO3	**3**	**3**	**2**	**2**	**2**	**-**	**2**	**-**	**-**	**-**	**-**	**1**
	CO4	**2**	**2**	**2**	**2**	**2**	**1**	**2**	**-**	**-**	**1**	**-**	**-**

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSE 5.1T: Cyber Security **3 credits**

Course Outcome (CO):

Students able to:

CO1: Analyse cyber threats, vulnerabilities, and attack vectors to assess risks in information systems and networks.

CO2: Design and implement cryptographic protocols and algorithms to ensure data confidentiality, integrity, and authentication.

CO3: Evaluate and apply security mechanisms for protecting network infrastructure, web applications, and cloud-based systems.

CO4: Develop incident response strategies and utilize forensic tools to investigate and mitigate cyber-attacks.

Detailed Syllabus:

Unit 1: Introduction to Cyber Security **[6L]**

Fundamentals of cyber security: confidentiality, integrity, availability (CIA triad), Types of cyber threats: malware, phishing, ransomware, social engineering, Security policies, risk assessment, and threat modelling, Overview of cyber security frameworks: NIST, ISO/IEC 27001, Legal and ethical issues in cyber security

Unit 2: Cryptography and Secure Communication **[8L]**

Symmetric and asymmetric cryptography: AES, RSA, ECC, Hash functions and digital signatures: SHA, HMAC, Public Key Infrastructure (PKI) and certificate authorities, Secure communication protocols: TLS/SSL, IPsec, SSH, Quantum cryptography basics

Unit 3: Network Security **[7L]**

Network vulnerabilities: DDoS, man-in-the-middle, packet sniffing, Firewalls, intrusion detection and prevention systems (IDPS), Virtual Private Networks (VPNs) and secure tunnelling, Network access control: authentication protocols (Kerberos, OAuth), Tools for network security: Wireshark, Snort, Nmap

Unit 4: Application and Cloud Security **[6L]**

Web application vulnerabilities: SQL injection, XSS, CSRF, Secure software development lifecycle (SDLC), Cloud security models: shared responsibility, zero trust architecture, Securing cloud environments: AWS, Azure, Google Cloud, Container security: Docker, Kubernetes

Unit 5: Cyber Security Operations and Forensics **[8L]**

Incident response lifecycle: preparation, detection, containment, recovery, Digital forensics: evidence collection, chain of custody, forensic tools (Autopsy, FTK), Malware analysis: static and dynamic analysis techniques, Security Information and Event Management (SIEM) systems, Emerging trends: AI-driven security, blockchain for secure transactions

DSE 5.1P: Cyber Security Lab **1 credit [20L]**

Program Details:

Unit 1: Introduction & Environment Setup

Setting up a virtual lab environment using VirtualBox/VMware, Installing Kali Linux, Wireshark, Nmap, and other security tools, Understanding legal & ethical aspects of penetration testing

Unit 2: Network Scanning & Enumeration

Using Nmap for host discovery & port scanning, Identifying open services and potential vulnerabilities, Banner grabbing & service fingerprinting

Unit 3: Cryptography Implementation

Implementing symmetric encryption (AES) and asymmetric encryption (RSA) in Python, Generating & verifying digital signatures, Hashing techniques: SHA-256, HMAC

Unit 4: Secure Communication Protocols

Setting up TLS/SSL for a web server, Testing SSH-based secure remote access, Exploring IPsec VPN configuration

Unit 5: Web Application Security

Testing for SQL Injection & Cross-Site Scripting (XSS) using DVWA (Damn Vulnerable Web Application), Implementing input validation & secure coding practices, Using Burp Suite for web traffic analysis

Unit 6: Network Security Tools

Configuring and using Snort as an Intrusion Detection System, Setting up firewall rules using ip tables or pfSense, Capturing and analyzing network traffic with Wireshark

Unit 7: Cloud & Container Security

Basics of securing AWS/Azure cloud resources, Configuring IAM roles & policies, Securing Docker containers against vulnerabilities

Unit 8: Digital Forensics & Malware Analysis

Using Autopsy for file system forensics, Basic malware analysis (static & dynamic) in a sandbox, Tracing attack footprints using logs and SIEM tools

References Books:

1. William Stallings – *Cryptography and Network Security: Principles and Practice*, Pearson Education.
2. Michael E. Whitman, Herbert J. Mattord – *Principles of Information Security*, Cengage Learning.
3. Chuck Easttom – *Computer Security Fundamentals*, Pearson Education.
4. Matt Bishop – *Computer Security: Art and Science*, Addison-Wesley.
5. Dafydd Stuttard, Marcus Pinto – *The Web Application Hacker's Handbook*, Wiley.

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	CO1	3	3	2	2	2	2	2	–	1	3	1	–
	CO2	3	2	3	2	3	2	2	–	–	2	–	–
	CO3	3	3	3	2	3	2	2	–	–	3	1	2
	CO4	2	2	2	3	2	3	2	2	2	2	2	1

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSE 5.2T: Computer Vision 3 credits

Course Outcome:

Students able to:

CO1: Understand image formation, representation, and processing techniques.
CO2: Extract features and match visual patterns using classical and deep learning methods.
CO3: Apply object detection and segmentation in real-world tasks, and build and evaluate vision systems using modern deep learning frameworks.
CO4: Develop and demonstrate an applied computer vision project using public datasets.

Details Syllabus:

Unit 1: Introduction to Computer Vision [3L]

Difference between Image Processing and Computer Vision, Human vs. Computer Visual Perception, Pinhole Camera Model, Camera Parameters, Applications: Face Recognition, AR/VR, Robotics, Surveillance

Unit 2: Image Representation and Preprocessing [6L]

Image Types: Binary, Grayscale, Color (RGB, HSV, YCbCr), Color Spaces and Transformations, Image Filtering: Smoothing, Sharpening, Gaussian, Median filters, Edge Detection: Sobel, Prewitt, Canny, Morphological operations: Dilation, Erosion, Opening, Closing.

Unit 3: Feature Detection and Matching [6L]

Corner Detection: Harris, FAST, Blob Detection: LoG, DoG, Feature Descriptors: SIFT, SURF, ORB, HOG, Feature Matching: Brute Force, FLANN, RANSAC, Geometric Transformations: Affine, Homography.

Unit 4: Image Segmentation and Region Analysis [6L]

Thresholding: Global, Adaptive, Otsu's method, Region-based segmentation: Region growing, Watershed, Clustering: K-means, Mean-Shift, Superpixel segmentation, Contour Detection and Analysis, Connected Component Labelling.

Unit 5: Deep Learning for Vision Tasks **[8L]**

Introduction to Convolutional Neural Networks (CNNs), CNN Architectures: LeNet, AlexNet, VGG, ResNet, Transfer Learning and Fine-tuning, Object Detection: R-CNN, YOLOv3/v5, SSD, Semantic Segmentation: FCN, U-Net, DeepLab, Tools & Libraries: OpenCV, PyTorch, TensorFlow

Unit 6: Motion Analysis and Object Tracking **[6L]**

Optical Flow: Lucas-Kanade, Horn-Schunck, Background Subtraction: MOG, GMM, Object Tracking Techniques: Kalman Filter, Meanshift, CAMShift, SORT, Action and Gesture Recognition Basics

DSC 11P: Computer Vision Lab **1 credit**
[20L]

Program Details:

Unit 1: Image Representation and Preprocessing

Reading, writing, and displaying images using OpenCV. Conversion between color spaces (RGB, HSV, YCbCr). Applying smoothing and sharpening filters (Gaussian, Median, Laplacian). Edge detection using Sobel, Prewitt, and Canny operators. Performing morphological operations (dilation, erosion, opening, closing).

Unit 2: Feature Detection and Matching

Harris and FAST corner detection. Blob detection using LoG and DoG. Extracting SIFT, SURF, ORB, and HOG features. Feature matching using Brute Force and FLANN. Implementing RANSAC for geometric transformations.

Unit 3: Image Segmentation

Global, adaptive, and Otsu thresholding. Region-based segmentation (region growing, watershed).

Clustering-based segmentation (K-means, Mean Shift, Superpixels). Contour detection and connected component analysis.

Unit 4: Deep Learning for Vision

Building simple CNNs for image classification (MNIST/CIFAR datasets). Transfer learning with pre-trained networks (VGG, ResNet). Object detection using YOLO or SSD with public datasets. Semantic segmentation using U-Net or DeepLab.

Unit 5: Motion Analysis and Object Tracking

Optical flow computation (Lucas-Kanade, Farneback). Background subtraction using MOG and GMM. Object tracking with Meanshift, CAMShift, and SORT.

Unit 6: Project Work

Students will design, implement, and present a complete computer vision application (e.g., face recognition, OCR, medical image analysis, gesture recognition) using real-world datasets.

Reference Books:

1. "Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods
Publisher: Pearson Education
2. "Computer Vision: A Modern Approach" by David A. Forsyth and Jean Ponce
Publisher: Pearson Education
3. "Digital Image Processing and Computer Vision" by Sonka, Hlavac, Boyle
Publisher: Cengage (Earlier Tata McGraw-Hill)
4. "Learning OpenCV 4: Computer Vision with Python" by Adrian Kaehler and Gary Bradski
Publisher: O'Reilly Media (Distributed by PSI in India)
5. "Pattern Recognition and Machine Learning" by Christopher M. Bishop
Publisher: Springer (often distributed in India by PHI)
6. "Fundamentals of Digital Image Processing" by Anil K. Jain
Publisher: PHI Learning / Pearson
7. "Computer Vision: Algorithms and Applications" by Richard Szeliski
Publisher: Springer
8. "Deep Learning" by Ian Goodfellow, Yoshua Bengio, Aaron Courville
Publisher: MIT Press (International edition often available via PSI)
9. "Machine Learning" by Tom M. Mitchell *Publisher: McGraw-Hill*

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	**CO1**	**3**	**2**	**2**	**2**	**3**	**–**	**2**	**–**	**1**	**–**	**–**	**–**
	CO2	**3**	**3**	**2**	**2**	**3**	**–**	**2**	**–**	**–**	**–**	**1**	**–**
	CO3	**3**	**3**	**3**	**2**	**3**	**1**	**2**	**–**	**2**	**2**	**2**	**2**
	CO4	**2**	**2**	**3**	**3**	**3**	**1**	**3**	**2**	**3**	**2**	**3**	**2**

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSE 5.3T: Big Data Analytics **3 credits**

Course Outcome (CO):

Students able to:

CO1: Analyze the fundamental concepts, characteristics, and challenges of Big Data in various application domains.

CO2: Design and implement scalable storage solutions for Big Data using distributed file systems and NoSQL databases.

CO3: Develop and optimize data processing workflows using batch and stream processing frameworks such as Hadoop and Spark.

CO4: Apply advanced analytics techniques, including machine learning and data mining, to extract actionable insights from large datasets.

Detailed Syllabus:

Unit 1: Introduction to Big Data **[6L]**

Definition and characteristics of Big Data (volume, velocity, variety, veracity, value), Big Data ecosystem: hardware, software, and network considerations, Challenges in Big Data: storage, processing, analytics, security, and privacy, Opportunities and applications in domains such as healthcare, finance, social media, and IoT

Unit 2: Big Data Storage **[7L]**

Distributed file systems: HDFS, Google File System, NoSQL databases: key-value stores (e.g., Redis), document databases (e.g., MongoDB), column-family stores (e.g., Cassandra), graph databases (e.g., Neo4j), Data lakes and cloud storage solutions (e.g., AWS S3, Google Cloud Storage), Data formats and serialization techniques: Avro, Parquet, ORC

Unit 3: Big Data Processing **[8L]**

Batch processing frameworks: Hadoop MapReduce, Apache Hive, Apache Pig, In-memory processing with Apache Spark: RDDs, DataFrames, Spark SQL, Stream processing technologies: Apache Kafka, Apache Storm, Apache Flink, Workflow management tools: Apache Oozie, Apache Airflow

Unit 4: Big Data Analytics Techniques **[8L]**

Machine learning for Big Data: supervised and unsupervised learning algorithms, Spark MLlib, Data mining techniques for large datasets: association rule mining, clustering, classification, Graph analytics: algorithms for large graphs, Apache GraphX
Visualization tools for Big Data: Tableau, Power BI, Apache Zeppelin

Unit 5: Applications **[6L]**

Big Data applications in healthcare: patient data analysis, disease prediction, Finance: fraud detection, risk management, social media: sentiment analysis, trend detection, Big Data in IoT: sensor data processing, predictive maintenance, Ethical considerations, data governance, and privacy issues in Big Data

DSE 5.3P: Big Data Analytics Lab **1 credit**
[20L]

Program Details:

Exercise 1: Installation and configuration of Hadoop ecosystem (HDFS, YARN) on a single-node or pseudo-distributed cluster.

Exercise 2: Perform basic HDFS operations: uploading, downloading, listing, and deleting files.

Exercise 3: Implement a simple Hadoop MapReduce program (e.g., Word Count).

Exercise 4: Use Apache Hive to create tables, load data, and run SQL-like queries on large datasets.

Exercise 5: Implement data processing using Apache Pig scripts.

Exercise 6: Introduction to Apache Spark:

- Load data into RDDs and DataFrames.
- Perform basic transformations and actions.

Exercise 7: Use Spark SQL for structured data analysis.

Exercise 8: Implement a real-time data streaming pipeline using Apache Kafka and Spark Streaming.

Exercise 9: Apply Spark MLlib for a supervised learning task (e.g., classification) on a large dataset.

Exercise 10: Perform association rule mining or clustering on Big Data using Spark MLlib.

Exercise 11: Work with NoSQL databases (MongoDB / Cassandra) for storing and querying data.

Exercise 12: Use a Big Data visualization tool (Tableau / Apache Zeppelin) to create dashboards for insights.

Reference Books:

1. Thomas Erl, Wajid Khattak, Paul Buhler - Big Data Fundamentals: Concepts, Drivers & Techniques, Prentice Hall.
2. Tom White - Hadoop: The Definitive Guide, O'Reilly Media.
3. Bill Chambers, Matei Zaharia - Spark: The Definitive Guide, O'Reilly Media.
4. Foster Provost, Tom Fawcett - Data Science for Business, O'Reilly Media.
5. Jure Leskovec, Anand Rajaraman, Jeffrey Ullman - Mining of Massive Datasets, Cambridge University Press.

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	**CO1**	**3**	**2**	**1**	**2**	**2**	**–**	**2**	**–**	**1**	**1**	**–**	**–**
	CO2	**3**	**3**	**2**	**2**	**3**	**–**	**2**	**–**	**–**	**–**	**1**	**–**
	CO3	**3**	**3**	**3**	**2**	**3**	**–**	**2**	**–**	**–**	**1**	**2**	**2**
	CO4	**3**	**3**	**3**	**3**	**3**	**1**	**3**	**2**	**2**	**2**	**2**	**2**

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSC 5T: Machine Learning **3 credits**

Course Outcome:

Students able to:

CO1: Understand and explain core concepts, types, and real-world applications of machine learning, along with the mathematical foundations of common algorithms.
CO2: Implement and evaluate machine learning models using Python libraries such as

Scikit-learn, and gain exposure to advanced tools like TensorFlow or PyTorch.
CO3: Apply machine learning techniques to solve problems across various domains, including data preprocessing, model selection, and interpretation of results.
CO4: Analyze and select appropriate algorithms based on problem type, data characteristics, and performance metrics, while appreciating the role of data, algorithmic thinking, and system design in building effective, scalable, and responsible ML solutions.

Details Syllabus:

Unit 1: Foundations of Machine Learning **[3L]**

What is Machine Learning? Types of ML: Supervised, Unsupervised, Reinforcement Learning, Key ML Terminologies: features, labels, models, training, testing, ML Project Lifecycle and Pipeline Overview

Unit 2: Mathematics for Machine Learning **[3L]**

Linear Algebra: Vectors, matrices, dot products, Probability & Statistics: Mean, variance, Bayes' Theorem, distributions, Calculus: Derivatives, gradients, basics of optimization, Cost and Loss Functions

Unit 3: Data Preprocessing and Exploration **[4L]**

Data collection and loading, Exploratory Data Analysis (EDA), Handling missing data and outliers, Feature engineering and encoding, Data normalization and standardization, Train-test split and validation strategies

Unit 4: Supervised Learning - Regression **[4L]**

Linear Regression (simple & multivariate), Polynomial Regression, Regularization: Ridge, Lasso, ElasticNet, Evaluation metrics: MAE, MSE, RMSE, R^2

Unit 5: Supervised Learning - Classification **[4L]**

Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees, Support Vector Machines (SVM), Evaluation metrics: Accuracy, Precision, Recall, F1 Score, Confusion Matrix, ROC-AUC

Unit 6: Ensemble Learning Methods **[4L]**

Bagging vs Boosting, Random Forest, Gradient Boosting Machines (GBM), XGBoost, LightGBM, CatBoost, Feature importance and interpretability

Unit 7: Unsupervised Learning **[5L]**

Clustering: K-Means, DBSCAN, Hierarchical Clustering, Dimensionality Reduction: PCA, t-SNE, UMAP, Anomaly Detection and Applications, Association Rule Learning (Apriori, Eclat)

Unit 8: Model Evaluation and Optimization **[3L]**

Cross-validation (K-Fold, Stratified), Hyperparameter tuning: Grid Search, Random Search, Bias-Variance Tradeoff, Underfitting vs Overfitting, Model selection strategies

Unit 9: Intro to Neural Networks (Bridge to Deep Learning) **[3L]**

Perceptron and Multi-layer Perceptron (MLP), Forward and Backpropagation, Activation Functions, Use cases: classification, regression, Brief intro to TensorFlow or PyTorch

Unit 10: Time Series and Sequential Data **[2L]**

Time series components: trend, seasonality, noise, Lag features and rolling statistics, ARIMA and forecasting basics, Evaluating forecasting models

DSC 5P: Machine Learning Lab **3 credits [20L]**

Program Details:

Unit 1: Python & ML Environment Setup

Setting up Python environment for ML (Anaconda, Jupyter Notebook, VS Code), Introduction to NumPy, Pandas, Matplotlib, Seaborn for data handling & visualization.

Unit 2: Data Preprocessing & EDA

Importing datasets and performing exploratory data analysis (EDA), Handling missing data, outliers, categorical encoding, normalization, and standardization.

Unit 3: Regression Models

Implementing Simple & Multiple Linear Regression, Implementing Polynomial Regression and regularization techniques (Ridge, Lasso).

Unit 4: Classification Models

Implementing Logistic Regression and K-Nearest Neighbors (KNN), Implementing Decision Trees and Support Vector Machines (SVM) with evaluation metrics.

Unit 5: Ensemble Learning

Implementing Random Forest and Gradient Boosting (XGBoost/LightGBM), Feature importance analysis and visualization.

Unit 6: Unsupervised Learning

Implementing K-Means and DBSCAN clustering with visualization, Performing Principal Component Analysis (PCA) and t-SNE for dimensionality reduction.

Unit 7: Model Optimization

Hyperparameter tuning with Grid Search and Random Search, Cross-validation and performance comparison of models.

Unit 8: Intro to Neural Networks

Implementing a simple Multi-layer Perceptron (MLP) using TensorFlow/Keras.

Unit 9: Time Series Analysis

Building a simple ARIMA forecasting model and evaluating results.

Unit 10: Model Deployment

Saving and loading ML models with Pickle/Joblib, Deploying an ML model using Flask or FastAPI.

References Books:

1. Christopher M. Bishop – Pattern Recognition and Machine Learning – Springer, 2006.
2. Trevor Hastie, Robert Tibshirani, Jerome Friedman – The Elements of Statistical Learning: Data Mining, Inference, and Prediction – Springer, 2nd Edition, 2009.
3. Kevin P. Murphy – Machine Learning: A Probabilistic Perspective – MIT Press, 2012.
4. Ian Goodfellow, Yoshua Bengio, Aaron Courville – Deep Learning – MIT Press, 2016.
5. Aurélien Géron – Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow

– O’Reilly Media, 3rd Edition, 2022.

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	CO1	3	2	1	2	2	–	2	–	1	–	–	–
	CO2	3	3	2	2	3	–	2	–	–	–	1	–
	CO3	3	3	3	2	3	–	2	–	–	1	2	1
	CO4	3	3	3	3	3	1	3	2	2	1	2	2

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Field Visit/ Industry Visit/ Case Study/ Hands-on Practical/ Skill Enhanced Course
2 credits

Semester - III

DSC 6T: Advanced Data Structure and Algorithm **3 credits**

Course Outcome:

Students able to:

CO1: Solve real-life computational problems by selecting suitable data structures and algorithmic strategies, analyzing algorithm complexity, and justifying decisions using Big-O notation.

CO2: Implement basic and advanced data structures such as arrays, stacks, queues, linked lists, trees, heaps, hash tables, and graphs.

CO3: Design and implement efficient searching and sorting techniques, and apply paradigms like dynamic programming, greedy strategies, and divide-and-conquer to optimize solutions.

CO4: Demonstrate the ability to write modular, readable, and reusable code, and compare algorithmic solutions based on time-space trade-offs to guide system design.

Details Syllabus:

Unit 1: **[4L]**

Introduction

Review of Data and Information, Abstract Data Types (ADTs), Array, Linked List, Linear Data Structure-Stack, applications of stack- Infix to Postfix conversion, evaluating postfix expressions, Queue-variants of queue and their applications, Non-linear data Structure - graph, tree.

Searching and Sorting Algorithms - Linear and Binary Search; Sorting – Mergesort, Quick sort, Heap sort; Stable and unstable sort, Internal and external Sort.

Unit 2: **[4L]**

Trees and Graphs - Binary Search Trees, AVL Trees, Red-Black Tree, B-trees & variants. Tree Traversal Algorithms, Heaps and its applications (e.g. implementation of Priority Queue) Graph representations: Adjacency Matrix and Adjacency List, Graph Traversal Algorithms: BFS and DFS, Minimum Cost Spanning Tree, Shortest Path Algorithms.

Hashing - Terminologies, Hash functions, Collision Resolution Strategies, Types of Hashing

Unit 3: **[4L]**

Asymptotic Notation: Big-O, omega, theta etc.; finding time complexity of well-known algorithms.

Algorithm Design Techniques: Recursion- Definition, Use, Limitations, Examples: Tower of Hanoi problem. Tail Recursion.

Divide and Conquer: Basic method, Use, Examples: Merge sort, Quick Sort, Binary Search.

Unit 4: **[7L]**

Dynamic Programming: Basic method, Use, Examples: Matrix-chain multiplication, Longest Common Subsequence, Single-source shortest path, Travelling Salesman problem

Greedy Method: Basic method, use, Examples: Knapsack problem

Branch and Bound: Basic method, use, Examples: The 15-puzzle problem

Unit 5: **[6L]**

Backtracking: Basic method, use, Examples: Eight queens' problem, Graph coloring problem, Hamiltonian problem

Disjoint Set Manipulation: Set manipulation algorithm like UNION-FIND, union by rank, Path compression.

NP-completeness: P class, NP-hard class, NP-complete class, Circuit Satisfiability problem, Clique Decision Problem.

Unit 6: **[10L]**

Approximation algorithms: Necessity of approximation scheme, performance guarantee, Polynomial time approximation schemes: 0/1 knapsack problem.

Network Algorithms

Routing Algorithms-Dijkstra's Algorithm, Floyd Warshall algorithm for all pair shortest path, Traversal Algorithms - BFS, DFS, Minimum spanning tree algorithms - Prim's and Kruskal's algorithms.

Cryptography algorithms

DES, AES algorithm, RSA Algorithm, Diffie-Hellman algorithm, Secure Hash Algorithms.

DSC 6P: Advanced Data Structure and Algorithm Lab **1 credit [20L]**

Program Details:

List of Experiments: Experiments should include but not limited to:

1. Write a program in C to implement simple Stack and their operations using array and Linked list
2. Write a program in C to implement Queue and their operations using array and linked list.
3. Write a program in C to implement Circular Queue and Priority Queue using array.
4. Write a menu driven program that implements singly linked list for the following operations: Create, Display, Concatenate, merge, union, intersection
5. Write a menu driven program that implements singly linked list for the following operations: a) Insert before first node, after last or any intermediate position. b) delete first node, last or any intermediate node.
6. Write a program in C to insert a node in a sorted linked list.
7. Write a menu driven program that implements doubly linked list for the following operations: Create, Display, Count, Insert, Delete, Search, Copy, Reverse, Sort
8. Write a menu driven program that implements doubly linked list for the following operations: Create, Display, Concatenate, merge, union, intersection
9. Write a program in C to implement the deque operations.
10. Write a menu driven program that implements Singly circular linked list for the following operations: Create, Display, Count, Insert, Delete, Search, Copy, Reverse, Sort
11. Write a program to implement Inorder, Preorder and Post Order traversal of a binary tree both recursive and non-recursive way.
12. Write a menu driven program in C to Create a binary search tree, Traverse the tree in Inorder, Preorder and Post Order, Search the tree for a given node and delete the node
13. Write a program in C to implement insertion and deletion in B tree.
14. Write a program in C to implement insertion and deletion in AVL tree
15. Write a menu driven program that implements Heap tree (Maximum and Minimum Heap tree) for the following operations. (Using array) Insert, Delete.
16. Write a program in C to covert an infix to postfix expression.
17. Write a program in C to represent polynomial and add and subtract two polynomials.
18. Write a program to implement double hashing technique to map given key to the address space.
19. Write a program to implement for the collision resolution (linear probing).
20. Write a program in C to represent KMP pattern matching algorithm.
21. Write a program in C for sorting methods.
22. Write a program in C to sort a list of numbers using quick sort or Mergesort.
23. Write a program in C to find the minimum number of multiplications needed to multiply a chain of matrices using dynamic programming approach.
24. Write a program in C to find the longest common subsequence in a given string using dynamic programming approach.
25. Write a program in C to solve a given 0/1 knapsack problem using greedy approach.
26. Write a program in C to find the shortest distances between every pair of vertices in a given edge weighted directed graph using dynamic programming.
27. Write a program in C to solve the N Queen problem using backtracking.
28. Write a program in C to implement Dijkstra's shortest path algorithm for a given directed graph.
29. Write a program in C to insert and delete nodes in graph using adjacency matrix.
30. Write a program in C to implement Breadth First search using linked representation of graph.
31. Write a program in C to implement Depth first search using linked representation of graph.
32. Write a program in C to create a minimum spanning tree using Kruskal's algorithm.
33. Write a program in C to create a minimum spanning tree using Prim's algorithm, etc.

Reference Books:

1. "Introduction to Algorithms" – Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein (CLRS) – *MIT Press*
2. "Algorithm Design" – Jon Kleinberg, Éva Tardos – *Pearson*
3. "Data Structures and Algorithm Analysis in C++" – Mark Allen Weiss – *Pearson*
4. "The Algorithm Design Manual" – Steven S. Skiena – *Springer*
5. "Algorithms" – Robert Sedgewick, Kevin Wayne – *Addison-Wesley*
6. Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed – *Fundamentals of Data Structures in C* – University Press
7. Mark Allen Weiss – *Data Structures and Algorithm Analysis in C* – Pearson Education
8. Robert Lafore – *Data Structures and Algorithms in Java/C++* – Sams Publishing
9. Sanjay Mohapatra, Pritimoy Paul – *Data Structures Through C in Depth* – BPB Publications
10. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein – *Introduction to Algorithms (CLRS)* – MIT Press
11. Steven S. Skiena – *The Algorithm Design Manual* – Springer
12. Robert Sedgewick, Kevin Wayne – *Algorithms* – Addison-Wesley
13. Narasimha Karumanchi – *Data Structures and Algorithms Made Easy* – CareerMonk Publications

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	CO1	3	3	2	3	2	1	2	1	1	1	1	1
	CO2	3	3	3	2	3	1	2	1	1	1	1	1
	CO3	3	3	3	3	3	1	2	1	1	1	1	1
	CO4	3	3	3	2	2	1	3	1	2	1	1	1

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSC 7T: Advanced Database Management System **3 credits**

Course Outcome:

Students able to:

CO1: Comprehend and analyze different database models, user types, and system architectures.

CO2: Apply advanced relational database concepts, including dependencies,

normalization, and chase algorithms.
CO3: Understand and implement parallel, distributed, active, and deductive database systems, including transaction processing protocols.
CO4: Model complex data using object-oriented and object-relational databases, and analyze real-world case studies for practical applications.

Details Syllabus:

Unit 1: Introduction to Advanced Database Concepts [3L]

Concept and overview of DBMS, database models, database languages, database users, DBA functions, and the three-schema architecture.

Unit 2: Relational Databases and Integrity Constraints [5L]

Functional, multivalued, and join dependencies; template algebraic, inclusion and generalized functional dependencies; chase algorithms; synthesis of relational schemes.

Unit 3: Query Processing and Optimization [5L]

Evaluation of relational operations, transformation of expressions, indexing, optimization strategies, null values, and partial information.

Unit 4: Parallel and Distributed Databases [5L]

Fragmentation and replication, location and fragment transparency, distributed query processing and optimization, distributed transactions and concurrency control, deadlock handling, commit protocols, and parallel query evaluation.

Unit 5: Advanced Transaction Processing [5L]

Nested and multilevel transactions, compensating transactions, sagas, long-duration transactions, weak consistency levels, transaction workflows, processing monitors.

Unit 6: Active Databases [3L]

Triggers in SQL, ECA rules (Event-Condition-Action), concurrency control, compensation, recovery.

Unit 7: Deductive Databases [4L]

Datalog and recursion, evaluation strategies, recursive queries with negation.

Unit 8: Object-Oriented and Object-Relational Databases [5L]

Modeling complex semantics: specialization, generalization, aggregation, object identity and references, architecture of object-oriented and object-relational databases.
Case studies: Gemstone, O2, Object Store, SQL3, Oracle XXI, DB2.

DSC 7P: Advanced Database Management System Lab 1 credit [20L]

Program Details:

Unit 1: Structured Query Language (SQL):

Creating a database, Creating tables and specifying data types, Specifying constraints (primary key, foreign key, check, not null), Creating indexes

Unit 2: Table and Record Handling:

Using INSERT, SELECT, DELETE, UPDATE, TRUNCATE, DROP, ALTER statements

Unit 3: Retrieving Data from Database:

SELECT with WHERE, logical operators, Using IN, BETWEEN, LIKE, ORDER BY, GROUP BY, HAVING, Aggregate functions (SUM, COUNT, AVG, MIN, MAX), Table joins: INNER JOIN, OUTER JOIN, SELF JOIN, Subqueries and nested queries

Unit 4: Database Management:

Creating and using views, Using column aliases, Creating users, Granting and revoking privileges

Unit 5: Advanced PL/SQL (Oracle):

Using cursors in PL/SQL, Writing stored procedures in PL/SQL

Reference Books:

1. Abraham Silberschatz, Henry Korth, and S. Sudarshan – Database System Concepts, McGraw-Hill.
2. Raghu Ramakrishnan – Database Management Systems, WCB/McGraw-Hill.
3. Bipin Desai – An Introduction to Database Systems, Galgotia.
4. J. D. Ullman – Principles of Database Systems, Galgotia.
5. R. Elmasri and S. Navathe – Fundamentals of Database Systems, Addison-Wesley.
6. Serge Abiteboul, Richard Hull, and Victor Vianu – Foundations of Databases, Addison-Wesley.

7. Abraham Silberschatz, Henry Korth, and S. Sudarshan – Database System Concepts, McGraw-Hill.
8. Raghu Ramakrishnan – Database Management Systems, WCB/McGraw-Hill.
9. Bipin Desai – An Introduction to Database Systems, Galgotia.
10. J. D. Ullman – Principles of Database Systems, Galgotia.
11. R. Elmasri and S. Navathe – Fundamentals of Database Systems, Addison-Wesley.
12. Serge Abiteboul, Richard Hull, and Victor Vianu – Foundations of Databases, Addison-Wesley.

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	CO1	3	3	2	2	2	-	-	-	-	-	-	-
	CO2	3	3	2	2	2	-	-	-	-	-	-	-
	CO3	3	3	3	3	3	-	2	-	-	-	-	-
	CO4	3	2	3	2	2	-	2	-	2	-	-	-

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSC 8T: Deep Learning **3 credits**

Course Outcome:

Students able to:

CO1: Understand the fundamental principles behind deep learning models, including MLPs, CNNs, RNNs, and attention-based mechanisms, and analyze state-of-the-art CNN architectures for tasks such as classification, detection, and segmentation.
CO2: Develop hybrid models combining CNNs and RNNs for spatio-temporal applications like video understanding and activity recognition, and apply attention models and vision transformers in vision-language tasks such as image captioning and visual question answering.
CO3: Understand and implement deep generative models such as GANs and VAEs for creative applications, including image synthesis, super-resolution, and inpainting, while exploring recent trends in few-shot, one-shot, and zero-shot learning, as well as modern paradigms like self-supervised and reinforcement learning.
CO4: Design, build, and optimize deep learning pipelines for advanced research and industrial applications in Artificial Intelligence.

Detailed Syllabus:

Unit 1: Introduction **[3L]**

Basic concept of Deep Learning, Multi-layer Perceptrons, Backpropagation

Unit 2: Convolutional Neural Networks (CNNs) **[7L]**

Basic concept of CNNs; CNN Architectures evolution: AlexNet, ZFNet, VGG, InceptionNets, ResNets, DenseNets; Visualization of Kernels; Backprop-to-image/Deconvolution Methods; Deep Dream, Hallucination, Neural Style Transfer; CAM, Grad-CAM, Grad-CAM++; Recent Methods (IG, Segment-IG, SmoothGrad)

Unit 3: CNNs in different applications **[7L]**

CNNs for Recognition and Verification (Siamese Networks, Triplet Loss, Contrastive Loss, Ranking Loss); CNNs for Detection: Basic concept of Object Detection, R-CNN, Fast R-CNN, Faster R-CNN, YOLO, SSD, RetinaNet; CNNs for Segmentation: FCN, SegNet, U-Net, Mask-RCNN

Unit 4: Recurrent Neural Networks (RNNs) **[5L]**

Basic concept of RNNs; CNN + RNN Models for Video Understanding: Spatio-temporal Models, Action/Activity Recognition

Unit 5: Attention Models **[5L]**

Basic concept of Attention Models; Vision and Language: Image Captioning, Visual QA, Visual Dialog; Spatial Transformers; Transformer Networks; Vision Transformer

Unit 6: Deep Generative Models and its applications **[8L]**

Basic concept of Deep Generative Models: GANs, VAEs; Other Generative Models: PixelRNNs, NADE, Normalizing Flows, etc; Image Editing, Inpainting, Superresolution, 3D Object Generation, Security; Variants: CycleGANs, Progressive GANs, StackGANs, Pix2Pix, etc.

DSE 5.2P: Deep Learning Lab **1 credit [20L]**

Program Details:

Unit 1: Environment Setup & Basics

Setting up Python, PyTorch/TensorFlow, Jupyter Notebooks, Using GPU acceleration (Colab / local CUDA), Building a basic MLP from scratch and with PyTorch

Unit 2: CNN Basics & Architectures

Implementing a basic CNN for MNIST/CIFAR-10 classification, Training AlexNet and VGG from scratch or using pretrained weights, Visualizing convolutional kernels and feature maps

Unit 3: Advanced CNNs & Visualization

Implementing ResNet, DenseNet using transfer learning, Grad-CAM and Grad-CAM++ for visual explanations, Neural Style Transfer and Deep Dream implementation

Unit 4: CNNs in Applications

Implementing object detection with YOLO/SSD/Faster R-CNN, Semantic segmentation with U-Net/FCN, Image similarity search with Siamese Networks & Triplet Loss

Unit 5: RNNs & Spatio-Temporal Models

Implementing a basic RNN and LSTM for sequence modeling, Combining CNN + RNN for video action recognition, Experiment with 3D CNNs for video tasks

Unit 6: Attention Models & Vision Transformers

Implementing an image captioning model (CNN encoder + RNN/Transformer decoder), Visual Question Answering (VQA) basics with pre-trained transformers, Working with Vision Transformer (ViT) for classification

Unit 7: Deep Generative Models

Implementing a simple GAN for image generation, Implementing Variational Autoencoder (VAE) for image reconstruction, Style Transfer and Super-Resolution with GAN-based models (CycleGAN, Pix2Pix)

References Books:

1. Deep Learning From Scratch: Building with Python from First Principles by Seth Weidman published by O`Reilley
2. Deep Learning with Python by Manning Publications
3. Deep Learning with PyTorch by Manning Publications
4. Grokking Deep Learning by Andrew W. Trask published by Manning Publications
5. Deep Learning by Ian Goodfellow, Yoshua Bengio and Aaron Courville published by MIT Press

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12

DSC 1 (T+P)	**CO1**	**3**	**3**	**2**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
	CO2	**3**	**3**	**3**	**2**	**3**	**-**	**2**	**-**	**-**	**-**	**-**	**-**
	CO3	**3**	**3**	**3**	**3**	**3**	**-**	**2**	**-**	**-**	**-**	**-**	**2**
	CO4	**3**	**3**	**3**	**3**	**3**	**-**	**3**	**2**	**2**	**-**	**2**	**3**

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSC 9T: Advanced Network **3 credits**

Course Outcome (CO):

Students able to:

CO1: Design IP addressing schemes with subnetting and implement routing algorithms (link-state, distance-vector, hierarchical) for efficient network topologies and configure Internet routing protocols (RIP, OSPF, BGP) and optimize broadcast/multicast routing strategies using ICMP for network diagnostics.

CO2: Implement packet classification techniques to enable Quality of Service (QoS) mechanisms, including traffic shaping, policing, and Differentiated Services.

CO3: Architect SDN solutions using OpenFlow and SDN controllers (ONOS/OpenDaylight) and deploy NFV service chains for use cases like SD-WAN or network slicing.

CO4: Design secure VPN infrastructures and mobile networks (Mobile IP, Mobile TCP) for ad-hoc, sensor, and PAN environments.

Detailed Syllabus:

Unit 1: Logical Addressing and Routing **[10L]**

IP Addressing, Subnetting, Routing Algorithm (Link State, Distance Vector, Hierarchical), Routing in the Internet (RIP, OSPF, BGP), Broadcast and Multicast Routing Algorithms, ICMP

Unit 2: Packet Classification and Services **[10L]**

Need for packet classification and methods for packet classification. Differentiated Service, Quality of Service, Traffic Polishing, Traffic Shaping

Unit 3: Software-Defined Networking (SDN) and Network Function Virtualization (NFV) **[8L]**

SDN fundamentals: control plane and data plane separation, OpenFlow protocol, SDN controllers: ONOS, OpenDaylight, Ryu, NFV concepts: virtualized network functions, service chaining, Orchestration frameworks: Kubernetes for network management

Use cases: SD-WAN, network slicing

Unit 4: VPN and Mobile Networks **[7L]**

Overview of VPN networks. Mobile Networks: LAN, PAN, Sensor Networks, Ad-hoc Networks, Mobile IP, Mobile TCP

DSC 9P: Advanced Network Lab **1 credit [20L]**

Program Details:

Experiments should include, but not be limited to:

1. Implementation of Stop and Wait Protocol and Sliding Window Protocol.
2. Study of Socket Programming and Client-Server model
3. Write a code simulating ARP/RARP protocols.
4. Write a code simulating PING and TRACEROUTE commands
5. Create a socket for HTTP for web page upload and download.
6. Write a program to implement RPC (Remote Procedure Call)
7. Implementation of Subnetting.
8. Applications using TCP Sockets, like
    a. Echo client and echo server
    b. Chat
    c. File Transfer
9. Applications using TCP and UDP Sockets, like
    a. DNS
    b. SNMP
    c. File Transfer
10. Simulation of Congestion Control Algorithms.
11. Perform a case study about the different routing algorithms to select the network path with its optimum and economical during data transfer.
    a. Link State routing
    b. Flooding
    c. Distance vector

Reference Books:

1. Andrew S. Tanenbaum, David J. Wetherall -- Computer Networks, Pearson Education.
2. James F. Kurose, Keith W. Ross -- Computer Networking: A Top-Down Approach, Pearson.
3. William Stallings -- Cryptography and Network Security: Principles and Practice, Pearson.
4. Behrouz A. Forouzan -- Data Communications and Networking, McGraw-Hill Education.
5. Thomas D. Nadeau, Ken Gray -- SDN: Software Defined Networks, O'Reilly Media.

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1	CO1	3	3	3	2	2	-	-	-	-	-	-	-

(T+P)	**CO2**	**2**	**3**	**3**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
	CO3	**3**	**3**	**3**	**2**	**3**	**-**	**2**	**-**	**-**	**-**	**2**	**2**
	CO4	**2**	**2**	**3**	**2**	**2**	**2**	**-**	**-**	**-**	**2**	**-**	**-**

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Social Service / Community Engagement **2 credits [25L]**

Course Outcome:

Students able to:

CO1: Connect classroom knowledge with real-life situations through hands-on experience in community service.
CO2: Foster civic responsibility and awareness among students.
CO3: Engage students in social service activities and fieldwork, integrating academic learning with community engagement.
CO4: Develop skills, knowledge, and sensitivity for working effectively with diverse groups in society.

Details Syllabus:

1. Working as Motivators under the Swatch Bharat Campaign of the Government.
2. Literacy drive : Teaching in the Charitable School
3. Enroll as NSS Volunteers for various projects (Cleanliness, Women/child health awareness).
4. Counseling camps in villages
5. Tree plantation
6. Enroll in the Gandhian Studies Centre as student Volunteer for surveys in villages.
7. Women Empowerment Programmes.
8. Environment Awareness (Reduce Pollution)
9. Disaster Management/Relief Work.

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	**COs**	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
DSC 1 (T+P)	**CO1**	**-**	**2**	**-**	**-**	**-**	**-**	**2**	**-**	**2**	**2**	**2**	**-**
	CO2	**-**	**2**	**-**	**-**	**-**	**3**	**2**	**-**	**2**	**3**	**2**	**-**

	CO3	**-**	**2**	**-**	**-**	**-**	**2**	**2**	**-**	**2**	**3**	**3**	**-**
	CO4	**-**	**-**	**-**	**-**	**-**	**2**	**2**	**-**	**2**	**3**	**3**	**2**

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

MOOCs **4 credits**

Social Service/ Community Engagement **2 credits**

Semester - IV

DSC 10T: Generative AI **3 credits**

Course Outcome:

Students able to:

CO1: Explain the principles, models, and evaluation metrics of Generative AI.
CO2: Analyze and compare architectures such as GANs, VAEs, Transformers, and Diffusion Models.
CO3: Apply generative techniques for text, image, and multimodal content creation.
CO4: Design and implement generative models using frameworks such as PyTorch/TensorFlow, OpenAI, and HuggingFace. Evaluate performance, scalability, and ethical implications of generative models through experiments and projects.

Details Syllabus:

Unit 1: Introduction to Generative AI **[8L]**

Foundations: probability, latent variable models, likelihood vs. sampling approaches. Key differences between discriminative and generative models. Applications of Generative AI (text, images, audio, multimodal).

Unit 2: Core Models **[14L]**

Variational Autoencoders (VAEs): encoder–decoder framework, ELBO. Generative Adversarial Networks (GANs): architecture, loss functions, training challenges, variants (DCGAN, StyleGAN, CycleGAN). Diffusion Models: denoising diffusion probabilistic models, sampling processes. Transformers in Generative AI: GPT family, LLMs, attention mechanism.

Unit 3: Training & Evaluation **[8L]**

Data preprocessing for generative tasks. Stability and convergence issues in GANs and diffusion models. Evaluation metrics: FID, IS, BLEU, ROUGE, perplexity.

Unit 4: Advanced Topics & Applications **[5]**

Multimodal generative models (e.g., CLIP, DALL·E, Stable Diffusion). Fine-tuning and prompt engineering in LLMs. Domain-specific applications: art, healthcare, drug discovery, robotics. Ethical issues: bias, misinformation, copyright, deepfakes.

DSC 10P: Generative AI Lab **1 credit [20L]**

Lab Exercise:

1. **VAE Implementation** – Build a simple Variational Autoencoder for MNIST or CIFAR dataset.
2. **Basic GAN** – Train a GAN for handwritten digit/image generation.
3. **GAN Variants** – Implement DCGAN or CycleGAN for domain transfer tasks.
4. **Transformer for Text Generation** – Implement GPT-like model on a small text corpus.
5. **Diffusion Models** – Experiment with denoising diffusion models.
6. **Using Pre-trained Models** – Generate images with Stable Diffusion, text with GPT/HuggingFace models.
7. **Fine-tuning** – Fine-tune a small LLM or GAN on domain-specific dataset (e.g., medical text, sketches).
8. **Evaluation Metrics** – Compute FID, BLEU, perplexity on generated samples.
9. **Ethical Experimentation** – Explore prompt engineering for safe generation and document limitations.
10. **Mini Project** – Develop an applied generative AI solution (text-to-image, dialogue generation, creative writing, music, etc.) using public datasets.

Reference Books:

1. *Deep Learning*, Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT Press, 2016.
2. *Generative Adversarial Networks (GANs)*, Ian Goodfellow et al., MIT Press, 2020.
3. *GANs in Action: Deep Learning with Generative Adversarial Networks*, Jakub Langr and Vladimir Bok, Manning Publications, 2019.
4. *Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play*, David Foster, O'Reilly Media, 2nd Edition, 2022.
5. *Natural Language Processing with Transformers: Building Language Applications with Hugging Face*, Lewis Tunstall, Leandro von Werra, and Thomas Wolf, O'Reilly Media, 2022.
6. *Deep Learning for Natural Language Processing*, Palash Goyal, Sumit Pandey, and Karan Jain, Apress, 2018.
7. *Diffusion Models in Vision: A Comprehensive Guide*, Patrick Esser et al., Springer, 2023.

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12

DSC 1 (T+P)	**CO1**	**3**	**2**	**-**	**-**	**2**	**2**	**1**	**-**	**-**	**1**	**-**	**-**
	CO2	**3**	**3**	**2**	**2**	**3**	**-**	**1**	**-**	**-**	**-**	**-**	**-**
	CO3	**2**	**2**	**3**	**-**	**3**	**1**	**1**	**-**	**-**	**2**	**-**	**2**
	CO4	**2**	**3**	**3**	**2**	**3**	**3**	**2**	**2**	**2**	**2**	**2**	**2**

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

DSC 11T: High Performance Computing **3 credit**

Course Outcome:

Students able to:

CO1: Explain the principles, models, and performance metrics of parallel and high-performance computing systems.
CO2: Analyze parallel computer architectures, memory organizations, and interconnection networks for efficient data communication.
CO3: Apply fundamental parallel algorithmic techniques (divide-and-conquer, partitioning, pipelining, synchronization) to computational problems.
CO4: Develop, implement, and evaluate high-performance programs using tools like OpenMP, MPI, and CUDA.

Details Syllabus:

Unit 1: Introduction to High Performance Computing (HPC) **[6L]**

Basics of parallel and distributed computing, Need for HPC and applications in science, engineering, and AI, Performance metrics: speedup, efficiency, Amdahl's law, Gustafson's law.

Unit 2: Parallel Architectures and Memory Models **[8L]**

Flynn's taxonomy (SISD, SIMD, MISD, MIMD), Shared vs. distributed memory models, Interconnection networks: bus, crossbar, mesh, hypercube, fat tree, Cache coherence and consistency models.

Unit 3: Parallel Algorithms and Models **[12L]**

Principles of parallel algorithm design, Techniques: partitioning, divide-and-conquer, pipelining, synchronization, Parallel algorithms for searching, sorting, matrix multiplication, graph problems.

Unit 4: Programming Models and Frameworks **[7L]**

Shared memory programming with OpenMP, Distributed memory programming with MPI, GPU programming with CUDA fundamentals, Hybrid programming approaches (MPI + OpenMP, CUDA + MPI)

Unit 5: Applications and Case Studies **[4L]**

HPC in weather prediction, bioinformatics, AI/ML training, Benchmarking and performance evaluation tools, Introduction to cloud-based HPC and exascale computing

DSC 11P: High Performance Computing Lab **1 credit**

Lab Exercise:

1. Performance analysis of serial vs. parallel programs
2. Implementing parallel searching and sorting using OpenMP
3. Matrix multiplication using OpenMP and MPI
4. Solving graph problems (shortest path, spanning tree) using MPI
5. CUDA programming – vector addition, matrix operations
6. CUDA programming – parallel reduction, prefix sum
7. Hybrid parallelization: MPI + OpenMP integration
8. Performance evaluation and scalability testing of parallel programs

Reference Books:

1. *Parallel Programming in C with MPI and OpenMP*, Michael J. Quinn, McGraw-Hill, 2003.
2. *An Introduction to Parallel Programming*, Peter S. Pacheco, Morgan Kaufmann, 2011.
3. *Parallel and Distributed Programming Using C++*, Cameron Hughes and Tracey Hughes, Addison-Wesley, 2003.
4. *CUDA by Example: An Introduction to General-Purpose GPU Programming*, Jason Sanders and Edward Kandrot, Addison-Wesley, 2010.
5. *Programming Massively Parallel Processors: A Hands-on Approach*, David B. Kirk and Wen-mei W. Hwu, Morgan Kaufmann, 3rd Edition, 2016.
6. *High Performance Computing: Modern Systems and Practices*, Thomas Sterling, Matthew Anderson, and Maciej Brodowicz, Morgan Kaufmann, 2018.

Mapping of CO (Course outcome) and PO (Programme Outcome)

Course	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
DSC 1 (T+P)	CO1	3	2	-	-	2	-	1	-	-	-	-	-
	CO2	3	3	2	2	2	-	1	-	-	-	-	-
	CO3	2	2	3	2	2	-	1	-	-	-	-	2
	CO4	2	3	3	3	3	-	2	2	2	-	2	2

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Research Project/ Dissertation **8 credits**

Internship / Capstone Project / Applied Field or Industry Project/ Innovation & Incubation/ Entrepreneurship/ Start-up Proposal or Practice **4 credits**

Intellectual Property Right (IPR) / Skill Enhanced Course **2 credits**