VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

CURRICULUM AND SYLLABI

(2025-2026)

M. Tech. Computer Science and Engineering (Artificial Intelligence and Machine Learning)

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

VISION STATEMENT OF VELLORE INSTITUTE OF TECHNOLOGY

Transforming life through excellence in education and research.

MISSION STATEMENT OF VELLORE INSTITUTE OF TECHNOLOGY

- **World class Education**: Excellence in education, grounded in ethics and critical thinking, for improvement of life.
- **Cutting edge Research**: An innovation ecosystem to extend knowledge and solve critical problems.
- **Impactful People**: Happy, accountable, caring and effective workforce and students.
- **Rewarding Co-creations**: Active collaboration with national & international industries & universities for productivity and economic development.
- **Service to Society**: Service to the region and world through knowledge and compassion.

VISION STATEMENT OF THE SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

- To be a world-renowned centre of education, research and service in computing and allied domains.

MISSION STATEMENT OF THE SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

- To offer computing education programs with the goal that the students become technically competent and develop lifelong learning skill.
- To undertake path-breaking research that creates new computing technologies and solutions for industry and society at large.
- To foster vibrant outreach programs for industry, research organizations, academia and society.

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

M. Tech. Computer Science and Engineering (Artificial Intelligence and Machine Learning)

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO 1: Graduates will acquire a core competency in technology development and practical skills in Artificial Intelligence and Machine Learning enabling them to contribute to advancements in designing and deploying smart systems.

PEO 2: Graduates will pursue career as successful AI engineers, machine learning engineers, researchers or entrepreneurs by providing sustainable solutions across various industry domains.

PEO 3: Graduates will develop the ability to communicate, coordinate and collaborate effectively within teams and cultivate ethical behaviour to succeed in their professional career.

PEO 4: Graduates will adopt a holistic approach to address the challenges in evolving technology and AI trends.

VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

M. Tech. Computer Science and Engineering (Artificial Intelligence and Machine Learning)

PROGRAMME OUTCOMES (POs)

PO1: Independently carry out research/investigation and development work to solve practical problems

PO2: Write and present a substantial technical report/document

PO3: Demonstrate a degree of mastery in designing and developing intelligent systems applying all the relevant standards with realistic constraints

PO4: Analyse the complex engineering problems and design sustainable solutions

PO5: Communicate effectively at the work place and deploy engineering and AI tools with ethical considerations.

PO6: Recognize the need for independent and life-long learning in the broadest context of technological change and evolving AI landscape.

VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

CURRICULUM

M. Tech. Computer Science and Engineering (Artificial Intelligence and Machine Learning) - (2025)

Category Credit Detail			
Sl.No.	**Description**	**Credits**	**Maximum Credit**
1	UCC - University Core Courses	39	39
2	OEC - Open Elective Courses	3	3
3	PFCC - Professional Core Courses	24	24
4	PFEC - Professional Elective Courses	14	14
Total Credits		80	

University Core Courses

sl. no	Course Code	Course Title	Course Type	Version	L	T	P	J	Credits
1	MACSE698	Internship I/ Dissertation I	Project	1.0	0	0	0	0	10.0
2	MACSE699	Internship II/ Dissertation II	Project	1.0	0	0	0	0	10.0
3	MAENG501	Technical Report Writing	Embedded Theory and Lab	1.0	1	0	4	0	3.0
4	MASET697	Project Work	Project	1.0	0	0	0	0	10.0
5	MASTS601	Competitive Coding I	Soft Skill	1.0	3	0	0	0	3.0
6	MASTS602	Competitive Coding II	Soft Skill	1.0	3	0	0	0	3.0

Open Elective Courses

sl.no	Course Code	Course Title	Course Type	Version	L	T	P	J	Credits
1	MASTS501	Qualitative and Quantitative Skills Practice I	Soft Skill	1.0	3	0	0	0	3.0

Professional Core Courses

sl.no	Course Code	Course Title	Course Type	Version	L	T	P	J	Credits
1	MACSE501	Data Structures and Algorithm Analysis	Embedded Theory and Lab	1.0	3	0	2	0	4.0
2	MACSE512	Operating Systems	Embedded Theory and Lab	1.0	3	0	2	0	4.0
3	MACSE513	Computer Networks	Embedded Theory and Lab	1.0	3	0	2	0	4.0
4	MACSE517	Computer Architecture and Organization	Embedded Theory and Lab	1.0	3	0	2	0	4.0

5	MACSE518	Database Modelling and Design	Embedded Theory and Lab	1.0	3	0	2	0	4.0
6	MACSE519	Machine Learning and Applications	Embedded Theory and Lab	1.0	3	0	2	0	4.0

M. Tech. Computer Science and Engineering (Artificial

Professional Elective Courses									
sl.no	**Course Code**	**Course Title**	**Course Type**	**Version**	**L**	**T**	**P**	**J**	**Credits**
1	MACSE622	Data Engineering and Visualization	Embedded Theory and Lab	1.0	3	0	2	0	4.0
2	MACSE623	Text and Speech Analytics	Theory Only	1.0	3	0	0	0	3.0
3	MACSE629	Deep Learning and Reinforcement Learning	Embedded Theory and Lab	1.0	3	0	2	0	4.0
4	MACSE630	Generative AI and Large Language Models	Embedded Theory and Lab	1.0	3	0	2	0	4.0
5	MACSE631	MLOps	Embedded Theory and Lab	1.0	3	0	2	0	4.0
6	MACSE632	Multi-Task and Meta Learning	Theory Only	1.0	3	0	0	0	3.0
7	MACSE633	Graph Database and Analytics	Theory Only	1.0	3	0	0	0	3.0
8	MACSE634	Responsible Artificial Intelligence	Theory Only	1.0	3	0	0	0	3.0

Course Code	Course Title	L	T	P	C
MACSE698	Internship I/ Dissertation I	0	0	20	10
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

To provide sufficient hands-on learning experience in the design, development, and analysis of appropriate products or processes.
To enhance technical skill sets in the student's chosen field of study.
To foster a research-oriented mindset through practical engagement.

Course Outcomes

Gain in-depth expertise in the major subject or field of study, including comprehensive understanding and critical insight into ongoing research and development trends.
Develop the ability to approach complex problems holistically, enabling critical, independent, and creative thinking in identifying, formulating, and addressing multifaceted challenges.
Cultivate a strong awareness of the ethical dimensions associated with research and development activities, promoting integrity and responsible conduct in scholarly and professional work.
Enhance research communication and dissemination skills, including the ability to effectively present findings through technical writing, oral presentations, and scholarly discussions.
Achieve recognition through publications in peer-reviewed journals and international conferences, which serve as valuable indicators of research quality and impact.

General Guidelines

(PROJECT DURATION - One Semester): - The dissertation may encompass a wide range of scholarly activities, including theoretical analysis, modeling and simulation, experimental investigation, prototype design, equipment fabrication, data correlation and analysis, software development, or applied research. It is expected to be an individual effort, demonstrating the student's independent capability in their area of specialization. The work may be conducted either within the university or externally, such as in a relevant industry or research institution. While not mandatory, publications resulting from the dissertation in peer-reviewed journals or international conferences will be considered a valuable addition, reflecting the quality and impact of the research undertaken.

Mode of Evaluation : Project Reviews, Project Reports, VivaVoce/Student Interactions

Recommended by Board of Studies :	02-06-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE699	Internship II/ Dissertation II	0	0	20	10
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

To provide hands-on learning experience in the design of appropriate products or processes.
To develop practical skills related to the development and analysis of engineering solutions.
To enhance technical competencies in the student's chosen field through experiential learning.

Course Outcomes

Formulate well-defined problem statements for complex, real-world scenarios by applying appropriate assumptions and constraints.
Conduct comprehensive literature and patent searches to establish a strong foundation and context for the identified problem.
Design and implement experimental or analytical approaches, including iterative solution development, and systematically document findings.
Perform critical evaluations such as error analysis, bench marking, and cost estimation to assess the feasibility and efficiency of proposed solutions.
Synthesize results to derive meaningful scientific conclusions or practical solutions, and effectively communicate the outcomes through technical reports and presentations.

General Guidelines

(PROJECT DURATION - One Semester): - The dissertation may encompass a wide range of scholarly activities, including theoretical analysis, modeling and simulation, experimental investigation, prototype design, equipment fabrication, data correlation and analysis, software development, or applied research. It is expected to be an individual effort, demonstrating the student's independent capability in their area of specialization. The work may be conducted either within the university or externally, such as in a relevant industry or research institution. While not mandatory, publications resulting from the dissertation in peer-reviewed journals or international conferences will be considered a valuable addition, reflecting the quality and impact of the research undertaken.

Mode of Evaluation : Project Reviews, Project Reports, VivaVoce/Student Interactions

Recommended by Board of Studies :	02-06-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MAENG501	Technical Report Writing	1	0	4	3
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

To develop communicative competence in students.
To apply advanced technical communication principles in practical writing tasks for diverse professional contexts.
To execute research and analytical skills to produce coherent and evidence-based technical documents through practical exercises.

Course Outcomes

Apply the principles of effective report writing.
Demonstrate the ability to draft and present technical reports.
Produce key components of technical documents, including sections focusing on purpose, audience, and structure, through writing assignments.
Create technical summaries and abstracts by practicing techniques for summarizing and paraphrasing complex information.

Module:1	**Fundamentals of Technical Writing**	**3 hours**
Introduction to Technical Writing: Definition and typical forms (reports, instructions, proposals); Key Factors in Technical Writing: Purpose, Audience, and Tone; General Writing Basics: Clarity, fluency, effectiveness; The Process of Writing: Pre-writing, Writing, and Post-writing stages; Organization in Writing: Having an outline, using introductions, headings, lists, figures, and summaries.		
Module:2	**Technical Grammar**	**2 hours**
Concord; Tense Shifts		
Module:3	**Introduction to Reports and Report Structure Front Matter**	**2 hours**
Reports in Organizations: Role and importance of reporting in corporate and industrial segments.; Purpose of Reports: Conveying decisions, facts, and information accurately and up-to-date.; Report Structure Overview: Division into Front Matter, Main Body, and Back Matter; Components of the Front Matter		
Module:4	**Report Structure Main Body and Back Matter**	**4 hours**

Components of the Main Body: Introduction, Discussion or Description, Conclusions, and Recommendations; Writing Introduction, Discussion or Description, Conclusions and Recommendations, Methods of Reporting: The Letter Method and The Schematic Method; Routine Reports: Nature, frequency, and function in organizations (often statistical, fixed intervals).		
Module:5	**Technical Proposals**	**2 hours**
Punctuation Right words and phrases; avoiding cliches, jargons, foreign words and phrases, ambiguity, redundancy, circumlocution - Developing hints		
Module:6	**Contemporary Issues**	**2 hours**
Guest Lecture		
	Total Lecture Hours:	**15 hours**

Text Book(s)

Kumar. S & Pushplata. , "**Effective Communication Skills.** ", New Delhi OUP, **2018**
Muralikrishna and Sunita Mishra, "**Communication Skills For Engineers**", Pearson, 2nd Edition, **2011**
Shirley Mathew., "**Effective Communication Skills.**", Nirali Prakashan, **2025**

Reference Books

Indicative Experiments

1. **Introduction to the Technical Writing** Introduction to Technical Report Writing**;** Analyzing and identifying the characteristics of effective technical writing in sample documents.; Short writing exercises focusing on clarity, conciseness, and identifying purpose, audience, and tone in simple technical scenarios. **Activity:** Reviewing and providing feedback on short technical descriptions.	2 hours
2. **Analyzing Audience and Context for Practical Writing** Applying reader and stakeholder analysis techniques to specific engineering report scenarios; Drafting content segments tailored for different technical and non-technical audiences (e.g., writing an executive summary for managers vs. a technical description for fellow engineers); Simulating audience needs assessment based on given project descriptions. **Activity:** Developing a 'reader profile' for a major report project.	4 hours
3. **Technical Grammar and Style in Practice**	2 hours

Intensive practical exercises on complex grammatical structures, sentence syntax, and common errors in technical writing; Exercises in applying principles of clarity, conciseness, and precision to improve technical sentences and paragraphs; Practicing the appropriate use of active and passive voice in different report sections. **Activity:** Peer-editing session focusing on grammar, mechanics of writing, and technical style.	
4. **The Writing Process and Report Outlining Lab** Practicing prewriting techniques for complex technical topics: brainstorming, mind mapping, and systematic outlining; Developing detailed hierarchical outlines for a major technical report project, including main and sub-points; Planning content organization based on report type (e.g., feasibility, empirical research, lab report) and audience needs. **Activity:** Group exercise to compare and refine report outlines.	4 hours
5. **Writing the Report Front Matter and Introduction** Detailed practical guidance on drafting all components of the report Front Matter; Creating a professional Title Page, Forwarding Letter/Preface, Acknowledgements, and Table of Contents for the ongoing report project; Drafting effective Abstracts and Summaries (Executive Summaries) based on provided technical content, focusing on capturing the essence. **Activity:** Writing the Introduction section of the report project, including background, scope, and objectives.	4 hours
6. **Writing the Report Body: Data, Descriptions, and Discussion** Techniques for presenting data, technical descriptions, and analysis in the main body; Drafting sections of the report body focusing on presenting organized data and technical details; Writing the "Discussion" section, focusing on interpreting results and explaining findings based on provided or self-generated data. **Activity:** Integrating data and analysis points into the draft of the report body.	4 hours
7. **Writing the Report Body: Conclusions and Recommendations** Practical methods for drawing logical conclusions and formulating actionable recommendations; Drafting the "Conclusions" section based on the data and	4 hours

discussion from Module 6; Developing clear and practical recommendations based on the conclusions, considering the report's purpose and audience. **Activity:** Writing the "Recommendations" section of the report project.	
8. **Transcribing Visuals** Using charts, graphs and tables; Transcribing visuals that are clear, accurate, and effectively support the report's text; Integrating created visuals into the report draft, ensuring proper placement, captions, and referencing within the text. **Activity:** Peer review focusing on the effectiveness and integration of visuals.	2 hours
9. **Report Back Matter, Condensation, and Final Review** Practical session on creating the Back Matter: Appendices, Bibliography, Glossary, and Index (if applicable). Emphasis on consistent citation and referencing styles; Compiling Appendices and formatting a Bibliography for the report project; Practicing techniques for précis writing and summarization to condense longer texts. **Activity:** Final review of the complete report draft, focusing on overall structure, flow, formatting, and coherence.	2 hours
10. **Presentation Skills** Short Presentations, Formal Presentation with PPT Analytical Presentation of Charts, Graphs and Tables **Activity**: Presentations – Individual and Group	2 hours
Total Laboratory Hours:	**60 hours**
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Presentaion	
Recommended by Board of Studies :	16-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MASTS601	Competitive Coding I	3	0	0	3
Pre-requisite	NIL	Syllabus version			
		1.0			

Course Objectives

Equip learners with an in-depth understanding of linear data structures (linked lists, stacks, queues, deques, etc.). Enabling them to design, implement, and optimize these structures for various problem-solving scenarios. Emphasis is placed on both theory (complexities, operations) and hands-on coding to master data structure manipulation and usage.

Course Outcomes

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

● Confidently implement and utilize linked lists, stacks, queues, and deques

● Analyse and optimize the time/space complexities of linear data structure operations (insertion, deletion, traversal)

● Apply problem-solving strategies to real-world tasks and competitive coding challenges, leveraging linear data structures effectively

● Integrate debugging, testing, and best coding practices when working with linear data structures

Module:1	Linked List-1	6 hours

Introduction to Linked Lists: Definition, structure, advantages; Types of Linked Lists: Singly, Doubly, Circular.

Basic Operations: Creating a linked list; Insertion (at beginning, middle, end); Deletion (from beginning, middle, end); Traversal (iterative, recursive).

Module:2	Linked List-2	6 hours

Advanced Topics: Reversing a linked list (iterative, recursive); Detecting and removing loops; Merging and splitting linked lists; Intersection of two linked lists.

Module:3	Stack-1	5 hours

Introduction to Stack: Definition, LIFO property, real-world applications (e.g., undo functionality, browser history); Static vs dynamic stacks.

Operations on Stack: Push, pop, peek; Checking stack overflow and underflow.

Implementation: Using arrays; Using linked lists.

Module:4	Stack-2	4 hours

Applications of Stack: Balancing parentheses in expressions; Converting infix expressions to postfix; Evaluating postfix expressions; Solving backtracking problems (e.g., maze solving).

Module:5	Queue	7 hours

Introduction to Queue: Definition, FIFO property; Types of Queues: Simple, Circular, Double-Ended.

Operations on Queue: Enqueue (insertion), Dequeue (removal), Peek (front element); Checking queue overflow and underflow.

Implementation: Using arrays; Using linked lists.

Applications of Queue: Task scheduling (e.g., CPU scheduling, print queue); Breadth-First Search (BFS) implementation; Managing resources in real-time systems (e.g., job queues).

Module:6	Priority Queue	5 hours

Introduction to Priority Queue: Definition, importance in task prioritization; Differences between Priority Queue and Simple Queue.

Operations on Priority Queue: Enqueue and Dequeue operations based on priority.

Implementation: Using arrays; Using heaps (Binary Heaps: Min-Heap and Max-Heap).

Module:7	Hash map	9 hours
Introduction to Hash map: Definition, key-value pair structure, advantages over arrays. Hash Functions and Collision Handling: Direct addressing, simple hash functions, modulo operation; Collision resolution techniques: Chaining, Linear Probing, Quadratic Probing. Applications of Hash map: Frequency counting (e.g., counting occurrences of elements); Caching mechanisms (e.g., implementing LRU cache); Finding duplicates in arrays or strings; Solving anagram problems; Building indexes for fast lookup.		
Module:8	**Interview Prep - Networking**	**3 hours**
Networking Fundamentals: OSI model TCP/IP basics Network Protocols: HTTP/HTTPS DNS and DHCP FTP and SMTP - Practical Interview Questions: - Debugging network issues, Firewall, DNS, Ping and Trace route - Explaining client-server architecture - Designing scalable systems		
Total Lecture hours:		**45 hours**

Text Book	
1.	A Textbook of Data Structures and Algorithms 1: Mastering Linear Data Structures – 1st Edition by G A Vijayalakshmi Pai – Wiley Publication
Reference Books	
1.	Data Structure and Algorithms Made Easy by Narasimha Karumanchi -5th edition

Mode of Evaluation: Written assignment, Quiz, Project & FAT.			
Recommended by Board of Studies			
Approved by Academic Council		Date	

Course Code	Course Title	L	T	P	C
MASTS501	Qualitative and Quantitative Skills Practice I	3	0	0	3
Pre-requisite	NIL	Syllabus version			

Course Objectives:

1. To enhance the logical reasoning skills of students and improve problem-solving abilities
2. To strengthen the ability of solving quantitative aptitude problems
3. To enrich the verbal ability of the students for academic purposes

Course Outcomes:

1. Become experts in solving problems of quantitative Aptitude
2. Learn to defend and critique concepts of logical reasoning
3. Integrate and display verbal ability effectively

Module:1	Speed Maths	12 hours

Speed Maths

- Addition and Subtraction of bigger numbers
- Square and square roots
- Cubes and cube roots
- Vedic Maths
- Multiplication Shortcuts
- Multiplication of 3 and higher digit numbers
- Simplifications
- Comparing fractions
- Shortcuts to find HCF and LCM
- Divisibility tests shortcuts

Module:2	Percentages, Simple and compound Interest	5 hours

- Percentages
- Simple Interest
- Compound Interest
- Realtion between Simple and Compound Interest

Module:3	Profit and loss, Partnership	5 hours

- Profit and loss
- Partnership
- Averages
- Mixtures and Alligations

Module:4	Data Arrangements and Blood relations	5 hours

- Linear Arrangement
- Circular Arrangement
- Multi-dimensional Arrangement
- Blood relations

Module: 5	**Cryptarithmetic and Attention to detail**	**5 hours**
• Crypt Arithmetic • Attention to detail		
Module: 6	**Life skills**	**8 hours**
• Profile building • Decision making • Critical thinking • Problem solving		
Module:7	**Reading Comprehension for placements**	**3 hours**
• Types of questions • Comprehension Strategies • Practice exercises		
Module:8	**Critical Reasoning and Voices**	**2 hours**
• Critical Reasoning • Active voice • Passive voice		
	Total Lecture hours:	**45 hours**

Text Book(s)

1.	Aggarwal R.S. (2017). *Quantitative Aptitude for Competitive Examinations* 3rd (Ed.). New Delhi: S. Chand Publishing.
2.	FACE. (2016). *Aptipedia Aptitude Encyclopedia* pt (Ed.). New Delhi: Wiley Publications.
3.	ETHNUS. (2016). *Aptimithra,* pt (Ed.) Bangalore: McGraw-Hill Education Pvt.Ltd.

Reference Books

1.1 Sharma Arun. (2016). *Quantitative Aptitude,* 7th(Ed.). Naida: McGraw Hill Education Pvt. Ltd.

Mode of evaluation: CAT, Assessments and FAT (Computer Based Test)

Recommended by Board of Studies |		
Approved by Academic Council |	Date	

Course Code	Course Title	L	T	P	C
MACSE501	Data Structures and Algorithm Analysis	3	0	2	4
Pre-requisite	Nil	Syllabus Version			
		1			

Course Objectives

1. To familiarize the concepts of data structures and provide a mathematical framework for the design and analysis of algorithms.
2. To provide a deeper insight into linear and non-linear data structures.
3. To disseminate knowledge on how to develop algorithms using various design strategies to solve real-world problems.

Course Outcomes

1. Develop the ability to analyse the efficiency of algorithms using asymptotic notations and implement linear data structures.
2. Apply tree and graph-based data structures to design and implement efficient traversal, search, and pathfinding algorithms.
3. Demonstrate the ability to design and implement efficient solutions to complex problems using divide-and-conquer and greedy strategies.
4. Formulate optimal and exhaustive search-based solutions for computationally intensive problems using dynamic programming, backtracking, and branch-and-bound technique.
5. Critically analyse computational complexity, differentiate between P, NP, and NP-complete problems, and apply approximation algorithms to develop near-optimal solutions for intractable problems.

Module:1	Complexity Analysis and Elementary Data Structures	8 hours

Overview and importance of algorithms and data structures- Algorithm specification, Asymptotic Notation - The Big-O, Omega and Theta notation, Recursion, Performance analysis, Array, Stack, Queue, Linked-list and its types, various representations, operations and applications of Linear Data Structures.

Module:2	Non Linear Data Structures	10 hours

Trees: Binary trees, Properties of binary trees, Binary Search Tree, AVL, Red-Black Trees, B- Trees, Segment Trees - Heaps: Binary heaps, Min-Max heap – Graphs: Representation, Breadth first Search, Depth First Search, Shortest path algorithm –Bellman Ford algorithm - Network flow algorithm: Ford-Fulkerson and Push-relabel algorithm.

Module:3	Divide and Conquer and Greedy Techniques	8 hours

Divide and Conquer: Merge Sort, Quick Sort, Karatsuba's fast multiplication method, Strassen's Matrix Multiplication - Greedy Strategy: Job Sequencing Problem with Deadlines, Minimum Spanning Tree: Prims algorithm, Kruskal's algorithm, Huffman coding.		
Module:4	**Dynamic Programming and Backtracking Branch and Bound Techniques**	**9 hours**
Dynamic programming: Matrix Chain Multiplication, Longest Common Subsequence - Backtracking: N-Queens problem, Subset Sum problem, Graph coloring problem Branch and Bound: Subset sum, A-Star, LIFO-BB and FIFO BB methods.		
Module:5	**Complexity Classes and Approximation Algorithms**	**8 hours**
Class P - Class NP - Reducibility and NP-completeness - Circuit Satisfiability problem- 3CNF, Independent Set, Clique, Approximation Algorithms: Vertex Cover, Set Cover and Travelling salesman.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. , "**Introduction to Algorithms** ", MIT Press, **2022**
Allen, Weiss Mark. , "**Data structures and algorithm analysis in C++**", Pearson Education India, **2007**

Reference Books

Skiena, Steven S. , "**The Algorithm Design Manual (Texts in Computer Science)**", Springer, 3rd Edition, **2020**
Jon Kleinberg and EvaTardos, "**Algorithm Design**", Pearson Education, 1st Edition, **2014**
Brass, Peter., "**Advanced data structures**", Cambridge University Press, **2008**

Indicative Experiments

1. Linear data structures: stack, queue, Linked lists In the situation where there are multiple users or a network computer system, you probably share a printer with other users. When you request to print a file, your request is added to the print queue. When your request reaches the front of the print queue, your file is printed. This ensures that only one person at a time has access to the printer and that this access is given on a first-come, first served basis. Design and implement the algorithm for this scenario.	4 hours

2. Non-linear data structures: Trees & Graphs You are making an iPod playlist to hear the songs. Assuming that shuffle functions are not applicable, choose an appropriate data structure that will add and delete songs onto you're your iPod in such a way that the recently inserted song will always be the first song currently on the iPod.	4 hours
3. Graph algorithms: BFS, DFS, network flow You are developing a diagnostic tool for a large, complex communication network (e.g., an internet service provider's infrastructure, or a peer-to-peer file-sharing network). The network shown below is an undirected, unweighted graph, where devices are nodes (routers, servers, end-points) which are connected directly by edges. Nodes: A, B, C, D, E, F, G, H, I, J Edges: A-B, A-C B-D, B-E C-E, C-F, D-G, E-H, F-I, G-J, H-J, I-J. Implement functionalities primarily using Breadth-First Search (BFS) with required assumptions to answer the following queries: •□□□□□□ Given a source_device and a target_device, find the minimum number of hops (direct connections) required to establish a path between them. If no path exists, indicate so. •□□□□□□ Given a starting_device and a max_hops limit, determine all devices that can be reached from the starting_device within max_hops. This could represent the reach of a network broadcast or a virus. •□□□□□□ Given a specific pair of source_device and target_device, identify all "bridge devices" along every shortest path between them. A device is a "bridge device" in this context if its removal would increase the shortest path distance between the source_device and target_device or disconnect them entirely. This requires careful application or multiple runs of BFS. •□□□□□□ Given a starting_device, find the diameter of the connected component to which it belongs. The diameter is the longest shortest path between any two nodes in that component.	4 hours
4. Trees: AVL, Red-Black tree You are tasked with designing the core data structure for a large e-commerce platform's product catalog. Products are organized in a strict multi-level hierarchy (e.g., Electronics -> Laptops -> Gaming Laptops -> BrandX Gaming Laptop Model Y). Each product has a unique product_ID, a price, and a	4 hours

quantity_in_stock. The catalog is highly dynamic, with frequent additions, removals, price changes, and stock updates. Implement a robust and efficient tree-based data structure (e.g., a balanced Binary Search Tree like an AVL Tree or Red-Black Tree, where nodes can represent categories or individual products) to manage this hierarchical product catalog. Your implementation must support the CRUD operations.	
5. Divide & Conquer: Merge sort, Quick sort, Karatsuba's fast multiplication method You have n coins, all of which are gold except one coin which appears to be a gold coin, but it is fake. All gold coins are of the same weight, the fake coin weighs less than the others. You have a balance scale, you can put any number of coins on each side of the scale at one time and it will tell you if the two sides weight the same, or which side is lighter if they don't weight the same. The problem is to identify the fake coin. Design an algorithm to find the fake coin in the given n coins.	4 hours
6. Dynamic programming: MCM, LCS A chain of matrices A is called as Add-multiply-chain(AM-chain) if all the matrices of the chain are sqaure matrices of the same size and both addition & multiplication operations are involved in the chain. For example, $A = A_1 * A_2 * A_3 + A_4 * A_5 * A_6 * A_7 + A_8 * A_9 * A_{10}$ is an AM-chain. The above AM-chain A is written as (A, 10, 3, 7), which conveys that the chain A has 10 matrices, with two '+'operations, one '+'operation after A3, another '+' operation after A7, and multiplication is the operation in all the required places in the chain. Multiplication (or addition) between any two numbers is called a scalar operation. The number of scalar operations involved in the chain $A_1A_2 + A_3$ is 12, where all three matrices are square matrices of size 2. (A, n, i, j) is a chain A with n matrices with a '+' operation after A_i, another '+' operation after A_j, and all the other operations are multiplication. Given an AM-chain (A, n, i, j), design a pseudocode and implement it to parenthesize the chain A such that the number of scalar operations to evaluate the chain A is minimum.	2 hours
7. Backtracking: N-Queens problem, Subset Sum, Graph Coloring You have a standard 8x8 chessboard (if your are not familiar with game of chess, please get to know), empty but for a single knight on some square. Your task is to generate a series of legal knight moves that result in the knight visiting every square on the chessboard exactly once. In addition, the knight	2 hours

must end on a square that is one knight's move from the beginning square. The output of your program should indicate the order in which the knight visits the squares, starting with the initial position. Generalize your program for an n x n board where n>8.	
8. Branch & Bound: A-Star, Subset Sum You are developing the pathfinding core for a logistics company using autonomous delivery vehicles in a dense urban environment. The city map is represented as a grid, where each cell has an associated "traffic cost" (representing time delay due to congestion, road conditions, etc.). Some cells may be temporarily designated as "restricted zones" or "accident sites" with extremely high or infinite costs. Vehicles can move horizontally, vertically, and diagonally. You need to find the most cost-effective path from a starting depot to a delivery destination. Implement the A* search algorithm to find the minimum-cost path from a start_cell (start_row, start_col) to a target_cell (target_row, target_col) within a given M x N integer grid.	2 hours
9. Approximation Algorithm: Vertex Cover, Set Cover You are a network security engineer responsible for monitoring a critical communication network. The network consists of various communication nodes (e.g., servers, routers, switches) and the direct communication links between them. To detect potential intrusions or failures, you need to install monitoring agents on a subset of these nodes. An agent on a node can monitor all direct communication links connected to that node. Your goal is to select the minimum number of nodes to place agents on, such that every communication link in the network is monitored. However, finding the absolute minimum is a known NP-hard problem for large networks, so an efficient approximation is required. Implement a 2-approximation algorithm for the Minimum Vertex Cover problem.	4 hours
Total Laboratory Hours:	**30 hours**

Mode of Evaluation : Continuous Assessment Test, Final Assessment Test	
Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE512	Operating Systems	3	0	2	4
Pre-requisite	Nil	Syllabus Version			
		1			

Course Objectives

1. To provide a comprehensive understanding of the evolution and fundamental components of modern operating systems, including process/thread management and virtualization technologies.
2. To equip students with the ability to design and analyze system-level mechanisms such as CPU scheduling, memory and file management, and synchronization in both traditional and real-time environments.
3. To prepare students for modern computing environments by exploring cloud OS concepts, containerization, and security techniques in distributed and edge systems.

Course Outcomes

1. Explain the structure and key components of an operating system including system calls, interfaces, and virtualization techniques.
2. Analyze process lifecycle, scheduling techniques, and implement synchronization mechanisms in multi-threaded environments.
3. Demonstrate memory management techniques such as segmentation, paging, and virtual memory across architectures.
4. Evaluate storage management approaches, disk scheduling algorithms, and file system implementation strategies.
5. Apply protection and security mechanisms in OS and explain the role of containers and distributed system management.

Module:1	Introduction	5 hours

Operating System Components – Design Architectures: monolithic, Layered, Microkernel, Modular, Hybrid – Operating System Operations: Interrupt Handling, Dual Mode Operation, Multitasking, Context Switching - Process Management -Memory Management - Storage Management - Protection and Security - Kernel Data Structures – Open Source Operating Systems: Definition, Types, Structures, Uses - Operating System Interface Types: Command Line, Graphical User, Touch Based - System Calls -Types of System Calls - System Programs - Virtualization Basics: OS-Level vs Hardware Level Virtualization – Hypervisors

Module:2	Process Management	11 hours

Process Concept - Process Scheduling - Operations on Processes - Inter-process Communication - Multicore Programming - Multithreading Models -Thread Libraries - CPU Scheduling - Scheduling Criteria - Scheduling Algorithms - Thread Scheduling -Multiple-Processor Scheduling - Real-Time OS Scheduling (Rate Monotonic, EDF) and Schedulability analysis - Process Synchronization - The Critical-Section Problem - Peterson's Solution - Synchronization Hardware -

Mutex Locks – Semaphores – Monitors -Classic Problems of Synchronization – Deadlock – Prevention – Avoidance- Detection – Recovery		
Module:3	**Memory Management**	**9 hours**
Contiguous Memory Allocation – Segmentation - Paging - Structure of the Page Table - Intel 32 and 64-bit Architectures - ARM Architecture - Virtual Memory - Demand Paging -Copy-on-Write - Page Replacement- Allocation of Frames - Thrashing - Allocating Kernel Memory		
Module:4	**Storage Management**	**9 hours**
Mass-Storage Structure - Disk Structure - Disk Attachment - Disk Scheduling - Disk Management - Swap-Space Management - File-System Interface - File Concept - Access Methods - Directory and Disk Structure - File-System Mounting - File Sharing - Protection - File-System Implementation -File-System Structure - Implementation - Directory - Allocation methods - NFS , EXT4 File Systems - I/O Systems - I/O Hardware - Application I/O Interface - Kernel I/O Subsystem - Distributed Storage Management		
Module:5	**Protection Security and Containerization**	**9 hours**
Protection - Domain of Protection - Access Matrix - Implementation of the Access Matrix -Access Control - Revocation of Access - Security - Program Threats- System and Network Threats - User Authentication - Implementing Security Defenses - Firewalling - Systems and Networks - Linux System Case study - Distributed Systems Basics - Containerization: Docker and Kubernetes		
Module:6	**Contemporary Issues**	**2 hours**
Contemporary Issues		
	Total Lecture Hours:	**45 hours**
Text Book(s)		
Abraham Silberschatz, Peter B. Galvin, Greg Gagne, " **“Operating System Concepts”"**, Wiley United States, 10th Edition, **2018** Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau , **"“Operating Systems: Three Easy Pieces”"**, Arpaci Dusseau Books, 1st Edition, **2023**		
Reference Books		
Thomas Anderson, Michael Dahlin, " **“Operating Systems: Principles and Practice”** ", Recursive Books Austin Texas USA, 2nd Edition, **2020** Kaiwan N. Billimoria , **"“Linux Kernel Programming”"**, Packt Publishing Birmingham United Kingdom, 2nd Edition, **2022**		
Indicative Experiments		

1. To familiarize students with basic Linux/Unix shell commands used for file handling, process management, user administration, and system navigation.	3 hours
2. To set up and configure a virtualization environment using tools such as Virtual Box, VMware, or KVM, and understand the basics of virtual machines and hypervisors.	3 hours
3. To implement and observe process creation and control using system calls such as fork(), exec(), wait(), and exit() in a Linux environment.	3 hours
4. To demonstrate the creation and execution of threads using the POSIX pthread library, and understand the basic concepts of multithreading and thread management.	3 hours
5. To simulate and compare CPU scheduling algorithms such as FCFS, SJF, Round Robin, and Priority Scheduling and analyze their performance based on turnaround time and waiting time.	3 hours
6. To simulate real-time scheduling algorithms such as Rate Monotonic Scheduling (RMS) and Earliest Deadline First (EDF), and evaluate their schedulability in real-time systems.	3 hours
7. To simulate deadlock detection in a multi-process system using resource allocation matrices and implement a deadlock detection algorithm (e.g., Banker's Algorithm or Wait-for Graph).	3 hours
8. To implement process synchronization using semaphores in producer-consumer or reader-writer problem scenarios and understand race conditions and critical section handling.	3 hours
9. To simulate dynamic memory allocation techniques such as First Fit, Best Fit, and Worst Fit and evaluate memory utilization and fragmentation.	3 hours
10. To simulate and evaluate page replacement algorithms such as FIFO, LRU, and Optimal and compare their performance in terms of page faults.	3 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment	
Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE513	Computer Networks	3	0	2	4
Pre-requisite	Nil	Syllabus Version			
		1			

Course Objectives

1. To impart knowledge of various network models, their layered architecture, and associated protocols.
2. To equip with the foundational knowledge of routing algorithms and their applications in networking.
3. To facilitate the comprehension of the fundamentals, architecture, and characteristics of wireless and mobile networks.

Course Outcomes

1. Explore the Internet model and various network performance metrics.
2. Interpret the application layer protocols, services and evaluate the requirements for reliable services and implications of congestion at the transport layer services.
3. Analyze various functionalities required in the control and data plane at network layer services.
4. Explain the functions of data link layer including error control and access control.
5. Evaluate the characteristics of wireless and mobile networks, including their security protocols and standards.

Module:1	Computer Networks and Internet	9 hours

Internet: A Nuts-and-Bolts Description - Network Protocols - The Network Edge: Access Networks and Physical Media - The Network Core: Packet Switching, Circuit Switching - Network of Networks - Delay, Loss and Throughput in Packet-Switched Networks – Protocol Layers and Their Service Models(TCP/IP)- Principles of Network Applications: Architectures

Module:2	Application Layer and Transport Layer	10 hours

Application Layer: The Web and HTTP - Electronic Mail in the Internet, FTP - DNS—The Internet's Directory Service - Peer-to-Peer File Distribution - Transport Layer: Relationship Between Transport and Network Layers - Overview of the Transport Layer in the Internet - Multiplexing and Demultiplexing - Connectionless Transport: UDP – Reliable Data Transfer: Go-Back-N (GBN) and Selective Repeat (SR) - Connection-Oriented Transport: TCP, Flow Control and Congestion Control - Socket Programming: Creating Network Applications: TCP/UDP socket programming

Module:3	Network Layer	10 hours

Network Layer – Router - The Internet Protocol (IP): IPv4, Addressing and IPv6 – Generalized Forwarding and SDN - Control Plane: Per-router control and logically centralized control - Routing Algorithms - Link-State (LS) Routing Algorithm,

Distance-Vector (DV) Routing Algorithm, Intra-AS Routing in the Internet: OSPF and Routing Among the ISPs: BGP - SDN Control Plane		
Module:4	**Link Layer and LANs**	**8 hours**
Overview of Link Layer Services - Error-Detection and -Correction Techniques: Checksum and CRC - Multiple Access Links and Protocols: Channel Partitioning Protocols and Random-Access Protocols - Switched Local Area Networks: Link-Layer Addressing and ARP - Virtual Local Area Networks		
Module:5	**Wireless and Mobile Networks Security**	**6 hours**
Elements of a wireless network - Wireless Links and Network Characteristics - WiFi: 802.11 Wireless LANs - Mobility Management: Principles - Wireless and Mobility: Impact on Higher Layer Protocol- Security in Computer Network- Message Integrity and Digital Signatures - Network-Layer Security: IPsec and Virtual Private Networks		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

James F. Kurose, Keith W. Ross, "**Computer Networking: A Top-Down Approach**", Pearson, United Kingdom., 8th Edition, **2022**

Reference Books

Andrew S. Tanenbaum, "**Computer Networks**", Pearson, Singapore, 6th Edition, **2022**

Larry Peterson and Bruce Davie, "**Computer Networks: A Systems Approach**", Morgan Kaufmann, United States of America., 6th Edition, **2021**

Larry Peterson and Bruce Davie, "**Computer Networks: A Systems Approach**", Morgan Kaufmann, United States of America, 6th Edition, **2021**

Indicative Experiments

1. Hardware Demo (Demo session of all networking hardware and Functionalities) OS Commands (Network configuration commands): □ □ □ □□Identify and understand the functionalities of networking hardware (e.g., routers, switches, hubs, cables). □ Execute network configuration commands (e.g., ifconfig, ipconfig, ping, tracert, netstat) on operating systems like Linux/Windows to configure and troubleshoot network settings.	2 hours

□ □□Document the purpose and output of at least ten commands and describe the role of each hardware component.	
2. Network Packet Analysis using Wireshark: □ □ □ □ □ □ □ □ □ □**Packet Capture**: Capture live network traffic from a specified interface (e.g., Ethernet, Wi- Fi) for at least 5 minutes. □ □ □ □ □**Starting Wireshark**: Configure Wireshark with appropriate capture filters to focus on specific protocols (e.g., HTTP, TCP). □ □ □ □ □□**Viewing Captured Traffic**: Analyze captured packets, identifying key fields like source/destination IP, protocol, and payload. □ □ □ □ □**Analysis, Statistics, and Filters**: Generate statistical reports (e.g., protocol hierarchy, conversations) and apply display filters to isolate specific traffic (e.g., tcp.port == 80).	3 hours
3. Socket programming (TCP and UDP), Multi-client chatting: □ □ □ □□Develop a client-server application using socket programming in C language. Implement a TCP/UDP based server that handles multiple clients simultaneously for a chat application. □ □ □ □ □Implement a TCP/UDP based client-server application for simple message exchange. □ □ □ □ □Test the applications by running multiple clients and verifying message delivery.	4 hours
4. IP addressing, Classless addressing: □ □ □ □ □Configure IP addresses on a simulated network using tools like Cisco Packet Tracer. □ □ □ □ □Assign IP addresses to devices using both classful and classless (CIDR) addressing schemes. □ □ □ □ □Calculate subnets for a given network (e.g., divide a /24 network into subnets). □ □ □ □ □Verify connectivity using ping and document the addressing scheme. □□□□□□ Implement a simple client-server program where the server listens on a configured IP address and port. The client sends a message containing classless IP addresses to the server. The server	3 hours

calculates the first address, last address, and other related information. Test the program to ensure successful communication.	
5. Performance evaluation of routing protocols using simulation tools: □ □ □ □□Use the NS-3 simulator to evaluate the performance of routing protocols (e.g., RIPng, OSPFv3). □ □ □Write a C++ program in NS-3 to set up a network topology with at least five routers, configuring IPv6 addresses and enabling routing protocols (e.g., RIPng and OSPFv3). □ □ □ □Simulate traffic using NS-3's traffic generation tools (e.g., OnOffApplication) and measure performance metrics such as convergence time (using routing table updates), packet loss, and throughput (using FlowMonitor). □ □ □ □ □□Analyze the routing tables generated by each protocol to determine the shortest path between nodes. Compare the performance of at least two protocols (e.g., RIPng vs. OSPFv3) based on the collected metrics.	3 hours
6. SDN Applications and Use Cases Explore Software-Defined Networking (SDN) using a tool like Mininet. □□□□□□□ Create a simple SDN topology with a controller (e.g., OpenDaylight, Ryu). •□□□□□□□□ Implement a use case, such as traffic engineering or load balancing. •□□□□□□□□ Test the setup by generating traffic and analyzing controller behavior.	2 hours
7. Error detection and correction mechanisms, Flow control mechanisms: □ □ □ □□Implement and analyze error detection/correction and flow control mechanisms. □ □ □ □ □Write a program to simulate error detection (e.g., CRC, checksum) and correction (e.g., Hamming code) on sample data. □ □ □ □ □Implement a flow control mechanism (e.g., sliding window protocol) in a simulated environment. □ □ □ □ □Test the implementations with different error rates and window sizes.	4 hours

8. Wired and Wireless Network Simulation Using Cisco Packet Tracer: □ □ □ □ □ □Use Cisco Packet Tracer to design and simulate a network. □ □ □ □ □Create a topology with both wired (e.g., Ethernet) and wireless (e.g., Wi-Fi) components, including at least two routers, two switches, and five end devices. □ □ □ □ □Configure network settings (e.g., IP addresses, VLANs, SSIDs). Test connectivity and simulate data transfer.	3 hours
9. Security in Network - Use cases: □ □ □ □ □Explore network security use cases. □ □ □ □ □Set up a network in a simulation tool (e.g., GNS3, Packet Tracer) and implement at least two security mechanisms (e.g., ACLs, VPN, firewall rules). □ □ □ □ □Simulate an attack (e.g., unauthorized access attempt) and demonstrate how the security measures prevent it. □ □ □ □ □Document the setup and results.	3 hours
10. Network Security with IPsec and Message Integrity: □ □ □ □ □Use GNS3 to configure a secure network and implement message integrity checks. □ □ □ □ □Set up a network topology with two routers and two end devices, configuring an IPsec VPN tunnel between the routers to secure communication using ESP (Encapsulating Security Payload) in tunnel mode. □ □ □ □ □Write a C program to implement message integrity using HMAC-SHA256 for a simple client-server application running on the end devices. The program will compute and verify digital signatures for messages exchanged over the VPN. □ □ □ Simulate data transfer through the VPN and verify that the IPsec tunnel encrypts traffic (using Wireshark to inspect packets). Test the HMAC implementation by introducing tampered messages and checking detection. Submit the GNS3 configuration file, the C source code for the HMAC implementation, a report detailing the IPsec setup, packet capture analysis, and results of the message integrity tests.	3 hours

Submit the GNS3 configuration file, the C source code for the HMAC implementation, a report detailing the IPsec setup, packet capture analysis, and results of the message integrity tests.	
Total Laboratory Hours:	**30 hours**

Text Book(s)
James F. Kurose, Keith W. Ross, " **Computer Networking: A Top-Down Approach**", Pearson, United Kingdom, 8th Edition, **2022**
Reference Books
Andrew S. Tanenbaum, "**Computer Networks**", Pearson, Singapore, 6th Edition, **2022**
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Oral Examination, Seminar, Group Discussion

Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE517	Computer Architecture and Organization	3	0	2	4
Pre-requisite	Nil	Syllabus Version			
		1			

Course Objectives

1. To provide knowledge on the basics of computer architectures and organization that lays the foundation to study high-performance architectures.
2. To impart skills to design and develop parallel programs using parallel computing platforms such as OpenMP, CUDA.
3. To familiarize on the domain-specific architectures and real-world implementations in modern processors.

Course Outcomes

1. Outline the developments in the evolution of computer architectures and parallel programming paradigms.
2. Design and evaluate memory hierarchies including multi-level cache systems and virtual memory.
3. Develop parallel programs using OpenMP and analyze performance parameters such as speed-up, and efficiency for parallel programs against serial programs.
4. Analyze GPU and SIMD architectures and implement data-level parallelism using CUDA.
5. Compare domain-specific architectures and evaluate their impact in modern data center applications.

Module:1	Fundamentals of Computer Architecture	5 hours

Introduction to Parallelism-Flynn's Classification: SISD, SIMD, MISD, MIMD- Pipelining- Instruction and Arithmetic Pipelines, Superscalar and Super Pipelined Architectures- Metrics for Performance Measurement.

Module:2	Memory Hierarchy Design	8 hours

Optimizations of Cache Performance - Memory Technology and Optimizations- Virtual Memory and Virtual Machines -The Design of Memory Hierarchy-Case Study: Memory Hierarchies in Intel Core i7 and ARM Cortex-A8.

Module:3	Instruction-Level and Thread-Level Parallelism	11 hours

Instruction-level Parallelism: Concepts and Challenges- Basic Compiler Techniques for Exposing ILP –Dynamic Scheduling-Limitations of ILP- Multithreading: Exploiting Thread-Level Parallelism –Shared Memory Multicore Systems, Performance Metrics for Shared-Memory Multicore Systems- Multithreaded Programming using OpenMP.

Module:4	Data-Level Parallelism in Vector, SIMD, and GPU Architectures	10 hours

Introduction, Vector Architecture, SIMD Instruction Set Extensions for Multimedia, Graphics Processing Units, GPU Memory Hierarchy, Detecting and Enhancing Loop- Level Parallelism, CUDA Programming, Case Study: Nvidia Maxwell.		
Module:5	**Domain-Specific Architectures**	**9 hours**
Introduction - Guidelines for DSAs - Google's Tensor Processing Unit: An Inference Data Center Accelerator - Microsoft Catapult: a Flexible Data Center Accelerator - Intel Crest: a Data Center Accelerator for Training.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

J.L. Hennessy and D.A. Patterson. , "**Computer Architecture: A Quantitative Approach.**", Morgan Kauffmann Publishers, 6th Edition, **2019**
Gerassimos Barlas, "**Multicore and GPU Programming: An Integrated Approach.**", Morgan Kaufmann, 2nd Edition, **2022**

Reference Books

William Stallings., "**Computer Organization and Architecture: Designing for Performance.**", Pearson, 11th Edition, **2022**
T. Deakin and T.G. Mattson. , "**Programming Your GPU with OpenMP: Performance Portability for GPUs.**", MIT Press Cambridge, 1st Edition, **2023**
J. Sanders and E. Kandrot, " **CUDA by Example: An Introduction to General-Purpose GPU Programming.**", Addison-Wesley Boston, 1st Edition, **2010**

Indicative Experiments

1. Create a new project using Visual Studio or an equivalent IDE that supports OpenMP. Configure the project properties to enable OpenMP support, compile, and execute the basic "hello world" program.	2 hours
2. Construct an OpenMP program using the parallel construct that prints the thread number from each thread in the parallel region to demonstrate how multiple threads execute concurrently.	2 hours
3. Develop a OpenMP program that calculates the execution time of a computational task using timing function and performance counters. Compare their properties.	2 hours
4. Implement an OpenMP program using various environment routines to access processor run-time information that simulates a group of students (threads) solving a set of problems simultaneously. Each student reports their ID, total students participating, maximum possible students, and processor details.	2 hours
5. Write an OpenMP program that simulates operations on a deck of cards. The program should use the loop construct to shuffle the deck in	4 hours

parallel, the sections construct to deal cards concurrently to two players, and the single construct to print a game start announcement exactly once. Use these constructs to demonstrate how parallel tasks can be organized efficiently. Also, explain the purpose and behavior of each construct.	
6. Generate an OpenMP program to simulate a bakery model with 20 customer orders for varying preparation times. Use the static, dynamic, and guided scheduling clauses to assign orders to multiple threads. Observe and compare how each scheduling method affects workload distribution and overall execution time.	2 hours
7. Use Pin instrumentation and Cache grind profiling to analyze cache usage and identify where cache misses happen.	2 hours
8. Track branch predictions and mispredictions during execution and identify inefficient branching and improve performance.	2 hours
9. Develop a CUDA-platform setup on NVIDIA / Google Colab.	2 hours
10. Build a CUDA program in C/C++ environment to add two large arrays using unified memory, validate the result with a CPU implementation, and measure GPU kernel execution time using CUDA events.	2 hours
11. Write a CUDA program in C/C++ environment to reverse the elements of a large array using a single thread block, utilize shared memory for efficient access, include boundary checks, and measure the execution time using CUDA events.	2 hours
12. Implement a CUDA program in C/C++ environment to perform matrix addition and multiplication using shared memory optimization, validate the results with CPU computation, and compare execution times using CUDA event profiling.	2 hours
13. Hands-on Lab: CNN Training with TensorFlow on TPUs.	4 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment	
Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE518	Database Modeling and Design	3	0	2	4
Pre-requisite	Nil	Syllabus Version			
		1			

Course Objectives

1. To inculcate the foundational knowledge of database concepts, ER modeling, relational algebra, and optimization techniques for effective database design.
2. To equip with advanced skills in EER modeling, complex relationship modeling, schema mapping, and transactional programming using SQL and PL/SQL.
3. To impart skills in normalization, distributed systems, advanced data modeling, and cutting-edge NoSQL and Big Data technologies.

Course Outcomes

1. Comprehend the fundamental of database systems and its architectures.
2. Model the database systems using ER and EER to design complex database systems.
3. Analyze and apply normalization theory to relational schemas and optimize the design for minimal redundancy.
4. Gain insight into advanced database technologies for handling complex problems.
5. Design NoSQL databases and handle big data to solve real-world problems.

Module:1	Introduction to Database Systems	6 hours

Data, Information, Metadata, Data Management - Purpose of Database– Data Models and concepts- Database System Components - Three Schema Architecture– Centralized and Client/Server Architectures–Conceptual Design using ER Modeling: Entities, Attributes, Relationships, Integrity Constraints, Strong Entities, Weak Entities– ER Modeling Process.

Module:2	Database Modeling and Database Languages	10 hours

Enhanced Entity-Relationship (EER) Modeling: Subclasses, Superclasses, and Inheritance– Specialization and Generalization– Constraints– Modeling of UNION Types Using Categories- Ternary and Higher Order Relationships, Recursive Relationships– Mapping ER and EER Models to Relational Schema– Introduction to Relational Algebra: Basic Operators: Selection, Projection, Join, Union, Intersection– Views and Materialized Views– Query Optimization– Indexing. Database Languages: Data Definition Language (DDL)– Data Manipulation Language (DML)– Data Control Language (DCL)– Display Queries (DQL)– Joins, Subqueries– Transaction Control Language (TCL)– COMMIT, ROLLBACK, SAVEPOINT– PL/SQL– Cursors, Triggers.

Module:3	Relational Database Design and Transaction Management	10 hours

Normalization – Functional Dependencies: Concept, Equivalence Set– Minimal Cover, Attribute Closure– Deriving Candidate Keys and Purpose of Keys– Normal Forms: 1NF, 2NF, 3NF, BCNF– Comprehensive Normalization Examples– Dependency Preserving Decomposition– Lossless Join Decomposition– Higher Normal Forms: Multi-valued Dependencies, 4NF, Join Dependencies, 5NF, Domain-Key Normal Form (DKNF) –Transactions – Concepts of Transactions– Serializability– Concurrency Control – Database Recovery.		
Module:4	**Advanced Database Systems**	**10 hours**
Parallel Databases: Architecture, Interoperation Parallelism and Intraoperation parallelism – Data Partitioning Strategies– Query Optimization for Parallel Execution– Distributed Database Concepts– Features– Fragmentation, Allocation, and Replication– Concurrency Control and Recovery in Distributed Databases– Query Processing and Optimization in Distributed Databases– Spatial Database– Temporal Database– Multimedia Database Concepts– Graph Databases - Knowledge Graphs.		
Module:5	**NoSQL Databases and Big Data Systems**	**7 hours**
Semi-Structured Data Models: JSON, XML, RDF– NoSQL Databases: MongoDB, Key–Value Stores– Big Data Systems: HDFS Overview, Apache Spark– Cloud Data Storage.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

R. Elmasri and S. Navathe, "**Fundamentals of Database Systems**", Addison-Wesley, 7th Edition, **2021**

D Abraham Silberschatz, Henry F. Korth, S. Sudarshan , "**Database System Concepts**", McGraw Hill, 7th Edition, **2021**

Reference Books

Edward Sciore, "**Database Design and Implementation: Second Edition (Data-Centric Systems and Applications)**", Springer, 2nd Edition, **2020**

Sadalage & Fowler, "**NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence**", Addison-Wesley, 2nd Edition, **2021**

Guy Harrison, "**Next Generation Databases: NoSQL, NewSQL, and Big Data**", Apress, 2nd Edition, **2020**

Indicative Experiments

1. Design an EER diagram for a Research Collaboration Portal. • Entities: Professors, Students, Projects, Publications, FundingAgencies. • Generalization: Person (Professor, Student).	2 hours

• Specialization: Professors can be Principal Investigator (PI) or Co-Investigator (Co-PI). • Recursive Relationship: Professors mentoring junior faculty or students. • Ternary Relationship: A project is supervised by a PI, funded by a FundingAgency, and has at least one student. • Constraints: One project cannot exceed ₹10,00,000 in funding per year. A student cannot be on more than 2 active projects. Cardinality, Participation Tools: Lucidchart, Draw.io, MySQL Workbench.	
2. Relational Schema Mapping: Transform the EER diagram from lab 1 into its corresponding relational schema representation. List all resulting tables with primary keys, foreign keys, and attributes. • Create and populate database tables using appropriate DDL and DML commands. • Specify how the following are handled: Superclass-subclass mapping strategy (e.g., one table per subclass). Recursive relationship. Ternary relationship. Constraints (primary key, foreign key, check, default, not null, unique). Platform: MySQL / PostgreSQL / Oracle.	2 hours
3. Execute advanced SQL queries on the relational schema derived from the EER diagram. This includes retrieving, filtering, and analyzing data using: • Aggregate functions (SUM, AVG, MAX, MIN, COUNT) • GROUP BY, HAVING clauses • Nested subqueries (scalar, correlated, multi-row) • Window functions (ROW_NUMBER, RANK) Platform: MySQL / PostgreSQL / Oracle.	2 hours
4. Develop and execute advanced SQL queries on the relational schema derived from the EER diagram, focusing on advanced data retrieval, filtering, and analysis. This includes the use of: • Standard Joins: Inner Join, Outer Join (Left, Right, Full), Self-Join • Special Joins: Cross Join, Natural Join • Other Joins: Equi Join, Non-Equi Join, Semi Join, Anti Join (using EXISTS / NOT EXISTS) Platform: MySQL / PostgreSQL / Oracle.	2 hours

5. DCL and TCL operations on the relational schema derived from the Research Collaboration Portal. This includes managing user permissions and ensuring consistency of transactional operations using: • DCL Commands: GRANT, REVOKE to control access privileges on tables such as Professors, Projects, and Publications. • TCL Commands: COMMIT, ROLLBACK, and SAVEPOINT to manage and recover database states during multi-step operations like funding updates or project allocations. Platform: MySQL / PostgreSQL / Oracle.	2 hours
6. Using the relational schema of the Research Collaboration, perform the following tasks: • Create indexes based on query requirements. • Define sequences for automatically generating unique IDs (Oracle uses sequences, MySQL uses AUTO_INCREMENT, and PostgreSQL supports both (SERIAL / SEQUENCE)). • Create views to simplify common queries. Platform: MySQL / PostgreSQL / Oracle.	2 hours
7. PL/SQL – Procedures and Functions: Implement and execute stored procedures and functions. Using the Research Collaboration Portal relational schema, implement and execute stored procedures and functions to automate and encapsulate common operations. Tasks: • Create a stored procedure to add a new project. The procedure should accept project details (title, PI ID, funding agency ID, funding amount, start and end dates) and validate that the funding amount does not exceed ₹10,00,000. If the validation fails, rollback the transaction and raise an appropriate error. • Create a function to count the number of active projects for a given student ID. The function should return the count of projects where the current date falls between the project start and end dates. • Create a stored procedure to assign a student to a project only if the student is currently involved in fewer than 2 active projects. If the student is eligible, insert the assignment; otherwise, raise an error. • Create a function that returns the total funding amount received by a specific professor (as PI) across all projects. Platform: MySQL / PostgreSQL / Oracle.	2 hours
8.	2 hours

PL/SQL – Cursors and Triggers: Using the Research Collaboration Portal schema, implement and execute PL/SQL cursors and triggers to handle complex data retrieval and enforce data integrity. Tasks: • Write an explicit cursor to fetch and display all projects along with their Principal Investigator's name and total funding amount. Include cursor open, fetch, and close operations with appropriate exception handling. • Use an implicit cursor to count the number of publications authored by each professor and display the results. • Create a BEFORE INSERT trigger on the Projects table to validate that the funding amount does not exceed ₹10,00,000. If the validation fails, raise an application error to prevent insertion. • Create an AFTER INSERT trigger on the Project_Student table to check that a student is not assigned to more than 2 active projects. If violated, raise an error and roll back the insert. • Create a BEFORE UPDATE trigger on the Professors table to prevent changing the designation of a professor from "Principal Investigator" to any other value if they currently supervise active projects. Platform: MySQL / PostgreSQL / Oracle.	
9. Design a MongoDB-based Student Management System where: • Each student has a unique ID, name, age, and department. • Each student is enrolled in multiple courses. • Each course includes course_code, course_name, credits Platform: MongoDB	2 hours
10. Spatial Databases using GIS Coordinates: Design a spatial database for managing university campuses, department buildings, and student hostels. Tasks: • Store GPS locations of all buildings • Find hostels within 1 km of academic buildings • Identify the nearest cafeteria to each department • Visualize results using GeoJSON exports Platform: PostgreSQL/PostGIS or MongoDB with GeoJSON	2 hours
11. Build a basic application that interacts with cloud-based databases or Big Data frameworks for scalable data storage, retrieval, and analysis. Tasks: • Connect and interact with a cloud database (e.g., Firebase, RDS, BigQuery) • Perform CRUD operations over the cloud	2 hours

• Integrate data ingestion and processing with Big Data tools (e.g., Spark or Hadoop) • Analyze large datasets in a distributed environment • Understand deployment, scalability, and cloud storage optimization Tools: Firebase Realtime DB / Firestore or AWS RDS / MongoDB Atlas or Google BigQuery / Azure Cosmos DB.	
Total Laboratory Hours:	**30 hours**
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Presentaion	
Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE519	Machine Learning and Applications	3	0	2	4
Pre-requisite	Nil	Syllabus Version			
		1			

Course Objectives

1. To deliver insights into the foundational and mathematical principles of machine learning algorithms.
2. To impart skills to select and apply appropriate machine learning models to address real-world problems across diverse domains.
3. To explore emerging trends in machine learning and its advancements emphasizing societal impact and ethical considerations

Course Outcomes

1. Comprehend the fundamental properties of machine learning techniques.
2. Design and Develop supervised learning techniques for real-world data challenges.
3. Evaluate ensemble methods to improve model accuracy and robustness.
4. Develop an appropriate unsupervised learning model to manage unknown patterns.
5. Apply Reinforcement Learning concepts and algorithms for intelligent system design.

Module:1	Introduction	6 hours

Overview of Machine Learning (ML) Concepts - Applications - Types of ML algorithms - Hypothesis space and Inductive Bias - Data - Data handling - Model Selection and Evaluation: Train test split and cross-validation - Evaluation metrics

Module:2	Supervised Learning	10 hours

Linear Regression: Simple and Multiple Linear regression - Evaluating regression fit – Non-Linear Regression – Applications: Predicting Sales, Predicting price of Real Estate, Predicting Concrete Compression Strength - Multi-Class and Multi-Label classification -Decision tree learning: Decision tree representation - Decision tree algorithms- Hypothesis space search in decision tree learning - Inductive bias in decision tree learning - Instance based Learning: K nearest neighbor - Support Vector Machine (SVM) - Dual formulation - Maximum margin with noise - Nonlinear SVM and Kernel function.

Module:3	Probabilistic and Ensemble Learning	9 hours

Probability and Bayes Learning: Logistic Regression - Bayesian Learning, Naïve Bayes - Bayesian Belief Networks - Artificial Neural Networks (ANN): Biological motivation - ANN representation - Perceptron - Multilayer perceptron and the back propagation algorithm - Ensembles: Bias-Variance Tradeoff – Bagging - Random

Forest, Boosting – Adaboost - Applications: Predicting Fraud Insurance Claims, Email Spam filtering		
Module:4	**Unsupervised Learning**	**10 hours**
Clustering- Partitional Clustering: k-means - k-medoids - Hierarchical Clustering: Agglomerative - Divisive - Density based clustering: DBSCAN-Spectral clustering - Dimensionality Reduction: Principal Component Analysis - Linear Discriminant Analysis - Expectation Maximization - Mixture of Gaussians- Outliers- Anomaly Detection – Applications :Document Clustering - Customer Segmentation		
Module:5	**Advanced Machine Learning Concepts**	**8 hours**
Self-supervised Learning - Reinforcement Learning (RL) - RL Framework - Markov Decision Process - Policy - Planning algorithms - Q Learning algorithm - Online learning algorithms : Winnow algorithm - On-line to batch conversion - Overview of MLOps - Explainability in ML - Ethics in ML.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar , "**Foundations of Machine Learning**", MIT Press, 2nd Edition, **2018**
Ethem Alpaydin, "**Introduction to Machine Learning**", MIT Press Prentice Hall of India, 4th Edition, **2020**

Reference Books

Tom Mitchell, "**Machine Learning**", McGraw Hill, 3rd Edition, **1997**
Shai Shalev-Shwartz and Shai Ben-David, "**Understanding Machine Learning: From Theory to Algorithm**", Cambridge University Press, **2014**
Stephen Marsland , "**Machine Learning: An Algorithmic Perspective**", Chapman and Hall CRC press, 2nd Edition, **2015**
Danish Haroon, "**Python Machine Learning Case Studies: Five Case Studies for the Data Scientist**", Apress, **2017**

Indicative Experiments

1. To effectively **predict alumni donation likelihood**, the university employs a comprehensive preprocessing strategy on its dataset. Numerical features such as annual income (in lakhs) and previous donation amounts (in rupees) are scaled to ensure all features contribute equally to the model. Categorical variables including employment sector, degree program, and marital status are encoded into numerical representations suitable for machine learning algorithms. Furthermore, the geographical location of alumni is transformed, potentially into binary features indicating specific regions or through one-hot encoding to capture more granular location information. These	2 hours

meticulous preprocessing steps are crucial for preparing the data for robust model training and ultimately enhancing the accuracy of predicting which alumni are most likely to donate.	
2. Use **linear regression** to predict a student's final academic performance based on features such as attendance percentage, internal assessment scores, assignment submissions, project grades, previous semester GPA, participation in extracurricular activities, study hours per week, socio-economic background, number of courses taken in the semester, involvement in academic clubs, time spent on campus, previous academic performance, and faculty feedback ratings.	2 hours
3. You have Computational Fluid Dynamics (CFD) simulation results for turbine blades under varying loading and flow conditions. Each simulation provides features like maximum principal stress, Von Mises stress, maximum temperature, displacement magnitude, surface pressure, fatigue life estimate, Reynolds number, and wall shear stress. Using this data, **formulate a decision tree approach** to classify whether a blade is likely to "Fail" or remain "Safe."	2 hours
4. Use the **KNN classifier** to predict the biochemical composition of a sample into enzyme concentration levels (e.g., Low, Medium, High) based on features such as pH level, enzyme activity, substrate concentration, temperature, protein concentration, absorbance at specific wavelengths, reaction time, catalyst concentration, substrate type, ion concentration, viscosity of the solution, and absorbance at 280 nm	2 hours
5. Apply **Support Vector Machines (SVM)** to predict whether a university student will stay enrolled or drop out based on features such as high school GPA, entrance exam score, extracurricular activities, letter of recommendation score, work experience, number of club activities, financial aid status, number of failed courses, mental health status, attendance percentage, family support, and work-study participation.	2 hours
6. **Implement logistic regression** to predict whether a hotel guest will cancel their booking based on features such as lead time, previous cancellations, deposit type (No Deposit, Non-Refundable, Refundable), customer type (Transient, Group, Contract, Transient-Party), number of special requests, average daily rate (ADR), reserved room type, total nights stayed, booking changes, market segment (Direct, Corporate, Online Travel Agent), assigned room type, repeated guest status, and distribution channel.	2 hours
7.	2 hours

A student at a multimedia school has created a web design project, which includes an animated logo, interactive media elements, and is designed using HTML/CSS. You need to predict whether this project will meet the design standards of a client and industry expectations based on the following features: **Features**: **Number of Pages Designed** (numeric): The number of pages in the website design. **Animation Complexity** (categorical): The complexity of the animation used (e.g., simple, moderate, complex). **Client Feedback Rating** (numeric): A rating (1-10 scale) provided by the client after previewing the project. **Interactive Features Implemented** (categorical): Whether the website includes interactive media elements like hover effects, videos, or dynamic content (yes/no). **HTML/CSS Coding Quality** (categorical): The quality of the HTML/CSS code written (e.g., high, medium, low). **Use of Multimedia Tools** (categorical): The tools used to create the multimedia content (e.g., Adobe After Effects, Blender, Final Cut Pro). **Project Deadline Met** (binary): Whether the project was completed on time (1 = yes, 0 = no). **Client Requirements Met** (binary): Whether the project met all client requirements (1 = yes, 0 = no). **Target Variable**: **Project Success** (binary): Whether the project is deemed successful (1 = success, 0 = failure) Using **Naive Bayes classification**, train a model to predict whether a multimedia project will be successful or unsuccessful based on the provided features. Evaluate the model's performance using accuracy, precision, recall, and F1-score	
8. Apply **K-Means clustering** to categorize cricket players based on their performance metrics using features such as batting average, strike rate, number of boundaries per match, total runs, bowling average, economy rate, number of wickets taken, dot ball percentage, fielding efficiency, and match impact score. The clustering can help group players into roles like power hitters, all-rounders, specialist bowlers, consistent batsmen, or fielding specialists.	2 hours
9.	2 hours

Apply DBSCAN clustering to categorize building construction projects based on features such as total construction cost, project duration, number of labor hours, material cost percentage, number of floors, safety incident count, construction method (traditional, modular, prefab), site complexity score, weather delays, and energy efficiency rating, to identify clusters of typical, high-risk, or high-efficiency construction projects.	
10. Use **Principal Component Analysis (PCA)** to analyze and reduce the dimensionality of participant and event data collected during a university techfest, based on features such as number of events participated, event types (coding, robotics, design, management), total scores earned, feedback ratings, number of workshops attended, team size, project submissions, number of prizes won, volunteer hours, and previous techfest experience.	2 hours
11. In the biochemical lab, researchers often run multi-step experiments like protein purification, DNA extraction, or enzymatic reactions. The experimental results depend heavily on parameters like temperature, pH, reaction time, reagent concentration and mixing speed Rather than manually tuning these parameters, you can use **Reinforcement Learning** to learn an optimal sequence of experimental settings that maximizes the experiment's success rate (e.g., purity of protein, yield of DNA, or enzymatic efficiency).	2 hours
12. Implement a **Random Forest classifier** to predict whether a patient is likely to have Tuberculosis (TB) based on clinical and diagnostic features. Use explainability techniques such as feature importance and individual prediction interpretation to ensure the model's decisions are understandable and medically relevant. **Features to include:** ● Chest X-ray abnormalities ● Duration of cough ● Weight loss ● Night sweats	2 hours

• Fever • Sputum test results • Previous TB history	
Total Laboratory Hours:	**30 hours**

Text Book(s)

Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar , "**Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar** ", MIT Press, 2nd Edition, **2018**
Ethem Alpaydin , "**Introduction to Machine Learning**", MIT Press Prentice Hall of India, 4th Edition, **2020**

Reference Books

Tom Mitchell, "**Machine Learning**", McGraw Hill, 3rd Edition, **1997**
Shai Shalev-Shwartz and Shai Ben-David, "**Understanding Machine Learning: From Theory to Algorithms**", Cambridge University Press, **2014**
Stephen Marsland , "**Machine Learning: An Algorithmic Perspective**", Chapman and Hall CRC press, 2nd Edition, **2015**

Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Seminar

Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE622	Data Engineering and Visualization	3	0	2	4
Pre-requisite	NIL	Syllabus Version			
		1			

Course Objectives

1. To impart foundational knowledge of data engineering concepts and the role of data engineering in modern data systems.
2. To develop skills to work with real-world data ingestion, storage, and processing tools and to prepare data for visualization and draw inferences.
3. To provide hands-on experience in visualizing and interpreting diverse data types for supporting decision-making and business intelligence.

Course Outcomes

1. Comprehend core data engineering concepts, data types, and the stages of the data engineering lifecycle.
2. Compare and implement data ingestion, storage, and processing workflows using batch and real-time frameworks.
3. Perform data cleaning, transformation, and statistical analysis to prepare datasets for downstream analytics.
4. Apply suitable data visualization techniques to represent and interpret scalar, vector, temporal, spatial, and hierarchical data.
5. Create interactive and insightful data stories that support decision-making and communicate key insights effectively.

Module:1	Foundations of Data Engineering	9 hours

Overview of Data Engineering - Data – Types of Data - Data Engineering Lifecycle: Data Capture, Data Storage, Data Processing - Batch vs. Real-time, Data Analysis and Data Visualization -Big Data and its characteristics - Real-world use cases of Data Engineering -Data Capture and Ingestion - Data Ingestion Frameworks: Batch vs. Stream Processing, Tools: APIs, Web Scraping, Kafka, NiFi, Real-time Ingestion with Apache Kafka / AWS Kinesis.

Module:2	Data Storage, Processing and Integration	10 hours

Data Storage: SQL vs NoSQL, Data Warehousing concepts, Big Data processing frameworks: Hadoop and Spark. Introduction to MongoDB JSON data format, CRUD operations: Create, Read, Update, Delete in MongoDB - Querying data using MongoDB Query Language - Data Processing Technologies: Batch- Apache Hadoop, Apache Spark – RDD – DataFrame - SQL and Stream - Apache Flink, Kafka Streams - Data Integration and ETL Pipelines: Extract, Transform, Load (ETL) processes, Data pipeline orchestration tools: Apache Airflow, Prefect.

Module:3	Data Analysis and Statistical Visualization	8 hours

Python for Analysis: NumPy, Pandas - Data Cleaning: Handling Missing Data - Data Transformation - Data Merging and Joining - Data Aggregation - Statistical

Summary - Data Distribution, Correlation and Covariance and Multivariate Analysis – Matplotlib - Customizing plots: Titles, legends, axis labels, gridlines - Subplots and multiple axes - Line plots, scatter plots, histograms, bar charts, and heat maps - Advanced visualizations: Box plots, violin plots, pair plots - Visualizing using Seaborn		
Module:4	**Visualization Methods for Complex Data**	**9 hours**
Data Abstraction - Task Abstraction - Scalar and Vector visualization techniques - Visual Variables - Map Color and Other Channels - Visual Analytics: Manipulate View - Reduce Items and Attributes - Visualization of Networks, Trees, Arrange Tables, Geospatial data and Time series data.		
Module:5	**Interactive Visual Analytics**	**7 hours**
Advanced Visualization Tool: Tableau - Creating Dashboards - Real-time Data Visualization, Interactive Visualizations and storytelling - Data Visualization for Predictive analytics - Use cases: Finance – Marketing - Insurance - Healthcare.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

Joe Reis and Matt Housley, "**Fundamentals of Data Engineering**", O'Reilly Media, 1st Edition, **2022**

Paul Crickard, "**Data Engineering with Python**", Packt Publishing, 1st Edition, **2020**

Shannon Bradshaw, Eoin Brazil, Kristina Chodorow , "**MongoDB: The Definitive Guide**", OReilly, 3rd Edition, **2019**

Tamara Munzer, "**Visualization Analysis and Design**", CRC Press, 1st Edition, **2014**

Reference Books

Sandeep Kumar Pandey, "**Data Engineering Fundamentals: A Step by Step Approach**", Notion Press, 1st Edition, **2024**

Richard J. Schiller & David Larochelle, "**Data Engineering Best Practices**", Packt , 1st Edition, **2024**

Wexler, S., Shaffer, J., Cotgreave, A, "**The Big Book of Dashboards: Visualizing Your Data Using Real-World Business Scenarios**", Wiley, 1st Edition, **2017**

Indicative Experiments

1. Explore and classify different types of data : structured, semi-structured, and unstructured types.	2 hours
2. Web Data Ingestion via Scraping: Develop a web scraping workflow using either BeautifulSoup or Scrapy to extract relevant data from websites and store it in a structured format.	2 hours

3. Data Cleaning and Transformation Using Pandas: Load a dataset with Pandas and perform comprehensive data cleaning: handle missing values, remove duplicates, and standardize formatting. Perform aggregation, grouping, and conditional filtering to derive insights.	2 hours
4. **Data Visualization and Abstraction Using Python:** Create a line chart using Matplotlib with appropriate labels, titles, and legends. Load a built-in dataset from Seaborn and generate scatterplots and pairplots to visualize feature relationships. Implement a bar plot with color-coded bars. Utilize Seaborn's visual encodings (size, shape, color) to represent data characteristics effectively.	2 hours
5. Graph and Geospatial Visualization: Construct a hierarchical tree (e.g., a family tree) using the networkx library and render it as a network graph. Additionally, use Plotly Express to map geographical data (e.g., population by country) and display structured tables using Pandas and Plotly.	2 hours
6. Build a basic ETL pipeline using Python and Apache Airflow to ingest CSV data, transform it, and load into a database.	2 hours
7. Create visualizations such as line charts (trends), bar charts (categorical comparison), pie charts (proportions), and heatmaps (correlation patterns) using Matplotlib, Seaborn, or Plotly.	2 hours
8. Explore and apply different Seaborn themes (e.g., whitegrid, darkgrid). Create advanced visualizations such as stacked bar charts and multiple subplots for comparative analysis.	2 hours
9. **MongoDB CRUD Operations and Querying:** Set up MongoDB (locally or via Atlas). Create a database StudentDB and a collection students. Insert documents with fields like name, age, department, and marks. Perform CRUD operations and run queries to retrieve, update, and delete records based on specific conditions using MongoDB Query Language (MQL). Perform the following operations: Fetch all students,Find students with marks > 80,Update a student's department based on name,Delete a student record based on age.	2 hours
10. **ETL with MongoDB -** Extract data from a CSV (e.g., employee records), Transform the data by cleaning and converting to JSON, Load into MongoDB, Perform queries such as filtering by department or salary. Find employees from a specific department. Find employees with salary greater than a threshold. (Optional) Use Apache Airflow or Prefect to orchestrate ETL steps in Python	2 hours
11. Understand predictive analytics by applying trend lines and forecasting models to real-world datasets. Tasks:Import a sales dataset (e.g., Superstore Sales).Create a line chart showing sales over time.Add a trend line and forecast future sales using Tableau's forecasting tools.Customize the model (linear,	2 hours

exponential).	
12. Create an interactive dashboard and develop a storyline with Tableau's Story feature. Tasks:Import a global COVID-19 dataset or Sales dataset.Create multiple charts (maps, bar graphs, pie charts).Combine charts into a single dashboard with:Filters,Action buttons.Build a story by sequencing sheets to narrate data insights.	2 hours
13. Build a Tableau dashboard to visualize financial KPIs (Key Performance Indicators). Tasks:Import a sample finance dataset (with Revenue, Profit, Expenses). Create:Profit and Loss report (Bar charts),Revenue vs Expense Trend (Line charts),Profit Margins (Pie or Donut chart).Create KPIs indicators with color coding (green for profit, red for loss).	2 hours
14. Design a dashboard to visualize customer segments and marketing campaign results. Tasks:Import marketing campaign dataset (customer age, income, campaign response),Use clustering to segment customers based on features. Create visualizations for:Customer profiles,Campaign success rates,Response rates by demographics,Interactive filters: Age range, Income level.	2 hours
15. Visualize healthcare data like patient vitals, admissions, and discharge rates. Tasks:Import a hospital dataset (patient id, age, disease, admission date, discharge date). Create:Admission vs Discharge Trends,Disease category breakdown,Patient readmission rates Build a dashboard with: Filters for department and date range,Highlight critical patient cases.	2 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment	
Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE623	Text and Speech Analytics	3	0	0	3
Pre-requisite	Nil	**Syllabus Version**			
		1			

Course Objectives

To develop a foundational understanding of Natural Language Processing (NLP) for text analytics.
To enable applying speech processing methods to analyze, extract, and interpret information from speech.
To build, evaluate, and Implement Language and Speech Models.

Course Outcomes

Apply foundational NLP techniques to process and analyze textual data.
Develop models for text classification, sentiment analysis, and topic modeling using appropriate NLP methods.
Extract meaningful features from speech signals by applying relevant speech processing techniques.
Build and evaluate models for speech recognition, speech synthesis and speaker identification using suitable algorithms.

Module:1	**Text Representation and Embeddings**	**8 hours**

Introduction to Natural Language Processing (NLP) - Levels of NLP - Basic Text processing- Language Modelling - Vector Space Model (VSM) - One-Hot Encoding - Word Embeddings: Word2Vec, GloVe, FastText - Contextual Embeddings - Case Study: Design word embeddings for a low-resource language

Module:2	**Text Classification and Summarization using Deep Learning**	**8 hours**

Traditional vs deep learning models for text classification– Introduction to Attention Mechanisms – Transformers – Fine-tuning pretrained models for classification tasks – Extractive vs Abstractive summarization - Transformer-based models for text summarization –Sequence Labeling for Parts-of-Speech and Named Entities.

Module:3	**NLP Applications**	**9 hours**

Machine Translation: Machine Translation using Encoder-Decoder – Information Retrieval: Information Retrieval with Dense Vectors, Information Retrieval with RAG – Information Extraction: Relations, Events, and Time – Semantic Role Labeling – Dependency Parsing: Graph-Based Dependency Parsing.

Module:4	**Basics of Speech Processing and Recognition**	**9 hours**
Phonetic representation - Short-Time Analysis of Speech - Spectrogram analysis - Feature Extraction - Automatic Speech Recognition (ASR): Acoustic Modeling - Traditional models, Transition to Deep Learning - Language Modeling - End-to-End ASR: Connectionist Temporal Classification (CTC), Attention-based and Transformer-based models - wav2vec, Whisper - Case Study: Implement CTC for Speech Recognition in Noisy Environments		
Module:5	**Speech Synthesis and Speaker Analytics**	**9 hours**
Speech Synthesis: Text-to-Speech (TTS) pipeline overview - Formant synthesis vs. Concatenative synthesis - Parametric synthesis - Neural TTS models – Variational Inference TTS - Voice conversion and style transfer - Sequence-to-sequence modelling - Attention mechanisms in TTS - Speaker Analytics: Speaker Verification, identification - Speaker embedding techniques: i-vectors, x-vectors, d-vectors - Metrics for speaker verification and identification - Case Study: Emotional states recognition using both spoken words (text) and vocal tone (speech).		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

Daniel Jurafsky, James H. Martin, "**Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**", Nil , 3rd Edition, **2025**

Denis Rothman, "**Transformers for Natural Language Processing**", Packt Publishing, **2022**

L R Rabiner and R W Schafer, "**Theory and Application of Digital Speech Processing**", Pearson, **2011**

Denis Rothman, "**RAG-Driven Generative AI**", Packt Publishing, **2024**

Reference Books

Lewis Tunstall, Leandro von Werra, Thomas Wolf, "**Natural Language Processing with Transformers**", O Reilly Media, **2022**

Dong Yu, Li Deng, "**Automatic Speech Recognition - A Deep Learning Approach**", Springer, **2015**

Beigi, Homayoon, "**Fundamentals of Speaker Recognition**", Springer, **2011**

Xu Tan, "**Neural Text-to-Speech Synthesis**", Springer, **2023**

Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Seminar, Presentation

Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE629	Deep Learning and Reinforcement Learning	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

To impart the knowledge on foundations and advancements in deep learning and reinforcement learning
To develop the skills to apply various deep learning models such as CNN, RNN and Transfer learning
To introduce reinforcement agents and models with applications in real-world domains

Course Outcomes

Discuss various deep learning architectures and applications
Implement deep learning models for vision and NLP tasks
Evaluate the deep learning model performance using metrics like accuracy, loss curves, reward functions, etc.
Explain the exploration vs. exploitation dilemma, evaluate and compare the performance, scalability, implementation complexity, and theoretical foundations.
Design and implement a real-time application using deep reinforcement learning to solve a practical problem in domains such as robotics, gaming, autonomous systems, or smart environments

Module:1	**Foundations of Deep Learning**	**8 hours**

Introduction to Deep Learning - Biological inspiration and neural networks - Multilayer Perceptrons - Backpropagation and gradient descent - Activation functions - loss functions - Overfitting - regularization - dropout - Hyper-parameter tuning
- Optimization algorithms: Gradient Descent with momentum - RMSProp - Adam.

Module:2	**Convolutional and Recurrent Networks**	**10 hours**

Convolutional Neural Networks: Filters, pooling, architecture design - LeNet, AlexNet, VGG, ResNet - Transfer Learning and Fine-tuning - Lightweight CNN architectures: EfficientNet and MobileNet - Recurrent Neural Networks - LSTM - GRU sequence modelling. Applications: Image classification, sequence prediction, time series prediction,Natural language processing

Module:3	**Advanced Deep Learning**	**8 hours**

Autoencoders and Variational Autoencoders - Generative Adversarial Networks – Transformers - Self-supervised learning-SimCLR - Continual Learning, Objection Detection:YOLO, FastRCNN		
Module:4	**Introduction to Reinforcement Learning**	**9 hours**
Basics of Reinforce Learning: Agent, Environment, Rewards - Markov Decision Processes - Dynamic Programming - Monte Carlo and Temporal Difference Learning - Multi-agent learning -Exploration vs. Exploitation -Policy Evaluation - Proximal policy optimization, Agentic AI - LLM - LangGraph.		
Module:5	**Deep Reinforcement Learning**	**8 hours**
Model-free approaches: Actor-Critic Methods - Model-based Reinforcement Learning - Dreamer and MuZero. Q-Learning and Deep Q-Networks, Policy Gradient Methods - Imitation Learning - Batch Reinforcement Learning - Data Efficient Reinforcement Learning		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**
Text Book(s)		
Ian Goodfellow, Yoshua Bengio, and Aaron Courville , "**Deep Learning**", MIT Press, 1st Edition, **2016** Richard S. Sutton and Andrew G. Barto – n, , 202, "**Reinforcement Learning: An Introduction**", MIT Press, 1st Edition, **2021**		
Reference Books		
François Chollet , "**Deep Learning with Python**", Manning, 1st Edition, **2017** Miguel Morales , "**Grokking Deep Reinforcement Learning**", Manning, 1st Edition, **2020**		
Indicative Experiments		
1. Build and train a Convolutional Neural Network (CNN): You are working as a machine learning engineer for a startup developing a digital examination platform. As part of a new feature, the platform needs to automatically recognize handwritten numeric answers provided by students through scanned answer sheets. To prove the feasibility of the system, your team wants a working prototype that can recognize individual digits (0–9) from isolated handwritten input images. Your task is to build and train a Convolutional Neural Network (CNN) that can accurately classify handwritten digits using the MNIST dataset		4 hours

2. Image Classification with Transfer Learning: You have recently joined an AI team at a pet adoption startup that allows users to upload pictures of stray animals to help find them a home. One of your first assignments is to assist in automating the classification of uploaded images into two categories: dogs and cats. The startup aims to use this classification to organize listings and speed up the adoption process. Due to limited time and computational resources, your team has decided to use transfer learning to leverage existing models trained on large image datasets. Use a pre-trained CNN (like VGG16 or MobileNetV2) to classify images of dogs and cats.	4 hours
3. Text Sentiment Classification: You have joined a media analytics company that offers sentiment analysis services to movie studios and streaming platforms. Your current task is to develop a prototype model that can classify movie reviews as either positive or negative. The company wants to demonstrate the capability to extract sentiment from unstructured customer reviews on platforms like IMDb and Rotten Tomatoes. Due to the sequential and contextual nature of text, your manager recommends using a Recurrent Neural Network (RNN), specifically a Long Short-Term Memory (LSTM) model, which is well-suited for text classification. Implement a sequential model to classify movie reviews as positive or negative using an LSTM-based RNN.	4 hours
4. Implement Generative Adversarial Networks to generate realistic Images. Use MNIST, Fashion MNIST or any human face datasets.	4 hours
5. Autoencoder for Image Denoising: You are working with a digital archive team in a historical research institute. They've scanned thousands of old handwritten documents, but many images suffer from noise due to aging, poor scanning quality, and dust. Manually cleaning them is impractical. You've been asked to create a deep learning model that can automatically denoise grayscale images while preserving the original content. To simulate and develop a prototype of this solution, you use the MNIST dataset and build an auto-encoder that learns to reconstruct clean digit images from noisy inputs.	2 hours
6. Implement an approximate q-learning agent that learns weights for features of states, where many states might share the same features. Write your implementation in Approximate Agent class	2 hours

7. Implement Q-learning to solve discrete state and action problems, such as maze navigation or simple games	2 hours
8. Implement actor-critic methods, which involve separate networks for policy learning (actor) and value function approximation (critic)	2 hours
9. Design a stimulation of self-driving taxis. It makes use of the Q-Learning algorithm. Sounds fun and exciting - no? You will be given a grid with four locations marked on it. A passenger is waiting for a taxi at one of these locations and wants to be dropped at another location. Like the taxi driver, it is your responsibility to make sure: Pick up the passenger from the right location Drop the passenger at the right location Take the shortest path possible.	2 hours
10. Implement the Trading Bot using Agentic AI for any trading, define actions like buy, sell, hold, etc, and positions (long/short) for the trading environment. Here is the Trading Bot using Agentic AI to the official GitHub repository for understanding more about the trading environment configuration and training process. To test or play around with these environments, you can download the data from MarketWatch.	4 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Seminar	
Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE630	Generative AI and Large Language Models	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

Introduce core concepts of generative AI, including statistical and neural language models, and their use in text generation, translation, and summarization.
Build skills to design, implement, and evaluate generative models like RNNs, LSTMs, VAEs, GANs, and transformers for sequence and language tasks.
Develop proficiency in prompt engineering, model deployment, and handling challenges related to retrieval systems, ethics, and deployment platforms.

Course Outcomes

Analyze the transition from statistical to neural language models and apply word embedding techniques in core NLP applications such as text generation and translation.
Design and implement deep sequence and generative models to address tasks in sequence modelling.
Apply transformer-based architectures and decoding strategies to build and evaluate large language models for various NLP tasks.
Formulate effective prompt engineering strategies and integrate vector databases in Retrieval-Augmented Generation (RAG) systems.
Deploy and manage large language models across cloud and on-premise environments while addressing ethical, legal, and technical deployment challenges.

Module:1	**Foundations of Generative AI and Language Models**	**7 hours**

Introduction to Generative AI – Statistical Models: N-Gram, HMM, and their limitations – Neural Language Models: Feedforward Neural Language Model architecture – Word Embeddings: Word2Vec, GloVe, FastText – Contextual Embeddings – Applications in text generation, translation, summarization, and chatbots.

Module:2	**Deep Sequence and Generative AI Models**	**9 hours**

Introduction to Sequence Models – Recurrent Neural Network (RNN) – Long Short Term Memory (LSTM) – Gated Recurrent Units (GRU) - Generative Models: Variational Audoencoders (VAE) - Generative Adversarial Networks (GAN) – CycleGAN – StyleGAN.

Module:3	Core Architectures of LLMs	9 hours
Introduction to Transformers – Encoder-decoder architecture – Self-attention mechanism – Text Generation: Greedy Search Decoding, Beam Search Decoding, Sampling methods (Top-k, Nucleus Sampling) – Models: ElMo, BERT, GPT, T5 - Evaluating model performance: BLEU, ROUGE - Multimodal transformers.		
Module:4	**Prompt Engineering**	**9 hours**
Fundamental of prompt engineering- Prompt Elements - Techniques: Zero-shot, one-shot, few-shot and Chain-of-thought (CoT) – Automatic Prompt Engineer (APE) – Graph Prompt - Vector Databases - FAISS - Pinecone – Parameter Efficient Fine Tuning - Prompting in Retrieval Augmented Generation (RAG) – Pipeline – Core Components - Forward-Looking Active Retrieval-Augmented Generation (FLARE).		
Module:5	**Deployment Models**	**9 hours**
Model Optimization for Deployment: Quantization Techniques - Cloud Hosting – OpenAI – Hugging Face – Vertex AI – FastAPI – On-Premises Hosting Requirements and Challenges – DeepLake – vLLM – Ollama – Hybrid LLM Deployment – Workflow – Benefits – Code generation and debugging with models like Codex - Ethical and Legal Considerations - Bias in models and data privacy - Copyright issues in generated content - Deployment challenges and regulation of AI models.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

Tanmoy Chakraborty, "**Introduction to Large Language Models – Generative AI for Text**", Wiley, 1st Edition, **2024**

Thimira Amaratunga, "**Understanding Large Language Models: Learning Their Underlying Concepts and Technologies**", Apress, 1st Edition, **2023**

Reference Books

Valentiana Alto, "**Modern Generative AI with ChatGPT and OpenAI Models: Leverage the capabilities of OpenAI's LLM for productivity and innovation with GPT3 and GPT4**", Packt Publishing, 1st Edition, **2023**

Omar Sanseviero, Pedro Cuenca, Apolinario Passos, Jonathan Whitaker, "**Hands-On Generative AI with Transformers and Diffusion Models**", O Reilly Media, 1st Edition, **2024**

Indicative Experiments

1. Develop a predictive text system using N-Gram and HMM models to predict the next word based on previous context. Dataset: Brown Corpus or Wikipedia dataset.	3 hours
2. Use Word2Vec, GloVe, or FastText embeddings to find similarities between product descriptions. Dataset: Amazon product dataset.	3 hours
3. Train an RNN to predict the next word in a product description based on the previous context. Dataset: Amazon product descriptions dataset.	3 hours
4. Generate new text from product descriptions using a Variational Autoencoder (VAE). Dataset: Amazon product descriptions dataset.	3 hours
5. Build a machine translation system using transformers to translate English to French. Dataset: IWSLT dataset (English-to-French).	3 hours
6. Fine-tune a pre-trained BERT model for sentiment classification of product reviews. Dataset: IMDb movie reviews dataset.	3 hours
7. Use GPT-3 for generating responses with zero-shot and few-shot learning. Dataset: SQuAD dataset.	3 hours
8. Implement FAISS for retrieving similar products based on descriptions. Dataset: MovieLens dataset.	3 hours
9. Deploy a T5 model as an API for text summarization using FastAPI. Dataset: CNN/Daily Mail dataset.	3 hours
10. Deploy a language model on Hugging Face or Vertex AI for customer support queries. Dataset: Customer service dataset.	3 hours
Total Laboratory Hours:	**30 hours**
Text Book(s)	
Tanmoy Chakraborty, "**Introduction to Large Language Models – Generative AI for Text**", Wiley, 1st Edition, **2024** Thimira Amaratunga, "**Understanding Large Language Models: Learning Their Underlying Concepts and Technologies**", Apress, 1st Edition, **2023**	
Reference Books	

Valentiana Alto, "**Modern Generative AI with ChatGPT and OpenAI Models: Leverage the capabilities of OpenAI's LLM for productivity and innovation with GPT3 and GPT4**", Packt Publishing, 1st Edition, **2023** Omar Sanseviero, Pedro Cuenca, Apolinario Passos, Jonathan Whitaker, "**Hands-On Generative AI with Transformers and Diffusion Models**", O Reilly Media, 1st Edition, **2024**	
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Oral Examination, Seminar	
Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE631	MLOps	3	0	2	4
Pre-requisite	NIL	Syllabus Version			
		1			

Course Objectives

1. To enable building production-ready, scalable, and responsible ML systems by mastering MLOps principles like lifecycle management, versioning, deployment, and monitoring.
2. To develop the ability to design and automate end-to-end ML pipelines, including data/model versioning, orchestration, NLP preprocessing, and deployment across cloud and edge, utilizing modern tools
3. To develop the skills to precisely explore and implement advanced AI concepts and generative models with a focused consideration of ethical and governance aspects.

Course Outcomes

1. Develop and automate ML pipelines for testing, developing, and deploying models.
2. Leverage tools such as Docker, Kubernetes, MLflow, and Kubeflow for large-scale model management.
3. Design responsible AI systems with reproducibility, reliability, and ethical deployment.
4. Assess model performance in production and address issues such as data drift and retraining of models
5. Create and configure CI/CD pipelines for machine learning solutions.

Module:1	Foundations of MLOps	6 hours

MLOps Evolution - DevOps vs. MLOps - ML Production Challenges - ML Lifecycle Management - Key Tools and Frameworks Overview - Cloud Computing Tools: AWS ML Tools, Google Cloud AI, IBM Watson - Representation Learning: The Core of Modern AI.

Module:2	Versioning, Pipelines, and Preprocessing	10 hours

Data Versioning with DVC - Model Versioning using MLflow - Pipeline Orchestration using Kubeflow - Automated Data Preprocessing and Validation - Experiment Tracking - Data Preprocessing for Language Models: Text Vectorization Layer - Standardization, Vocabulary Indexing - Embedding Word Vectors - TF-IDF - Tokenization Fundamentals: Byte Pair Encoding - Bag of Words Model and Sequential Models.

Module:3	Deployment Strategies and Transformers	10 hours

Deployment Strategies: Batch, Online, Streaming - Serving Models with FastAPI and Flask - Dockerizing ML Models - Kubernetes Basics for ML Deployment - Serverless Deployment - AWS Lambda, Google Cloud Functions - Transformer Encoder for Language Comprehension - BERT Models - Decoder-only GPT Models for Language Generation - Generative Adversarial Networks - Diffusion

Models for Image Generation - Autoencoders for Representation Learning - Pre-Training - Reinforcement Learning through Human Feedbac**k.**		
Module:4	**CI/CD and LLMOps**	**10 hours**
Continuous Integration and Continuous Delivery/Deployment (CI/CD) for ML Systems - Feature Stores - Testing ML Pipelines -Unit, Integration, Regression - Responsible and Ethical MLOps - Real-World Case Studies -Retail, Finance, Healthcare -LLMOps: Tools and Platforms - LangChain, OpenAI API - Prompt Engineering, Retrieval-Augmented Generation (RAG) - Low-Rank Adaptation (LoRA) and its variants - Guardrails and Processing of LLM Outputs.		
Module:5	**Monitoring and Governance**	**7 hours**
Model Monitoring: Accuracy, Drift, Bias - Logging and Observability Tools - Data Drift Detection - Model Retraining Pipelines - Governance and Compliance in ML Systems.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

Mark Treveil & Alok Shukla , "**Introducing MLOps: How to Scale Machine Learning Projects with Continuous Delivery, Monitoring, and Automation**", O'Reilly, 1st Edition, **2020**

Burkov, A, "**Machine learning engineering**", Montreal, QC, Canada, 1st Edition, **2020**

Reference Books

Noah Gift & Alfredo Deza , "**Practical MLOps: Operationalizing Machine Learning Models**", O'Reilly, 1st Edition, **2021**

Hannes Hapke & Catherine Nelson , "**Building Machine Learning Pipelines: Automating Model Life Cycles with TensorFlow**", O'Reilly, 1st Edition, **2020**

Chris Fregly & Antje Barth , "**Data Science on AWS: Building Scalable and End-to-End Machine Learning Workflows on AWS**", O'Reilly, 1st Edition, **2021**

Haviv, Y., & Gift, N, "**Implementing MLOps in the enterprise**", O'Reilly Media, Inc, 1st Edition, **2023**

Indicative Experiments

1. MLOps Workflow for Hospital Emergency Prediction using ETS Data on AWS The objective of this experiment is to develop and implement a Machine Learning Operations (MLOps) framework for predicting emergent medical situations at the point of patient admission to a hospital. This predictive model will leverage historical patient data, particularly the Emergency Triage Score (ETS) dataset which can be collected from https://www.kaggle.com/datasets/maalona/hospital-triage-and-patient-history-data , to probabilistically anticipate instances where a significant influx of patients is likely to occur. This experiment will harness data-	3 hours

driven insights to enhance the preparatory capacities of healthcare facilities in responding to emergent medical scenarios, thereby improving overall healthcare service efficiency and patient care outcomes. The goal of this experiment is to develop a Machine Learning Operations (MLOps) workflow to predict hospital emergencies at the time of patient admission. This predictive model will help healthcare providers allocate resources efficiently and prepare for potential surges in patient volume.	
2. Financial Fraud Detection using Deep Learning and MLOps This experiment aims to develop a robust solution for detecting financial fraud using MLOps methodologies. It involves analyzing historical transaction data, customer behavior, and relevant factors to identify and prevent fraudulent activities in the financial domain. The financial sector faces a growing threat from financial fraud, driven by increasingly sophisticated tactics used by fraudsters. This experiment addresses the need for advanced fraud detection solutions capable of adapting to emerging threats in real-time using the Paysim synthetic dataset collected from https://www.kaggle.com/datasets/ealaxi/paysim1.	3 hours
3. Customer Conversational Intelligence Platform A Customer Conversational Intelligence Platform is a system that employs advanced technologies, including machine learning and natural language processing, to analyze and make sense of customer interactions across various communication channels such as chatbots, call centers, emails, and social media, that modern businesses accumulate. The experiment will develop a platform that harnesses the power of machine learning to analyze vast amounts of customer interaction data. The aim is to derive actionable insights from these interactions, optimize customer service processes, and enhance overall customer experience. The dataset such as 3K Conversations Dataset for ChatBot, Customer Support on Twitter Dataset can be collected from https://www.kaggle.com/datasets used for developing the Customer Conversational Intelligence Platform.	3 hours
4. Driver Demand Prediction for Optimal Food Delivery Charges This experiment aims to predict the demand for delivery drivers in specific regions and times, leveraging MLOps methodologies. By analyzing order requests, driver activity, and related parameters, the goal is to optimize delivery charges, ensuring consistency and minimizing customer drop-offs. The unpredictable nature of delivery charges, primarily due to driver unavailability, often results in increased prices and subsequent customer dissatisfaction. The experiment seeks to bridge this gap by forecasting driver demand, thereby streamlining delivery pricing. The Food Delivery Time Prediction dataset collected from https://www.kaggle.com/datasets/bhanupratapbiswas/food-delivery-time-prediction-case-study can be used for predicting the demand for delivery drivers.	3 hours

5. Personalized Financial Advisor using Large Language Model (LLM) The field of finance can be complex and overwhelming for individuals seeking personalized financial advice. In order to make informed decisions regarding investments, retirement planning, budgeting, and financial products, individuals often require guidance from financial experts. The aim of this experiment is to develop an Intelligent Financial Advisor powered by a Large Language Model (LLM) to provide personalized financial advice and guidance to individuals. By leveraging NLP and machine learning techniques, the Intelligent Financial Advisor will assist users in making informed financial decisions and achieving their financial goals.	3 hours
6. Automated Search Engine Optimization (SEO) tool The goal of this experiment is to create an automated Search Engine Optimization (SEO) tool an AI-based chatbot system. The tool will use natural language processing (NLP) and machine learning (ML) algorithms to analyze website content, identify SEO issues, and provide recommendations for improvement. The tool will help website owners and SEO professionals to optimize their website's content and improve search engine rankings more efficiently and effectively.	3 hours
7. Automated Answer Validation for Science Question Answering using Siamese Text Similarity Model The goal of this experiment is to develop an automated answer validation system using a Siamese text similarity model. The system will compare student responses with the correct answer and distractors to determine the level of correctness and provide appropriate feedback. The automated answer validation system for science question answering will benefit educators and students in science-related subjects. It will streamline the assessment process, reduce manual effort, and ensure consistent evaluations, leading to improved learning outcomes in the science domain. The SciQ (Scientific Question Answering) dataset can be collected from https://www.kaggle.com/datasets/thedevastator/sciq-a-dataset-for-science-question-answering	3 hours
8. GAN based interior designs The goal of this experiment is to generate realistic new interior room designs by training a GAN network on the IKEA Interior Design Dataset collected from https://www.kaggle.com/datasets/ahmedkallam/ikea-sa-furniture-web-scraping.	3 hours
9. Image-based plant disease identification Crop losses due to diseases are a major threat to food security every year, across countries. Conventionally, plant diseases were detected through a visual examination of the affected plants by plant pathology experts. This	3 hours

was often possible only after major damage had already occurred, so treatments were of limited or no use. Recently, access to smartphone based image capturing has highly increased amongst farmers and agriculturists. This has led to the successful adoption of plant disease diagnostic applications based on deep learning techniques. This is of immense value in the field of agriculture and an excellent tool for faster identification and treatment of crop diseases. It holds key importance in preventing crop based food and economic losses. The goal of this experiment is to build a convolutional neural network or to use transfer learning and develop a plant disease identification tool using Crop Disease Image Dataset collected from https://www.kaggle.com/datasets/jawadali1045/20k-multi-class-crop-disease-images.	
10. Food Image Segmentation Food image segmentation is a critical and indispensable task for developing health-related applications such as automated estimation of food calories and nutrients as a means for dietary monitoring. One of the challenges in this area is the improvement of accuracy in dietary assessment by food image analysis. However, how to derive the food information (e.g., food type and portion size) from food images effectively is a challenging task and an open research problem. In this experiment, participants are expected to make a model that can segment the food components present in an input food image and build an application that can predict the food class and the food portions from it. The Food Segmentation Object Detection dataset collected from https://www.kaggle.com/datasets/trainingdatapro/food-segmentation used for implementing the experiment.	3 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Oral Examination, Presentaion	
Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE632	Multi-Task and Meta Learning	3	0	0	3
Pre-requisite	NIL	Syllabus Version			
		1			

Course Objectives

1. To introduce the theoretical foundations and motivations behind Multi-Task Learning (MTL) and Meta Learning
2. To impart knowledge to explore, analyze and implement state-of-the-art techniques in MTL and Meta Learning,
3. To develop skills to apply advanced meta-learning techniques to solve real-world problems

Course Outcomes

1. Analyze and differentiate between MTL, meta learning, and transfer learning, including task types and common pitfalls such as task interference and negative transfer.
2. Design and implement deep neural networks for multi-task learning using shared and task-specific layers, along with gradient conflict resolution strategies.
3. Apply various meta-learning paradigms to few-shot learning problems using benchmark datasets.
4. Evaluate advanced meta-learning models across domains, analyze their scalability, and reflect on ethical considerations in deploying them to real-world applications such as healthcare and robotics.

Module:1	Foundations of Multi Task and Meta Learning	6 hours

Introduction to Multi-Task Learning and Meta Learning - Motivation and differences between Multi-Task Learning, Transfer Learning, and Meta Learning - Types of tasks: homogeneous vs. heterogeneous - Key challenges: task interference, negative transfer, generalization - Popular datasets and benchmarks: Omniglot, Mini-ImageNet

Module:2	Deep Multi Task Learning Techniques	10 hours

Hard parameter sharing vs. soft parameter sharing - Task-specific layers and shared backbone designs - Task balancing and loss weighting - uncertainty weighting - Gradient-based techniques for conflict resolution: GradNorm, PCGrad – Applications: NLP: GLUE - Vision :NYUv2

Module:3	Meta Learning Algorithms	10 hours

Introduction to few-shot learning and fast adaptation - Optimization-based methods: Black Box Meta Learning, Model Agnostic Meta Learning, Reptile - Metric-based methods: Matching Networks, Prototypical Networks - Model-based methods: Memory-augmented networks, RNN-based learners - Episodic training and meta-objective formulation

Module:4	Advanced Topics in Meta Learning	10 hours

Meta-regularization and meta-optimization strategies - Multi-task meta-learning and task conditioning - Bayesian meta-learning and uncertainty estimation - Cross-domain and multi-modal meta-learning - Meta-learning in reinforcement learning environments: PEARL, RL^2		
Module:5	**Applications and Tools**	**7 hours**
Use cases in healthcare, Multimedia, robotics, recommendation systems, autonomous driving - Benchmarking tools and standardized evaluation: Meta-Dataset - Ethics, Overfitting in meta-learning, scalability issues		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

Gerardus Blokdyk, "**Multi-Task Learning: A Complete Guide**", Kindle Edition 5STARCooks, **2020**

Lan Zou, "**Meta-Learning: Theory, Algorithms and Applications**", Academic Press London United Kingdom, 1st Edition, **2023**

Reference Books

Wenwu Zhu and Xin Wang, "**Automated Machine Learning and Meta-Learning for Multimedia**", Springer Nature Cham Switzerland, 1st Edition, **2022**

George Drosatos, Pavlos S. Efraimidis, and Avi Arampatzis, "**Federated and Transfer Learning Applications**", MDPI Basel Switzerland, 1st Edition, **2022**

Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Seminar, Presentaion

Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE633	Graph Database and Analytics	3	0	0	3
Pre-requisite	NIL	Syllabus Version			
		1			

Course Objectives

1. To impart graph concepts and model real-world problems using graph databases.
2. To provide knowledge on graph analytics techniques, including traversal, centrality, and link prediction, for extracting insights from graph-structured data.
3. To articulate graph features and node embeddings into machine learning workflows.

Course Outcomes

1. Comprehend the graph data modelling using modern graph database technologies.
2. Analyze the graph traversal and search algorithms to build and query graph databases.
3. Apply centrality measures, community detection algorithms and link prediction for graph analytics.
4. Develop machine learning models that incorporate link graph features and node embeddings.

Module:1	Graph Databases and Graph Data Modelling	7 hours

Introduction to graph database -Fundamental concepts of Graph- Graph Database- Comparison with other types of databases - Graph data modeling: The data modeling process: Understanding the problem- Developing the white board model- Constructing the logical data model

Module:2	Graph Traversals and Building on graph database	9 hours

Pathfinding traversals and mutating graphs - Mutating a graph – Paths - Traversing and filtering edges - Formatting results – Application development

Module:3	Graph Search Algorithms	9 hours

Breadth First search-Depth first search-Shortest path-All pair shortest path-Single source shortest path-Minimum Spanning tree-Random walk- Centrality Algorithms - PageRank- Community Detection Algorithms

Module:4	Machine Learning on Graphs	9 hours

Machine Learning and the Importance of Context- Connected Feature Extraction and Selection: Graph Features -Graph Algorithm Features - Link Prediction – Multi-relational Graphs - Node Classification – Clustering - Graph Classification - Graph Statistics and Kernels Methods- Neighborhood overlap detection.

Module:5	**Node Embeddings and Graph Neural Networks**	**9 hours**
Node Embeddings - Neighborhood Reconstruction Methods - Optimizing an Encoder-Decoder Model - Multi-relational Data and Knowledge Graphs. Introduction to Graph Neural Network Model - Graph Neural Networks in Practice.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

Perryman, J., & Bechberger, D. , "**Graph Databases in Action: Examples in Gremlin.** ", Simon and Schuster Manning Publications, 1st Edition, **2020**

Needham, M., & Hodler, A. E. , "**Graph algorithms: practical examples in Apache Spark and Neo4j.** ", O Reilly Media, 1st Edition, **2019**

William L. Hamilton. , "**Graph Representation Learning**", Morgan and Claypool McGraw Hill, 1st Edition, **2020**

Reference Books

Bratanic, T. , "**Graph Algorithms for Data Science: With Examples in Neo4j** ", Simon and Schuster Manning Publishers , 1st Edition, **2024**

Scifo, E. , "**Hands-On Graph Analytics with Neo4j: Perform graph processing and visualization techniques using connected data across your enterprise.**", Packt Publishing Ltd, 1st Edition, **2020**

Robinson, I., Webber, J., & Eifrem, E. , "**Graph databases: new opportunities for connected data.** ", O Reilly Media Inc, 1st Edition, **2015**

Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Seminar

Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No.	12-06-2025

Course Code	Course Title	L	T	P	C
MACSE634	Responsible Artificial Intelligence	3	0	0	3
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

To impart the ethical, legal, and social implications of AI technologies across various domains.
To introduce AI systems bias, fairness, transparency, privacy, and accountability concerns using real-world case studies.
To develop skills to evaluate and apply responsible AI frameworks, tools, and governance models in designing and deploying ethical AI solutions.

Course Outcomes

Detect ethical risks in the lifecycle of AI systems and mitigate bias in datasets and algorithms using fairness-aware techniques.
Apply explainability and transparency tools to improve interpretability of AI models.
Assess privacy and security vulnerabilities in AI systems and implement appropriate defense mechanisms.
Demonstrate awareness of AI governance frameworks, policies, and the global outlook on responsible AI development.

Module:1	**Foundations of Responsible AI**	**9 hours**

Introduction to AI Ethics - Importance-Ethical theories-Core principles: Fairness, Accountability, Transparency, Privacy, Safety-The lifecycle of AI systems and ethical touchpoints-Stakeholders and their roles-Case Studies: Early AI mishaps

Module:2	**Bias Fairness and Inclusion in AI**	**8 hours**

Sources of bias in data and algorithms-Types of fairness: Demographic parity, Equal opportunity, Equalized odds-Detecting and mitigating bias-Methods: Re-sampling, Re-weighting, Fair representation learning-Trade-offs between fairness and accuracy, Case Studies.

Module:3	**Transparency Explainability and Accountability**	**9 hours**

Importance of explainable AI-Approaches: LIME, SHAP, counterfactual explanations-Transparency vs interpretability-Accountability-Mechanisms for documenting models: Datasheets, Model Cards, FactSheets-Real-world auditing practices: Internal vs external audits

AI policies and regulations: GDPR, IDPR, EU AI Act, U.S. Algorithmic Accountability Act- Global perspectives: China, EU, U.S., UNESCO guidelines- Governance tools: Ethics boards, regulatory sandboxes, independent audits- Responsible innovation and AI for social good initiatives- Future challenges: Generative AI, synthetic media, autonomous decision-making.		
Module:4	**Privacy Security and Safety in AI Systems**	**9 hours**
Data privacy: Concepts of anonymity, pseudonymity-Privacy-preserving techniques: Differential privacy, k-anonymity, Federated learning-AI security threats: Adversarial attacks, data poisoning-Defense strategies: Adversarial training, robust optimization- AI in safety-critical systems: healthcare, autonomous vehicles.		
Module:5	**Governance Policy and the Future of Responsible AI**	**8 hours**
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

Silja Voeneky, Philipp Kellmeyer, Oliver Mueller, Wolfram Burgard , "**The Cambridge Handbook of Responsible Artificial Intelligence: Interdisciplinary Perspectives**", Cambridge University Press, **2022**

Lu, Q., Zhu, L., Whittle, J., & Xu, X, "**Responsible AI: Best practices for creating trustworthy AI systems**", Addison Wesley Professional, **2023**

Reference Books

Virginia Dignum, "**Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way**", Springer Nature, **2019**

Christoph Molnar, "**Interpretable Machine Learning**", Lulu, 1st Edition, **2019**

Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Final Assessment Test

Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025