AC-01-Item No. 2.0, 2.1, 3.0 & 3.1 dated 29th April 2024

Agnel Charities

Fr. C. Rodrigues Institute of Technology

Sector 9A, Vashi, Navi Mumbai, 400703, Maharashtra, India

www.fcrit.ac.in

An Autonomous Institute Affiliated to the University of Mumbai

Department of Computer Engineering

Curriculum Structure –

Multidisciplinary Minor Courses offered by

Department of Computer Engineering

Prepared by: Board of Studies for Computer Engineering

Approved By: Academic Council of Fr. C. Rodrigues Institute of Technology

Effective from :2025-26

Revision: 2024.1

Contents

Multidisciplinary Minor Courses Offered by the Department for the Other Program Students

Curriculum Structure for MDM Courses

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		L	P	T	L	P	T	Total
CEMDM301	Data Structures and Algorithms	3	–	–	3	–	–	3
CEMDM402	Database Management Systems	3	–	–	3	–	–	3
CEMDM503	Cloud computing	3	–	–	3	–	–	3
CEMDM604	Soft Computing	4	–	–	4	–	–	4
CEMDL601	Machine Learning Laboratory	–	2	–	–	1	–	1
Total		**13**	**2**	**–**	**13**	**1**	**–**	**14**

Examination Scheme for MDM Courses

Course Code	Course Name	Examination Scheme					Total
		In-Semester Assessment$		End Sem Exam (ESE)	Exam Durationfor Theory (in Hrs)		
		Continuous Assessment	Mid-Sem Exam		Mid-Sem	End-Sem	
CEMDM301	Data Structures and Algorithms	20	30	50	1.5	2	100
CEMDM402	Database Management Systems	20	30	50	1.5	2	100
CEMDM503	Cloud computing	20	30	50	1.5	2	100
CEMDM604	Soft Computing	20	30	50	1.5	2	100
CEMDL601	Machine Learning Laboratory	25	–	25	–	–	50
Total		**105**	**120**	**225**	–	–	**450**

Syllabi of Multidisciplinary Minor(MDM) Courses

Course Type	Course Code	Course Name	Credits
MDM	CEMDM301	DATA STRUCTURES AND ALGORITHMS	03

Examination Scheme					
Distribution of Marks			Exam Duration (Hrs.)		Total Marks
In-semester Assessment		End Semester Exam (ESE)			
Continuous Assessment	Mid-Semester Exam (MSE)		MSE	ESE	
20	30	50	1.5	2	100

Pre-requisite :

1. ESCLC103- C Programming Laboratory
2. BSC101- Engineering Mathematics

Program Outcomes addressed:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design/Development of Solutions
4. PO5: Modern Tool Usage
5. PO11: Lifelong Learning

Course Objectives:

1. Acquaint students with fundamental data structures like arrays, linked lists, and trees, graphs.
2. Familiarize students with algorithms and their complexities using asymptotic notations.
3. Introduce analytical and critical thinking skills for real-world problem-solving with modern tools.
4. Educate students on sorting and searching algorithms for implementation and analysis
5. Equip students to learn to develop well-structured technical reports and presentations.

Module	Details	Hrs
	Course Introduction The Data Structures and Algorithms (DSA) course offers a fundamental understanding of efficiently organizing and manipulating data. Learners will develop the analytical skills to choose the appropriate data structure for various applications. The course includes analysis of algorithms, code optimization, and tackling challenges in complex computations. Its goal is to develop skills for software development and problem-solving. Its aim is to build abilities for both software development and problem-solving. Furthermore, this subject serves as the prerequisite for disciplines like Database Management Systems (DBMS) and the Data Science stream, laying a solid foundation for future specialization.	**01**

01.	**Introduction to Data structures and Algorithms** ***Learning Objective/s:*** *To familiarize with the fundamental principles of data structures and algorithms, recognizing their critical role in implementation, distinguishing between different types of data structures, and understanding the various asymptotic notations employed in analyzing algorithms.*	**5**
	Contents: Data Structures – Basic Terminology, Importance. Types of Data structures, Operations on Data Structures. Abstract Data type (ADT), Advantages of Data structures. Algorithms and Analysis: Performance measures of algorithms, efficiency of algorithms, complexities- Time and Space complexity , importance of algorithm analysis. Asymptotic notations - Big O, Omega, Theta. Graphical and mathematical representations of Asymptotic Analysis.	
	Self-Learning Topics: *Amortized analysis for asymptotic analysis.* ***Learning Outcomes :*** *A learner will be able to* *LO 1.1: Apply techniques to represent data in appropriate data structures. (P.I.-1.1.1)* *LO 1.2: Apply the concepts of asymptotic analysis to compute and represent time and space complexities. (P.I.-1.4.1)* *LO 1.3: Compare the performance of data structures and algorithms used in real-life applications to select the most suitable solutions (P.I.-2.2.4).* *LO 1.4: Analyze and identify sources of errors and limitations in data structure and algorithm solutions. (PI 2.4.3).*	
02.	**Stack & Queue Data Structure** ***Learning Objective/s:*** *To facilitate understanding, implement, and compare Stack and Queue structures via linked lists and arrays, explore diverse queue types and applications, and apply them in real-world problem-solving.*	**9**
	Contents:	
	Introduction to Stack, definition of stack, array-based and linked list-based stack implementation, operations on stack. Applications of Stacks: Decimal to binary conversion, Algorithm to test a string for palindrome, solving a maze, backtracking, polish expressions- Prefix, Infix, Postfix expression and conversions, evaluations of polish expressions. Recursion: Common loops, Factorials, Fibonacci series, Towers of Hanoi. Introduction to Queue , definition, array-based and linked list-based queue implementation, operations on queue. Types of Queue-Circular Queue, Priority Queue, Double Ended Queue. Implementation of multiple stack Queues, implementing a Deque with a circular array. Applications of Queue: Task scheduling examples, airline ticket counter.	
	Self-Learning Topics: ***Applications of Queues in network packet scheduling***	

	Learning Outcomes : *A learner will be able to* *LO 2.1: Apply mathematical techniques to devise algorithms for stack and queue applications, such as decimal to binary conversion and infix expression evaluation. (P.I.-1.1.1)* *LO 2.2: Describe the operations, properties, and constraints of stack and queue data structures and analyze their applications to solve practical problems. (P.I.-2.1.1)* *LO 2.3: Analyze various types of queues, including circular queues and priority queues and their applications such as task scheduling and breadth-first search.(P.I.-2.2.4)* *LO 2.4: Use technical literature to explore and apply stack and queue data structures in real-world scenarios. (P.I.-11.3.1)*	
03.	**Link list Data structure and Hashing Functions** ***Learning Objective/s:*** *To demonstrate differentiation between linked lists, execute operations on them along with hashing structures, and apply this knowledge to implement stack and queue data structures.*	**9**
	Contents: Linked Lists- Basic Terminologies, Linked Lists versus Arrays, Memory Allocation and De-allocation for a Linked List .Types of linked lists- Singly, Doubly, Circular, Circular Doubly Linked List. Operations on link lists: Node Creation, Node Insertion and Deletion from Beginning, End and Specified Position. To implement of a stack and a queue using linked list. Applications of linked lists: Polynomial Representation using link list, polynomial arithmetic- Addition, Subtraction, Multiplication, Doubly Linked List Application. Introduction to Hashing and functions: Hashing-Concept, Hash Functions, Common hashing functions, Collision Resolution Techniques (CRT): Separate Chaining for Collision Handling, Open Addressing for Collision Handling. Link list use in implementation of anti-collision of hashing.	
	Self-Learning Topics: *Hashing quadratic probing.* ***Learning Outcomes :***	
	A learner will be able to *LO 3.1: Compare different types of linked lists and operations to select the best type for given applications.P.I.-2.2.4)* *LO 3.2: Apply linked list data structure to construct stack and queue and hashing data structures. (P.I.- 3.2.1)* *LO 3.3: Develop a linked list algorithm to provide optimal solutions to mathematical problems (P.I.-3.3.1).* *LO 3.4: Learn to source and understand recent technical literature on hash-based data structures (P.I.-11.3.1).*	
04.	**Trees & Graph data structures** ***Learning Objective/s:***	**7**

	To acquire understanding of tree terminologies, proficiently perform fundamental Binary Search tree (BST) operations, and explore Adelson-Velsky and Landis (AVL) trees for efficient balance maintenance.	
	Contents: **Tree Data structure :**Introduction, Tree Terminologies, Tree Height, Level and Depth, Representation of binary tree. Types of Binary Tree, Binary Tree Traversals (In-order, Post-order, Preorder). Operations on Binary Search Tree: Insertion, Deletion. Conversion of General Trees to Binary Trees. AVL Tree, height and weight balancing algorithm. Applications of Binary Tree- Expression Tree, Huffman Encoding, **Graph Data structure:** Introduction, Graph Terminologies, Matrix Representation of Graphs, Concept of self-loop, Operations on Graph, Graph Traversals: Breadth First Search, Depth First Search, Applications of the Graph: Shortest Path Algorithms: Dijkstra's Algorithm, Minimum Spanning Tree: Kruskal and Prims Algorithm.	
	Self-Learning Topics: *Red-Black trees and Graph Partitioning.* ***Learning Outcomes :*** *A learner will be able to* *LO 4.1: Compare ,analyze and design algorithms for BST operations (insertion, deletion, conversion) to ensure optimal time complexity and memory usage.(P.I.-2.2.4)* *LO 4.2: Design and implement BFS and DFS algorithms for solving various problems. (P.I.-3.2.1)* *LO 4.3: Utilize modern hardware and software tools to effectively apply graph algorithm in solving real-world problems. (P.I.-5.1.1)* *LO 4.4: Utilize technical literature and credible sources to enhance your understanding of advanced tree and graph concepts. (P.I.-11.3.1)*	
05.	**Searching & Sorting algorithms** ***Learning Objective/s:*** *To implement, and analyze various searching and sorting methods to enable competent application in real-world scenarios.*	**7**
	Contents:	
	Searching algorithms: Sequential, Index Sequential, Binary search, comparison of various searching algorithms. Sorting algorithms: Bubble sort, Selection sort, Insertion Sort, Shell Sort. Divide and Conquer Sorting: Merge, Quick and Heap Sort, Choice of Pivot and Worst case, Time and Space analysis. Comparison of various sorting algorithms. Apply searching/sorting algorithms methods and justify the most suitable method, for the given following scenario: library catalog search optimization, E-commerce product sorting, profile searches in a social media platform, financial transactions, medical records sorting for efficient retrieval, locate specific DNA sequences in bioinformatics applications, smartphone contact search optimization.	
	Self-Learning Topics: *Parallel and Distributed Algorithms in Machine Learning.*	

	Learning Outcomes : *A learner will be able to* *LO 5.1: Compare the performance of various searching and sorting algorithms to choose the most appropriate algorithm to solve the given problem. (P.I.-2.2.4).* *LO 5.2:Analyze and identify sources of error and inherent limitations in searching and sorting algorithms, considering their efficiency, applicability, and performance measures. (P.I.-2.4.3).* LO 5.3: *Apply and adapt tools and techniques to solve practical problems in diverse contexts such as media profile searches, financial transactions, medical records sorting, and smartphone contact optimization. (P.I.-5.1.2)* *LO 5.4 :Source and comprehend technical literature and credible sources to deepen understanding of advanced searching and sorting algorithms.(P.I.-11.3.1)* *LO 5.5: Demonstrate proficiency in applying searching and sorting techniques using mathematical models and clear documentation. (P.I. 1.1.3, 1.3.1, 9.3.1)*	
06.	**Advanced Algorithms** ***Learning Objective/s:*** *To familiarize with advanced algorithms, and apply representation techniques to solve diverse domain problems.* **Contents:** Advance Algorithms (any two): a. Page Rank Algorithm (Web Search) b. Rete Algorithm (Rule-Based Systems) ***Self-Learning Topics:*** *Expectation-Maximization (EM) Algorithm, A priori Algorithm. Beam Search Algorithm.* ***Learning Outcomes :*** *A learner will be able to* *LO 6.1: Compare the performance of various* advanced algo*rithms to choose the most appropriate algorithm to solve the given problem. (P.I.-2.2.4).* *LO 6.2: Learn advanced algorithms to recognize and adapt to the need for lifelong learning in technological changes. (P.I.-11.1.1)* *LO 6.3: Demonstrate the use of advanced algorithms with mathematical techniques and present findings clearly. (P.I. 1.1.3, 1.3.1, 9.1.2)*	**7**
	Course Conclusion	**01**
	Total	**45**

Performance Indicators:

P.I. No.	**P.I. Statement**
1.1.1	Apply mathematical techniques such as calculus, linear algebra, probability and statistics, boolean algebra to solve problems.
1.1.3	Apply advanced mathematical techniques of computer and information science to describe/solve/construct a mathematical model of a system.
1.3.1	Apply fundamental engineering concepts to solve engineering problems.
1.4.1	Apply concepts of electronics and communication engineering and allied disciplines to solve engineering problems.
2.1.1	Articulate problem statements and identify primary objectives and key constraints

2.2.4 Compare and contrast alternative solutions to select the best methodology.

2.3.2 Identify assumptions (mathematical and physical) necessary to allow modeling of a system at the level of accuracy required.

2.4.1 Apply Engineering mathematics and computations to solve mathematical models

2.4.2 Produce and validate results through skilful use of contemporary engineering techniques.

2.4.3 Identify sources of error in the solution process, and limitations of the solution.

2.4.4 Extract desired understanding and conclusions consistent with objectives and limitations of the analysis

3.2.1 Apply formal idea generation tools to build models/prototypes in order to develop multiple engineering design solutions.

3.3.1 Apply formal decision-making tools to select optimal engineering design solutions for further development.

5.1.1 Identify modern hardware and software engineering tools, techniques and resources for engineering activities.

5.1.2 Use/adapt/modify/create tools and techniques to solve engineering problems

9.1.2 Create clear, well-constructed, and well-supported written engineering documents and/or presentation.

9.3.1 Create engineering-standard figures, reports and drawings to complement writing and presentations

11.1.1 Describe the rationale for the requirement for continuing professional development/.

11.1.3 Develop ability to learn independently through methods distinct from instructor provided materials.

11.3.1 Source and comprehend technical literature and other credible sources of information.

Course Outcomes: A learner will be able to -

1. Demonstrate proficiency in selecting appropriate linear data structures, representations, analysis and operations for effective solving problem. (LO 1.1 ,LO 1.2 ,LO 1.3, LO 1.4, LO 2.1 ,LO 2.2 ,LO 2.3, LO 2.4)
2. Demonstrate proficiency in selecting appropriate non-linear data structures, representations, analysis and operations for effective solving problem.
(LO 3.1 ,LO 3.2 ,LO 3.3, LO 3.4, LO 4.1 ,LO 4.2 ,LO 4.3, LO 4.4)

3 Apply algorithmic analysis techniques and modern tools to effectively utilize searching and sorting algorithms in real-world problem-solving. (LO 5.1 ,LO 5.2 ,LO 5.3, LO 5.4)

4 Implement and analyse algorithms to solve complex domain problems. (LO 6.1 ,LO 6.2)

5 Demonstrate clear and logically structured technical documentation and presentations to communicate algorithmic solutions effectively. (LO 5.5, LO 6.3)

CO-PO Mapping Table with Correlation Level

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CEMDM01.1	3	3	-	-	-	-	-	-	-	-	2
CEMDM02.2	-	2	3	-	2	-	-	-	-	-	2
CEMDM03.3	-	3	-	-	2	-	-	-	-	-	2
CEMDM04.4	-	2	-	-	-	-	-	-	-	-	2
CEMDM04.5	3	-	-	-	-	-	-	-	3	-	-
Average	3	3	3	-	2	-	-	-	-	-	2

Text Books:

1. "*Data Structures and Algorithm Analysis*" by Clifford A. Shaffer, Dover Publications, 3rd Edition (2007).
2. "Algorithm Design Manual" by Steven S. Skiena, Springer, 2nd Edition (2008)
3. "*Introduction to Algorithms*" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, The MIT Press, 3rd Edition (2009).
4. "*Programmes and Data Structures in C*", Leen Ammeral, Wiley Professional Computing, G.W Rowe
5. "*Data Structures and Algorithms Made Easy*" by Narasimha Karumanchi, CareerMonk Publications, 1st Edition (2011)
6. "*Introduction to the Design and Analysis of Algorithms*" by Anany Levitin, Pearson, 3rd Edition (2011)
7. "*Data Structures using C and C++* ", Y Langsam, MJ Augenstein and A.M, Tanenbaum, Prentice Hall India, Second Edition 2015
8. "Data Visualization: A Practical Introduction", Kieran Healy, 1st Edition, 2018, Princeton University Press.

Reference Books :

1. *"Data Structures and Algorithms in Python"* by Michael T. Goodrich, Roberto Tamasin, and Michael H. Goldwasser, Wiley, 1st Edition (2013).
2. "*Algorithms in C++"* by Robert Sedgewick, Addison-Wesley, 3rd Edition (2001)
3. "*Data Structures and Algorithm Analysis in C++"* by Mark A. Weiss, Pearson,4th Edition (2013)
4. "*Algorithms"* by Robert Sedgewick and Kevin Wayne, Addison-Wesley, 4th Edition (2011)
5. "*Cracking the Coding Interview*" by Gayle Laakmann McDowell, CareerCup, 6th Edition (2015)
6. Data Visualization with Tableau, Alexander Loth, 2016, Packt Publishing.

Other Resources :

1. NPTEL Course: Introduction to Data Structures and Algorithms, by Prof. Naveen Garg, Department of Computer Science and Engineering, IIT Delhi.

Web link- https://nptel.ac.in/courses/106/102/106102064/

2. NPTEL Course: Design and Analysis of Algorithms. by Prof. Madhavan Mukund, Department of Computer Science and Engineering, Chennai Mathematical Institute.
Web link-https://nptel.ac.in/courses/106/106/106106131/

A. IN-SEMESTER ASSESSMENT (50 MARKS)

1. Continuous Assessment (20 Marks)

Suggested breakup of distribution

a. Numerical Assignments (min 20 problems) - 05 marks

b. One Class test based on above numerical assignments- 05 marks

c. Technical report writing - 05 Marks

d. Regularity and active participation- 05 marks

2. Mid Semester Exam (30 Marks)

Mid semester examination will be based on 40% to 50% syllabus.

B.END SEMESTER EXAMINATION (50 MARKS)

End Semester Examination will be based on syllabus coverage up to the Mid Semester Examination (MSE) carrying 20% to 30 % weightage, and the syllabus covered from MSE to ESE carrying 70 to 80% weightage.

Course Type	Course Code	Course Name	Credits
MDM	CEMDM402	DATABASE MANAGEMENT SYSTEMS	03

Examination Scheme					
Distribution of Marks			Exam Duration (Hrs.)		Total Marks
In-semester Assessment		End Semester Exam (ESE)			
Continuous Assessment	Mid-Semester Exam (MSE)		MSE	ESE	
20	30	50	1.5	2	100

Pre-requisite :

1. ECMDM301- Data Structures and Algorithms
2. BSC101- Engineering Mathematics-I

Program Outcomes addressed :

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design/Development of Solutions
4. PO5: Modern tool usage
5. PO11: Lifelong Learning

Course Objectives :

1. To teach students the fundamental principles of database management, including data models, architectures, Entity-Relationship (ER) diagrams, and transaction management techniques.
2. To enable students to analyse real-world problems and design database schemas using ER models and normalization techniques to develop efficient solutions.
3. To guide students in developing proficiency with SQL queries using modern database tools to solve practical, real-world problems.
4. To introduce students to advanced database topics such as indexing, hashing, concurrency control, and distributed databases for effective management of large-scale systems.
5. To assign problem-solving tasks, case studies, and projects that help students apply DBMS concepts in designing and testing data management systems.
6. To introduce emerging database technologies to address modern data management challenges.

Module	Details	Hrs
	Course Introduction The Database Management Systems (DBMS) course is essential for learning how to effectively manage, organize, and optimize data in today's data-driven world. It covers core concepts such as database architectures, design methodologies, normalization techniques, querying, ACID properties, Transaction Control Commands (TCL),	**01**

	concurrency control (deadlock), and recovery systems relevant to modern industries. Students will learn to analyze case studies, tasks and design relational database systems to address real-world challenges. This course provides a foundation for advanced studies in Advanced DBMS, Cloud Data Management, Big Data Analytics, and AI-driven database systems, emphasizing intelligent decision-making, predictive modeling, and database optimization.	
01.	**Introduction to Database Management Systems** ***Learning Objective/s:*** *To introduce the fundamental concepts of database management systems, covering data models, architectures, database languages, administrative roles, storage techniques, and access methods in modern applications..*	**6**
	Contents: Introduction to Databases - Database terminology, evolution of database systems, database languages- Data Definition Language (DDL), Data Manipulation Language (DML), and Data Control Language (DCL). Role of Database Administrators (DBAs). Different storage structures and technologies like RAID and flash storage, and the advantages of database systems over traditional file management systems. Types of data models (Network, Hierarchical, Relational, Entity-Relationship and Object-Oriented Models). Types of database architecture, degrees of data abstraction (physical, logical, and view levels) and the concept of data independence in database systems. Indexing and Access Methods: clustered and non-clustered indices, B+ tree indices, hashed files, bitmap indices for efficient data access.	
	Self-Learning Topics: 1. *Graph Data Mode for modern applications like social networks and recommendation engines (e.g., Amazon Neptune).* 2. *Time Series Data (time stamped data) applicable for IoT, financial, and analytics applications (e.g., InfluxDB, TimescaleDB).* ***Learning Outcomes :*** *A learner will be able to* *LO1.1: Apply fundamental concepts of database systems and the role of Database Administrators in solving numerical problems on storage, indexing, and query optimization. (PI: 1.1.1)* *LO1.2:Apply concepts of database architectures and data abstraction techniques (physical, logical, and view levels) to design solutions that ensure data independency and efficient data management. (PI: 1.4.1)* *LO1.3: Analyze the given indexing and access methods problem statements and identify suitable indexing techniques to improve performance issues in database systems. (PI: 2.1.1)* *LO1.4: Compare storage structures (RAID, flash storage) and traditional file systems to select the best option for a given database application. (PI: 2.2.4)*	

02.	**Database Design using ER-Model** ***Learning Objective/s:*** *Design and analyze ER models, translating them into relational schemas for effective problem-solving.*	**9**
	Contents: The Entity-Relationship (ER) Model- Entity, attributes, keys, relationship, cardinality, weak and strong entity sets. Design of DBMS using ER-Diagrams and their constraints, data integrity rules Relational Database Design- Codd's rules and their significance, relational schemas and logical views of data. Introduction to Unified Modeling Language (UML). Case Studies- Designing an ER model for various applications (e.g., hospital management system).	
	Self-Learning Topics: *Explore open source tools like MySQL Workbench (https://www.mysql.com/products/workbench/), Draw.io (https://app.diagrams.net/), Lucid chart (https://www.lucidchart.com/pages/examples/er-diagram-tool), to learn database design, entity-relationship modeling, schema creation.*	
	Learning Outcomes : *A learner will be able to* *LO2.1: Analyze the given problem statement by identifying key entities, attributes, relationships, and constraints essential for creating an ER model to address the given problem. (PI: 2.1.1.)* *LO2.2: Compare and contrast alternative ERD solutions and select the best ERD solution. (PI:2.2.4)* *LO2.3: Apply different cardinality entity relation types to develop multiple ERD design solutions. (PI: 3.2.1)* *LO2.4: Refine a conceptual ERD model into a detailed relational schema, ensuring consistency with functional requirements and constraints. (PI: 3.4.1)* *LO2.5: Identify open source tools such as open source tools like MySQL Workbench, DBDiagram.io, ModelDB, PlantUML applicable for generating ER diagrams and relational schemas in database design. (PI: 5.1.1)* *LO2.6: Demonstrate proficiency in using open source tools like MySQL Workbench, Draw.io , Lucid chart for generating ERD and relational schema. (PI: 5.2.2)*	
03.	**Relational Algebra and Calculus** ***Learning Objective/s:*** *To generate the queries using relational algebra and calculus to effectively manipulate and retrieve data from relational databases.*	**5**
	Contents: Relational Algebra: Introduction, unary and binary relational operation, selection and projection, set theory operations, binary relational operations, renaming, joins, division, syntax and semantics, relational operations (aggregate, grouping, etc.), relational comparison. Relational Calculus: Tuple relational calculus (TRC), Domain relational calculus (DRS), comparison between TRC and DRC.	

	Self-Learning Topics: *Examine the role of relational algebra in data warehousing and ETL (Extract, Transform, Load) processes.* ***Learning Outcomes :*** *A learner will be able to* *LO3.1: Apply fundamental principles of relational algebra to generate queries for efficient data retrieval from relational databases. (PI: 1.1.1)* *LO3.2: Apply appropriate operations and expressions of relational algebra and relational calculus to solve complex database problem. (PI: 1.4.1)* *LO3.3: Analyze a given database problem statement, identifying key requirements and constraints, and generate appropriate relational algebra and calculus expressions to address the problem effectively. (PI: 2.1.1.)* *LO3.4: Compare Tuple Relational Calculus (TRC) and Domain Relational Calculus (DRC) to evaluate their suitability for database queries. (PI:2.2.4)*	
04.	**Normalization in RDBMS** ***Learning Objective/s:*** *Analyze functional dependencies and apply normalization (1NF, 2NF, 3NF, BCNF) to optimize database design.*	**6**
	Contents: Functional Dependency: Basic concepts, Closure of an Attribute set, decomposition using functional dependency, Desirable Properties of Decomposition (Lossless join, Loss join, Dependency Preservation) Concept of normalization: The need for normalization in RDBMS, Normal Forms- 1NF, 2NF and 3NF, Boyce-Codd Normal Form (BCNF)	
	Self-Learning Topics: *Join Dependencies and Fifth Normal Form* ***Learning Outcomes :*** *A learner will be able to* *LO4.1: Apply mathematical principles of set theory and attribute algebra in functional dependencies and normalization techniques to eliminate redundancy (PI: 1.1.1)* *LO4.2: Apply properties of functional dependencies, decomposition techniques, and normalization techniques to optimize database design. (PI: 1.4.1)* *LO4.3: Analyze functional dependencies in a given database schema and identify the need of Normalization (PI: 2.1.1)* *LO4.4: Compare different normalization forms (1NF, 2NF, 3NF, BCNF) and process them to achieve the highest normalization form, eliminating redundancy and dependency issues in database design. (PI: 2.2.4)*	
05.	**Structured Query Language (SQL)** ***Learning Objective/s:*** *Generate and execute SQL queries with constraints, views, and triggers using SQL tools.*	**11**

	Contents: SQL Introduction: SQL Data Definition, basic SQL query structure, set operations, nested subqueries, aggregation, null values, join expressions, Views in SQL, triggers, Data Control Commands Constraints: Entity integrity constraint, key constraints, domain Constraints, referential integrity, check constraints. SQL Views: Introduction to SQL Views, data independency, security, updates on views, comparison between tables and views. SQL Tools: Use any one tool for practice- MySQL, ORACLE 10G, POSTGRESQL.	
	Self-Learning Topics: *-Assign case studies on - Use of AI/ML tools into modern database management systems.* *-Python scripts to automate SQL tasks on cloud platforms like AWS and Azure.* ***Learning Outcomes :*** *A learner will be able to* *LO5.1: Identify modern engineering tools like MySQL, ORACLE 10G, POSTGRESQ, pgAdmin applicable for creating SQL queries in database system. (PI: 5.1.1)* *LO5.2: Demonstrate proficiency in using modern tools like MySQL, ORACLE 10G, POSTGRESQ, pgAdmin to create and execute SQL queries to solve given problems in database management systems. (PI: 5.2.2)* *LO5.3: Recognize the importance of staying updated with SQL advancements like optimization, cloud databases, NoSQL, and AI integration for continuous professional growth. (PI: 11.2.2)* *LO5.4: Source and assess relevant technical literature to track evolving trends in SQL and database management, integrating emerging methodologies to support continuous learning. (PI: 11.3.1)*	
06.	**Transaction Management and Recovery Systems** ***Learning Objective/s:*** *Design transaction schedule using serializability tests, analyze concurrency control techniques, design check rules and comprehend recovery methods to ensure data consistency and durability.*	**6**
	Contents: Transaction Management - Transaction concept, Transaction states, ACID properties, Transaction Control Commands (TCL), design transaction schedule using serializability tests, analyse concurrency control techniques, design check rules, View-based serial conflicts. Techniques for concurrency control- Two-phase locking techniques, Timestamp ordering-based concurrency control, Lock-based protocols, Multiple granularities locking, and Deadlock handling. Recovery System - Recovery concepts, Deferred and Immediate update-based recovery, Shadow paging, Log-based recovery, Undo-Redo recovery techniques, Write-ahead logging, Buffer management.	

	Self-Learning Topics: 1. *Explore the role of block chain in transaction integrity and ACID properties in distributed systems.* 2. *Advanced recovery techniques in NoSQL databases and their comparison with traditional RDBMS methods.*	
	Learning Outcomes : *A learner will be able to* *LO6.1: Design an efficient transaction schedule using serializability tests and precedence graphs to manage multiple transactions effectively. (PI: 3.2.1)* *LO6.2: Design check rules for transaction states, ACID properties, and Transaction Control Commands (TCL) to ensure effective transaction management and consistency in a database system. (PI:3.4.1)* *LO6.3:Recognize and adapt to advancements in transaction management, emphasizing lifelong learning to stay updated for secure and efficient database operations. (PI: 11.2.2)* *LO6.4:Utilize technical resources to explore evolving trends in improve concurrency control and recovery methods. (PI: 11.3.1)*	
	Course Conclusion	**01**
	Total	**45**

Performance Indicators:

P.I. No.	**P.I. Statement**
1.1.1	Apply mathematical techniques such as calculus, linear algebra, probability and statistics, boolean algebra to solve problems.
1.4.1	Apply concepts of engineering and allied disciplines to solve engineering problems.
2.1.1	Articulate problem statements and identify objectives.
2.2.4	Compare and contrast alternative solutions to select the best methodology.
3.2.1	Apply formal idea generation tools to develop multiple engineering design solutions.
3.4.1	Refine a conceptual design into a detailed design within the existing constraints (of *the resources)*
5.1.1	Identify modern engineering tools such as computer-aided drafting, modeling and analysis; techniques and resources for engineering activities
5.2.2	Demonstrate proficiency in using discipline-specific tools.
11.2.2	Recognize the need and be able to clearly explain why it is vitally important to keep current regarding new developments in your field.
11.3.1	Source and comprehend technical literature and other credible sources of information

Course Outcomes: A learner will be able to -

1. Apply fundamental principles of database management systems, including data models, architectures, and database languages, to solve real-world problems. (*LO1.1, LO1.2, LO1.3, LO1.4)*
2. Design Entity-Relationship models and translate them into relational schemas while ensuring data integrity and compliance with constraints.(*LO2.1,LO2.2, LO2.3. LO2.4, LO2.5, LO2.6)*
3. *Apply relational algebra and calculus operations to retrieve and manipulate data from relational databases.(LO3.1, LO3.2, LO3.3, LO3.4)*
4. *Apply normalization techniques (1NF, 2NF, 3NF, BCNF) to optimize database designs and improve relational schemas.(LO4.1, LO4.2, LO4.3, LO4.4)*
5. Create and execute SQL queries, incorporating constraints, views, and triggers, to solve database-related problems using SQL tools.(*LO5.1,LO5.2, LO5.3. LO5.4)*
6. Analyze concurrent transaction schedules and apply conflict serializability tests to convert them into serial schedules. (*LO6.1,LO6.2, LO6.3. LO6.4*)

CO-PO Mapping Table with Correlation Level

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CEMDM402.1	3	3	-	-	-	-	-	-	-	-	-
CEMDM402.2	-	3	3	-	3	-	-	-	-	-	-
CEMDM402.3	3	3	-	-	-	-	-	-	-	-	-
CEMDM402.4	3	3	-	-	-	-	-	-	-	-	-
CEMDM402.5	-	-	-	-	3	-	-	-	-	-	3
CEMDM402.6	-	-	3	-		-	-	-	-	-	3
Average	3	3	3	-	3	-	-	-	-	-	3

Text Books :

1. " Fundamentals of Database Systems " by Elmasri and Navathe, 5th Edition , Pearson Education, (2007).
2. " Database System Concepts " by Korth, Sudarshan , 6th Edition, McGraw Hill (2008).
3. “Database Systems Design, Implementation and Management”, Peter Rob and Carlos Coronel, Thomson Learning 5th edition, 2011.
4. " Database Systems: The Complete Book " by H Garcia-Molina, JD Ullman and Widom , 2nd Ed., Prentice-Hall, 2008.
5. " Database System Concepts " by Silberschatz, H Korth and S Sudarshan, 6th Ed., McGraw-Hill, 2010.
6. “Fundamentals of Database Systems “,R Elmasri, S Navathe , 6th edition, Addison-Wesley, 2010.
7. "Database Management Systems”, R Ramakrishnan, J Gehrke, 3rd Ed., McGraw-Hill, 2002.

Reference Books :

1. "Database Systems Design, Implementation and Management", Peter Rob, Carlos Coronel, 5th Edition, Thomson Learning, 2007.
2. "SQL and PL/SQL for Oracle 11g, Black Book", P.S. Deshpande, Black Book, Dreamtech Press, 2010.
3. "Introduction to Database Management", Mark L. Gillenson, Paulraj Ponniah, Wiley, 2006.
4. "Database Management Systems", Raghu Ramkrishnan, Johannes Gehrke, TMH (Tata McGraw Hill), 2003.
5. "Database Management Systems", Debabrata Sahoo, Tata McGraw Hill, 2009.

Other Resources :

1. NPTEL Course: Data Base Management System, by Prof. Partha Pratim Das, Prof. Samiran Chattopadhyay, Department of Computer Science and Engineering, IIT Kharagpur.
   Web link- https://onlinecourses.nptel.ac.in/noc22_cs91/preview
2. Khan Academy- Intro to SQL: Querying and managing data
   Web link- https://www.khanacademy.org/computing/computer-programming/sql

DBMS Online free Course, DBMS (Database Management system) Complete Playlist
3. Gate Smashers,
https://youtube.com/playlist?list=PLxCzCOWd7aiFAN6I8CuViBuCdJgiOkT2Y&feature=shared

DBMS Online Course with Certificate: Master the Fundamentals and Advanced Concepts, by
4. Srikanth Varma, Lead DSML Instructor at Scaler.
Web link- https://www.scaler.com/topics/course/dbms/

IN-SEMESTER ASSESSMENT (50 MARKS)

1. Continuous Assessment (20 Marks)

Suggested breakup of distribution

a) Problem statements / case studies/ tasks (Design of ERD DBMS schemas and SQL based Questions): 5 Marks.

b) Numerical Assignments (based on indexing, storage, relational algebra operators, functional dependences and closure, normalization, etc) (min 20 problems) : 05 marks.

c) One class test based on (a) and (b) :05 marks

d) Regularity and active participation: 05 marks

2. Mid Semester Exam (30 Marks)

Mid semester examination will be based on 40% to 50% syllabus.

END SEMESTER EXAMINATION (50 MARKS)

End Semester Examination will be based on syllabus coverage up to the Mid Semester Examination (MSE) carrying 20 to 30 % weightage, and the syllabus covered from MSE to ESE carrying 70 to 80% weightage.

Course Type	Course Code	Course Name	Credits
MDM	CEMDM503	CLOUD COMPUTING	03

Examination Scheme					
Distribution of Marks			Exam Duration (Hrs.)		Total Marks
In-semester Assessment		End Semester Examination (ESE)			
Continuous Assessment	Mid-Semester Exam (MSE)		MSE	ESE	
20	30	50	1.5	2	100

Pre-requisite:

1. NIL

Program Outcomes addressed:

1. PO 1: Engineering Knowledge
2. PO 2: Problem analysis
3. PO 4: Conduct investigations of complex problems
4. PO 5: Engineering Tool
5. PO 11: Life-long learning

Course Objectives:

1. Understand the core principles and need of cloud computing applications.
2. Explore the architecture and working of cloud platforms like AWS, Microsoft Azure, and Google Cloud
3. Understand various cloud computing services and platforms
4. Understand application design concepts, security aspects and the advances, in cloud computing
5. Address challenges in cloud computing, including security, privacy, cost optimization, and scalability.

Module	Details	Hrs.
	Course Introduction The course "Cloud Computing" is designed to provide students with a comprehensive understanding of the fundamental concepts, technologies, and applications of cloud computing in modern engineering. In today's interconnected world, cloud computing plays a pivotal role in enabling scalable, on-demand access to computational resources, storage, and applications over the internet. It serves as the backbone for a wide range of domains, including communication networks, IoT systems, real-time data analytics, and artificial intelligence. Emphasis will be placed on understanding how cloud computing transforms traditional telecommunication systems by providing efficient resource management, virtualization, and high availability for network services.	**01**

01.	**Introduction to Cloud** ***Learning Objective:*** ***1. To understand the fundamental concepts of cloud computing, including its characteristics, components,*** ***2. To explore cloud deployment models (public, private, community, hybrid) and service models (IaaS, PaaS, SaaS), and their practical applications in solving real-world problems***	**03-05**
	Contents: Introduction to Cloud Computing, Cloud Characteristics, Cloud Computing Components. Cloud Deployment model (Cloud types- Public, Private, Community, Hybrid), Service Models- (IaaS, PaaS, SaaS,). Introduction to popular cloud platforms (AWS, Azure, GCP)	
	Self-Learning Topics: *Applications and use cases of cloud computing (e.g., e-commerce, healthcare, IoT)*	
	Learning Outcomes: *A learner will be able to* *LO 1.1: Comprehend the basics of cloud computing and the terminology associated (PI 1.4.1)* *LO 1.2: Elaborate the advantages of cloud computing over traditional computing for given scenario (PI 1.3.1, PI 6.1.1)* *LO 1.3: Analyze and Compare Architectures identifying the unique features and benefits of each. (PI 1.4.1, 2.2.4)* *LO 1.4: Suggest the best suitable type of cloud for given scenario. (PI 4.3.4 6.1.1)*	
02.	**Cloud Architecture Design** ***Learning Objective:*** *1. Introduce students to the benefits of virtualization, along with the design requirements and mechanisms for implementing virtualization in hardware and software system* *2. Explore Cloud resource management, its characteristics and, cloud orchestration tools.*	**8-10**
	Contents: Cloud computing architecture: front-end and back-end, Virtualization: concepts, types, and hypervisors, Containers vs. Virtual Machines. Cloud resource management and scalability, Load balancing and auto scaling in cloud environments, Introduction to cloud orchestration tools. Multi-tenancy, elasticity, and fault tolerance, Micro services architecture and its relevance to the cloud.	
	Self-Learning Topics:	

	Case study: Xen Architecture white paper	
	Learning Outcomes: *A learner will be able to* *LO 2.1: Elaborate Concepts and Benefits of Virtualization (PI 1.4.1)* *LO 2.2: Define the different characteristics of cloud resource management. (PI 1.3.1)* *LO 2.3: Evaluate and compare virtualization mechanisms like Hypervisors, Xen Architecture, Para Virtualization (PI 1.3.1, 2.2.4)*	
03.	**Cloud Computing Services** ***Learning Objective:*** ***1.*** *make students aware of pricing models of cloud service provides and their cost optimization techniques.* ***2.*** *introduce the various* **services** *offered by any one major cloud service provider.*	**10-12**
	Contents: Overview of major cloud service providers (AWS, Azure, GCP), Comparative analysis of cloud services, Understanding pricing models and cost optimization. Services on any one the major cloud service providers: Compute Services, Storage Services, Database Services, Data Storage Approaches - Relational (SQL) Approach, Non-Relational (No-SQL) Approach, Application Services, Content Delivery Services, Analytics Services, Deployment & Management Services, Identity & Access Management Services Demonstrations on the prime cloud service providers utilities	
	Self-Learning Topics: ***Case study: Setting up and navigating a free-tier AWS/GCP account***	
	Learning Outcomes: *A learner will be able to* *LO 3.1: Select and justify the best cloud services for given scenario (PI 1.4.1, 2.2.3).* *LO 3.2: Analyze and Compare Cloud Services (PI 1.4.1, 2.2.4)* *LO 3.3: Utilize any one service offered by one of cloud providers based on given scenario (PI 5.1.1)* *LO 3.4: Analyze and interpret information from given resources about computing services to answer related questions and demonstrate understanding. (PI 4.3.4)*	
04.	**Cloud Application Design** ***Learning Objectives:*** *1. Make the students aware of cloud application design methodologies, including Service-Oriented Architecture (SOA), Cloud Component Models, Model-View-Controller (MVC), RESTful Web Services and data storage approaches.* *2. Introduce the key design considerations for cloud applications*	**6-8**

	Contents: Design Considerations for Cloud Applications - Scalability, Reliability & Availability, Security, Maintenance & Upgradation, Performance Cloud Application Design Methodologies - Service Oriented Architecture, Cloud Component Model, IaaS, PaaS and SaaS services for cloud applications, Model View Controller, RESTful Web Services.	
	Self-Learning Topics: ***Case study: Tools: Docker, Kubernetes, Jenkins, Terraform***	
	Learning Outcomes: *A learner will be able to* *LO 4.1: Comprehend different design consideration for Cloud design applications. (PI:1.4.1).* *LO 4.2: Elaborate on different cloud application design methodologies. (PI 1.3.1)* *LO 4.3: Apply the design considerations to a practical scenario and propose design solutions for a societal application. (PI 3.1.3, 6.1.1)*	
05.	**Cloud Security** ***Learning Objective/s:*** *1. Learn cloud security fundamentals along with security mechanisms like cloud storage gateways and firewalls.* *2. learn the **AAA (Authentication, Authorization, and Accounting) model** and its application in cloud environments, including SSO (Single Sign-On) and resource utilization management.*	**5-7**
	Contents: Cloud security fundamentals: data encryption, identity, and access management (IAM), Cloud Storage Gateways, Cloud Firewall. AAA Administration for Clouds -AAA model, SSO for Clouds, Authentication management and Authorization management in clouds. Case studies: Cloud security breaches and lessons learned	
	Self-Learning Topics: *Hands-on: Setting up IAM roles and policies in AWS/Azure*	
	Learning Outcomes: *A learner will be able to* *LO 5.1: Comprehend AAA model for cloud computing. (PI 1.4.1)* *LO 5.2: Analyze cloud security breaches reported in contemporary literature and study their impact on the society, from the case studies (PI 6.2.1)*	
06.	**Emerging Trends and Advanced Topics**	**5-7**
	Learning Objective/s: ***1.*** *Develop a foundational understanding of APIs, cloud SDKs, DevOps, and CI/CD principles, and demonstrate their implementation in cloud-based environments.*	

	2. Explore advanced topics such as edge computing integration with cloud systems, Cloud AI/ML services, and IoT-cloud architectures to design innovative and scalable solutions	
	Contents: Introduction to APIs and cloud SDKs, DevOps and CI/CD in the cloud. Edge computing and cloud computing integration. Cloud AI and ML services. Internet of Things (IoT) and cloud computing.	
	Self-Learning Topics: *Deploying a web application using cloud services*	
	Learning Outcomes: *A learner will be able to* *LO 6.1: Describe the basics of APIs, cloud SDKs, DevOps, CI/CD, and their roles in cloud computing. (PI 1.4.1)* *LO 6.2: Identify the relevant component and develop the pipeline for integrating and IoT service with cloud for predictive analysis and action. (PI 4.1.2, 5.1.2)* *LO 6.2: Evaluate the impact of IoT and cloud computing integration on performance, scalability, and real-world applications from certain case studies (PI 11.1.3).*	
	Course Conclusion	**01**
	Total	**45**

Performance Indicators:

P.I. No.	**P.I. Statement**
1.3.1	Apply fundamental engineering concepts to solve engineering problems.
1.4.1	Apply concepts of electronics and communication engineering and accepted practice areas to solve engineering problems.
2.2.2	Identify resources to assemble/integrate information using mathematical tools.
2.2.4	Compare and contrast alternative solutions to select the appropriate methodology.
4.3.4	Synthesize information and knowledge about the problem from the data to reach appropriate conclusions
5.1.1	Identify modern hardware and software engineering tools, techniques and resources for engineering activities.
6.2.1	Comprehend legal requirements relevant to engineering design and propose solution complying to engineering standards.
11.1.3	Develop ability to learn independently through methods distinct from instructor provided materials.

Course Outcomes: A learner will be able to -

1. Recognize the need to use cloud computing as an alternative to traditional computing and describe concepts related to cloud computing. *(LO 1.1,1.2,1.3, LO 3.4)*
2. Determine objectives & design requirements for virtualization and identify various characteristics of cloud resource management. *(LO 2.1, 2.2)*
3. State methodologies of accessing/deploying services on any one of the cloud providers. *(LO3.1,3.2,3.3,3.4)*

4. Compare various clod architectures, virtualization techniques, services *(LO 1.3, LO 2.3, LO 3.2).*
5. Identify data security concerns and recent trends in Cloud computing *..(LO 5.1,5.2,5.3 LO 6.1,6.2)*

CO-PO Mapping Table with Correlation Level

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
1.	3	2	-	2	-	-	-	-	-	-	-
2.	3	-	-	-	-	-	-	-	-	-	-
3.	2	3	-	2	2	-	-	-	-	-	-
4.	2	3	-	-	-	-	-	-	-	-	-
5.	3	-	-	-	-	2	-	-	-	-	2
Average	**3**	**3**	**-**	**2**	**2**	**2**	**-**	**-**	**-**	**-**	**2**

Text Books :

1. Cloud Computing - A Hands-on Approach - Arshdeep Bahga and Vijay K. Madisetti
2. Mastering Cloud Computing: Foundations and Applications Programming Paperback – by Rajkumar Buyya , Christian Vecchiola , S.Thamarai Selvi , Publisher: Morgan Kaufmann
3. Amazon Web Services For Dummies (For Dummies Series) Paperback by Bernard Golden, Publisher: John Wiley & Sons
4. The Cloud Computing Book: The Future of Computing Explained" , Douglas E. Comer
5. Cloud Computing for Dummies, Judith Hurwitz Daniel Kirsch

Reference Books :

1. Cloud Computing Black Book: Kailash Jayaswal, Jagannath Kallakurchi, Donald J. Houde, Dr. Deven Shah by Kogent Learning Solutions, Publisher: Dreamtech Press
2. Cloud Computing Concepts Technology and Architecture - Erl second hand book online from Used Books Factory
3. Practical Cloud Security by Chris Dotson, Publisher(s): O'Reilly Media, Inc. ISBN: 9781492037514
4. AWS Whitepapers & Guides https://aws.amazon.com/whitepapers/
5. Azure whitepapers https://azure.microsoft.com/en-in/resources/whitepapers/
6. Google Cloud whitepapers https://cloud.google.com/whitepapers

Other Resources :

1. NPTEL Swayam Course on Cloud computing By Prof. Soumya Kanti Ghosh https://nptel.ac.in/courses/106/105/106105167/

2. Cloud Computing and Distributed Systems By Prof. Rajiv Misra https://onlinecourses.nptel.ac.in/noc22_cs18/preview
3. Google Cloud Computing Foundation Course https://nptel.ac.in/courses/106/105/106105223

A. IN-SEMESTER ASSESSMENT (50 MARKS)

1. Continuous Assessment-Theory (20 Marks)

Suggested breakup of distribution

a) Open book Tests*: 05 Marks

b) One hand –on assignment: 10 marks

c) Regularity and active participation: 05 Marks

*Average of multiple tests conducted during continuous assessment

3. Mid Semester Examination (30 Marks)

Mid semester examination will be based on 40% to 50% syllabus.

B. END SEMESTER EXAMINATION (50 MARKS)

End Semester Examination will be based on syllabus coverage up to the Mid Semester Examination (MSE) carrying 20% to 30% weightage, and the syllabus covered from MSE to ESE carrying 70% to 80% weightage.

Course Type	Course Code	Course Name	Credits
MDM	CEMDM604	SOFT COMPUTING	04

Examination Scheme					
Distribution of Marks			Exam Duration (Hrs.)		Total Marks
In-semester Assessment		End Semester Examination (ESE)			
Continuous Assessment	Mid-Semester Exam (MSE)		MSE	ESE	
20	30	50	1.5	2	100

Pre-requisite:

1. XXPCC405 Engineering Mathematics-IV

Program Outcomes addressed:

1. PO-1: Engineering Knowledge
2. PO-2: Problem Analysis
3. PO-3: Design/Development of Solutions
4. PO-6: The Engineer and The World
5. PO-9: Communication
6. PO-11: Life-Long Learning

Course Objectives:

1. To impart knowledge on the principles and need for soft computing.
2. To teach key methodologies: fuzzy logic, neural networks, and genetic algorithms.
3. To analyse how soft computing is applied to solve real-world problems with the context of sustainability and societal or health issues.
4. To encourage self-learning and collaborative learning through case-studies.

Module	Details	Hrs.
	Course Introduction Soft Computing is a multidisciplinary field that combines concepts from Fuzzy Logic, Neural Networks, Evolutionary Algorithms and Probabilistic Reasoning. In an era driven by **AI and intelligent systems**, soft computing is the backbone of technologies that require: • Automation in unpredictable environments. • Personalized solutions (e.g., recommendation systems). • Decision-making under uncertainty. Soft Computing offers tools and techniques to address challenges that traditional methods struggle to solve. The scope of soft computing in many interdisciplinary applications from robotics and control systems to finance, healthcare, and AI.	**01**

	This course is a foundation for acquiring skills like problem solving, critical thinking, analytical and logical reasoning. The modality of content delivery should encourage self-learning among learners through quality resources on practical case-studies.	
01.	**Introduction to Soft Computing** ***Learning Objective: Introduce students to need and fundamentals of soft computing.***	**06**
	Contents: **Introduction:** Difference between soft computing and hard computing. Real-world applications of soft computing in society. **Components of Soft Computing:** Overview of fuzzy logic, artificial neural networks, and genetic algorithms. Importance of hybrid approaches. **Introduction to Genetic Algorithms:** Biological basis: natural selection and evolution. Basic concepts: population, chromosome, fitness function	
	Self-Learning Topics: Neuro-fuzzy and genetic-fuzzy systems.	
	Learning Outcomes: *A learner will be able to* *LO 1.1: Identify the clear differentiation between soft computing and hard computing and is able to provide best suited examples for soft computing. (PI 2.1.3)* *LO 1.2: Articulate the need analysis and benefits of soft computing in handling uncertainty. (PI 3.1.1)* *LO 1.3: Formulate relationship between concepts of fuzzy logic, neural networks, and genetic algorithms and their roles in societal or any other applications related to soft computing. (PI 1.4.1, 6.1.1)* *LO 1.4: Elaborate the role of population, chromosome, fitness function etc. in genetic algorithm component. (PI 1.4.1)*	
02.	**Fuzzy Logic** ***Learning Objective:*** *To teach concepts of Fuzzy sets and membership functions.* *To enable understanding of Fuzzy relations, compositions and rules.* *To introduce Defuzzification techniques their applications.* *To teach FIS designs from sustainable design perspective.*	**08**
	Contents: **Fuzzy Sets and Systems:** Crisp sets vs. fuzzy sets. Membership functions and their types (triangular, trapezoidal, Gaussian). Operations on fuzzy sets: union, intersection, complement. **Fuzzy Rules and Reasoning:** Fuzzy relations and compositions. Rule-based systems: fuzzy if-then rules. Sustainable design of Fuzzy inference systems (Mamdani and Sugeno). **Defuzzification:** Methods: centroid, weighted average, and max membership. Applications relevant in societal aspects: control systems, decision-making in health sector.	
	Self-Learning Topics: *Use-cases of Mamdani and Sugeno FIS in washing machine*	

	Learning Outcomes: *A learner will be able to* *LO 2.1: Differentiate between crisp sets and fuzzy sets and identify common types of membership functions that are applicable in common real-life scenarios. (PI 1.4.1, 2.1.1, 6.1.1)* *LO 2.2: Perform operations on Fuzzy sets in order to solve given engineering problems. (PI 2.1.2)* *LO 2.3: Design a sustainable fuzzy rule-based systems using if-then rules. (PI 3.1.5, 3.1.6, 6.4.1)* *LO 2.4: Provide solution to commonly observed engineering applications identifying role of Fuzzy systems in them. (PI 2.2.4, 6.1.1)*	
03.	**Introduction to Artificial Neural Networks** ***Learning Objective:*** *To introduce evolution of neural networks and their terminology.* *To facilitate understanding of role on artificial neural networks in machine learning and corresponding criteria.* *To relate different classes of learning with their use-cases.* **Contents:** **Introduction to Neural Networks:** Biological inspiration. Components of a neuron: weights, activation functions. Types of neural networks: feedforward and feedback. **Role of Neural Networks in Machine Learning**: Training, Testing, Cross-validation, Overfitting-under fitting, Stopping Criteria, Performance analysis. **Classification of Learning**: Supervised, Unsupervised and Reinforcement learning. Use case of each. **Applications of Neural Networks in contemporary areas:** Classification, regression, and clustering. ***Self-Learning Topics:*** *Case Study: Predictive modeling in finance or healthcare.* ***Learning Outcomes:*** *A learner will be able to* *LO 3.1: Identify components of neuron and role of activation function. (PI 1.4.1, 2.1.1)* *LO 3.2: Apply, analyze key concepts like training, testing, cross-validation and evaluate performance of a given neural network. (PI 2.1.3)* *LO 3.3: Identify suitable technique among supervised, unsupervised and reinforcement learning to suggest an efficient solution for given problem relevant to society. (PI 2.2.3, 6.4.1)* *LO 3.4: Develop ability to go through the provided resources related to the case study and present review on the given points. (PI 9.1.1, 11.1.3)*	**10**
04.	**Supervised Learning** ***Learning Objectives:*** *To explain supervised learning techniques like regression, classification and neural networks.* *To relate concepts, mathematical background and performance analysis of these techniques.* **Contents:** **Regression Techniques**: Linear regression, polynomial regression **Classification Techniques**: Logistic regression, Support Vector Machine, Decision Trees, Random Forests, Gradient Boost Techniques, K-nearest Neighbors	**12**

	Neural Networks: Perceptron learning rule. Introduction to multi-layer Perceptron (MLP) and backpropagation algorithm.	
	Self-Learning Topics: *Internet resources on: Ensemble learning approaches*	
	Learning Outcomes: *A learner will be able to* *LO 4.1: Illustrate regression and classification techniques through an example. (PI 1.4.1, 2.1.1)* *LO 4.2: Solve illustrative examples related to given supervised learning technique in order to specify the design parameters and provide an efficient solution. (PI 1.4.1, 3.1.6, 6.4.1)* *LO 4.3: Develop a flowchart for the perceptron and the error back-propagation supervised learning algorithm. (PI 9.1.3, 9.3.1)*	
05.	**Unsupervised Learning** ***Learning Objective/s:*** ***To develop understanding of various unsupervised learning methods and their use cases.***	**10**
	Contents: **Clustering Techniques**: K-means clustering, Hierarchical clustering. **Dimensionality Reduction Techniques**: Principal Component Analysis, Singular Value Decomposition **Self-Organizing Maps:** Kohonen self-organizing map networks **Applications** of unsupervised learning in social, cultural, environmental and human health-related issues	
	Self-Learning Topics: *Gaussian Mixture Models, Uniform Manifold Approximation and Projection (UMAP)*	
	Learning Outcomes: *A learner will be able to* *LO 5.1: Illustrate the clustering techniques for providing solution to a given unsupervised learning problem in the context of an application. (PI 1.4.1, 2.1.1, 6.1.1)* *LO 5.2: Accurately describe flow of the dimensionality reduction techniques (PCA and SVD) with a suitable flow-diagram. (PI 9.1.3)* *LO 5.3: Formulate a given problem in terms of KSOMs and design it's specifications to solve the problem for a given application. (PI 3.1.6, 6.1.1)* *LO 5.4: Develop an ability to find appropriate resources for and self-learn modern models like GMM and UMAP. (PI 11.3.1)*	
06.	**Deep Learning Techniques**	**12**
	Learning Objective/s: ***To provide sufficient exposure to learners about contemporary deep learning algorithms.***	

	Contents: **Convolutional Neural Networks**: Architecture, working, and application. **Recurrent Neural Networks**: General Architecture, Variants: LSTM and GRU, application. **Transformers:** Attention mechanism, Positional Encoding, Encoder-decoder architecture, popular models—BERT, GPT.	
	Self-Learning Topics: Generative Models: Generative Adversarial Networks. Autoencoders.	
	Learning Outcomes: *A learner will be able to* *LO 6.1: Apply knowledge of CNN architecture find the output of convolution, pooling and flattening layers for a given input image. (PI 1.4.1, 9.3.1)* *LO 6.2: Write the operation of RNNs using a functional diagram and corresponding mathematical expressions. (PI 2.4.4, 9.3.1)* *LO 6.3: Accurately describe functional aspects of transformers and develop self-learning about GANs and Autoencoders. (9.1.3, 11.1.3)*	
	Course Conclusion	**01**
	Total	**60**

Performance Indicators:

P.I. No.	**P.I. Statement**
1.4.1	Apply concepts of electronics and communication engineering and **allied disciplines** to solve engineering problems.
2.1.1	Articulate problem statements and identify primary objectives and key constraints.
2.1.2	Identify engineering systems, variables, and parameters to solve the problems
2.1.3	Identify the mathematical, engineering and other relevant knowledge that applies to a given problem.
2.2.4	Compare and contrast alternative solutions to select the best methodology.
2.4.4	Extract desired understanding and conclusions consistent with objectives and limitations of the analysis
3.1.1	Recognize that need analysis is key to good problem definition.
3.1.5	Explore and synthesize engineering requirements considering health, safety risks, environmental, cultural and societal issues.
3.1.6	Determine design objectives, functional requirements and arrive at specifications
6.1.1	Identify social, environmental and health related issues with the help of relevant viewpoints from stakeholders and propose engineering solutions.
6.4.1	Describe and evaluate engineering principles for sustainable development pertaining to relevant fields.
9.1.1	Read, understand and interpret technical and/or non-technical information.
9.1.3	Create flow in a document or presentation - a logical progression of ideas so that the main point is clear.
9.3.1	Create engineering-standard figures, reports and drawings to complement writing and presentations.
11.1.3	Develop ability to learn independently through methods distinct from instructor provided materials.
11.3.1	Source and comprehend technical literature and other credible sources of information

Course Outcomes: A learner will be able to -

1. Recognise the need for soft-computing and develop insights into applications of soft-computing. *(LOs: 1.3, 1.4, 2.1, 3.1, 4.1, 4.2, 5.1, 6.1)*
2. Develop sustainable solutions for commonly observed engineering applications using Fuzzy Inference Systems. *(LOs: 1.1, 2.2, 3.2)*
3. Design neural network-based solutions for issues relevant in various societal aspects, with the perspective of efficient computation. *(LOs: 1.3, 2.3, 2.4, 3.3, 4.2, 5.3)*
4. Apply suitable learning method from supervised /unsupervised/deep learning to solve given real-life case-based problems and develop their flow-diagrams. *(LOs: 1.2, 3.4, 4.3, 5.2, 6.1, 6.2, 6.3)*
5. Show-case the ability of finding suitable resources and developing self-learning skills for contemporary topics in ANNs and deep learning. *(LOs: 3.4, 5.4, 6.3)*

CO-PO Mapping Table with Correlation Level

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
XXMDM604.1	2	2	-	-	-	3	-	-	-	-	-
XXMDM604.2	-	2	3	-	-	3	-	-	-	-	-
XXMDM604.3	-	3	-	-	-	-	-	-	-	-	-
XXMDM604.4	2	-	2	-	-	-	-	-	3	-	2
XXMDM604.5	-	-	-	-	-	-	-	-	-	-	-
Average	2	2.6	2.5	-	-	3	-	-	3	-	3

Text Books :

1. S. N. Sivanandam and S. N. Deepa, Introduction to Soft Computing, Wiley India Publications.
2. Simon Haykin, Neural Network- A Comprehensive Foundation, Pearson Education
3. Ian Goodfellow, Yoshua Bengio,Aaron Courville, Deep Learning, MIT Press, 2016
4. Timothy J. Ross, Fuzzy Logic with Engineering Applications, Wiley India Publications
5. Mehrotra Kishan, Mohan C. K. Ranka Sanjay, Elements of Artificial Neural Networks, Penram Publications.

Reference Books :

1. Christopher M Bishop, Neural Networks For Pattern Recognition, Oxford University Press
2. Bart Kosko, Neural networks and Fuzzy Systems, Pearson Education

Other Resources :

1. Course: Fuzzy Logic and Neural Networks by Prof. Dilip Kumar Pratihar, IIT Kharagpur
url: https://onlinecourses.nptel.ac.in/noc21_ge07/preview

2. Course: Neural Network and Applications by Prof. Somnath Sengupta, IIT Kharagpur
url: https://nptel.ac.in/courses/117105084

3. Michael Nielsen, "Neural Networks and Deep Learning", Determination Press, 2015. http://neuralnetworksanddeeplearning.com/

IN-SEMESTER ASSESSMENT (50 MARKS)

1. Continuous Assessment-Theory (20 Marks)

Suggested breakup of distribution

a) Two class tests of five marks each: 10 Marks*

b) Team-pair-share: 05 Marks*

c) Regularity and active participation: 05 Marks

*: **Formative assessments, case-study assignments and collaborative learning tests will be conducted on regular basis and average of all performances throughout the course will be considered as CIA.**

2. Mid Semester Examination (30 Marks)

Mid semester examination will be a pen-paper based timed examination conducted centrally, based on 40% to 50% syllabus.

END SEMESTER EXAMINATION (50 MARKS)

End Semester Examination will be a pen-paper based timed examination conducted centrally; based on syllabus coverage up to the Mid Semester Examination (MSE) carrying 20% to 30% weightage, and the syllabus covered from MSE to ESE carrying 70% to 80% weightage.

Course Type	Course Code	Course Name	Credits
MDM	CEMDL601	MACHINE LEARNING LABORATORY	03

Examination Scheme		
Continuous Assessment	End-Semester Examination (ESE)	Total
25	25	50

Pre-requisite:

1. XXPCC405 Engineering Mathematics-IV

Program Outcomes addressed:

1. PO-1: Engineering Knowledge
2. PO-2: Problem Analysis
3. PO-4: Conduction of complex problem investigation
4. PO-5: Engineering tool usage
5. PO-6: Sustainability
6. PO-7: Ethics
7. PO-8: Individual and group work
8. PO-9: Communication

Course Objectives:

1. To impart knowledge on the principles and need for soft computing.
2. To teach key methodologies: fuzzy logic, neural networks, and genetic algorithms.
3. To analyse how soft computing is applied to solve real-world problems with the context of sustainability and societal or health issues.
4. To encourage self-learning and collaborative learning through case-studies/course project.

Module	Details	Hrs.
	Course Introduction This course, is a laboratory course in Machine Learning which introduces the students to practices of ML. It aims at students being able to apply these ML algorithms in application areas of engineering. This course gives the students a unique opportunity to engage in hands-on techniques of supervised/unsupervised and deep learning and enhance their skill-set as engineers equipped with ML skills to be applied in their domain. The demonstrations mentioned in the syllabus have to be in the form of spoken tutorials to facilitate self-paced learning.	**01**
01.	**Introduction to Machine Learning** ***Learning Objective/s:*** *To make students conversant with terminology, techniques, and performance metrics of machine learning.*	**04**
	Demonstrations on: 1. How to set up a complete pipeline for a ML application	

	2. How to test the ML algorithm using cross-validation, muti-fold validation techniques and the bias-variance trade-off 3. How to evaluate performance of the ML algorithm using accuracy and Precision Measurements, Precision, Recall, F1-score, Confusion Matrix, Receiver Operator Characteristic Curve (ROC Curve)	
	Self-Learning Topics: Data manipulation tools in Python PANDAS)	
	Suggested Task: Given a particular Train-Test Data-set and a standard Python code; do the following: 1. create confusion matrix, and plot the ROC curve. 2. demonstrate the usage of different strategies of validation and comment on the performance. For any one of the following applications: • E-mail spam detection • Digit Recognition • Flower species recognition	
	Learning Outcomes: *A learner will be able to* *LO 1.1: Use Python language to demonstrate understanding of performance parameters, validation strategies and concepts of underfitting-overfitting with the given resources. (PI 1.1.3, 1.4.1, 5.1.2, 5.2.2)* *LO 1.2: Interpret the write-up instructions correctly for performing the experiment and present a report on the experiments performed with proper interpretations of results and conclusion, following technical writing ethics. (PI 7.1.1, 9.1.1, 9.1.2)*	
02.	**Supervised Learning Techniques** ***Learning Objectives:*** *To teach various popular supervised learning techniques and develop understanding of their usage.*	**08**
	Demonstrations on: **1.** Available datasets for various purposes, their utility and the ethics for downloading the datasets. **2.** Learning Model Building in Scikit-learn	
	Suggested Tasks: (One from each part A, B and C) **A. Linear and Logistic regression:** 1. As an ML engineer, you have to perform housing price prediction from the historical data available for a locality. Implement linear regression on the Boston Housing dataset, perform prediction for test inputs with sklearn library, visualize the linear regression line and evaluate the model performance with R-square score. 2. As an ML engineer, you have to assist medical professional in early detection of type-II diabetes. Develop your own model for logistic regression and compare its performance with sklearn model on the diabetes dataset. **B. Neural Networks:** 1. An archeologist would like assistance from a ML engineer to convert handwritten scripts to typed documents, for preservation. Develop your own three-layer neural network to perform handwritten digit recognition 2. You have to demonstrate non-linearly separable ML problems to an educational professional. Develop own multi-layer neural network and implement back-propagation algorithm for solving the X-OR problem. **C. Other algorithms:** 1. Some rare species of flowers required to be classified for the purpose of preserving the species, by the forest department. You are provided with one such dataset of rare species: The Iris dataset. Implement KNN algorithm	

	using scikit learn to perform predictions on the iris dataset and plot the accuracy for train, test data vs. number of neighbors.	
	Self-Learning Topics: Visualization Libraries like Seaborn and its use for visualization of data	
	Learning Outcomes: *A learner will be able to* *LO 2.1: Use modern libraries from Python, and utilize them for implementing supervised learning algorithms on provided datasets. (PI 1.4.1, 2.1.1, 5.1.1, 5.3.1)* *LO 2.2: Interpret results obtained correctly to establish the relationship of the results with the theoretical principles through a logically progressing Python program and convey in a well- documented report. (PI 4.1.4, 4.3.4, 9.1.2, 9.1.3)* *LO 2.3: Practice the ethical principles in using the datasets and report the same in the citation section of the document on the experiments. (PI 7.1.1, 7.1.2)*	
03.	**Unsupervised Learning Techniques & Dimensionality Reduction** ***Learning Objectives:*** *1) To develop understanding about need for unsupervised learning and dimensionality reduction.* *2) To teach basic unsupervised learning and DR techniques.*	**06**
	Demonstrations on: Data pre-processing and normalization methods, packages	
	Suggested Tasks: (Any Two) 1. An educational Institute requires classification of its students among those who are good academic performers, those who have good art skills, those who have good management skills etc. You are provided such data without any labels. For the given number of features, samples and initial centroids, implement K-means clustering algorithm to add new data-points and visualize the shift in centroids. 2. A medical professional requires assistance in categorizing the cancer images, before sending it to expert for an opinion. Perform classification of benign and malignant cancer images from the publicly available cancer dataset employing SVM algorithm. 3. In Module 2-C, you have done Iris dataset classification using KNN. In this module, implement PCA and then perform Logistic regression in the transformed feature-space for the dataset and report your interpretations about utility of DR technique applied in your experiment.	
	Self-Learning Topics*: kernel method and non-linear hyperplanes in SVM*	
	Learning Outcomes: *A learner will be able to* *LO 3.1: Demonstrate proficiency in using suitable Python libraries to implement given unsupervised learning technique using efficient coding techniques in Python. (PI 5.2.2, 5.3.3)* *LO 3.2: Interpret results obtained correctly to establish the relationship of the results with the theoretical principles through a logically progressing Python program and convey in a well- documented report. (PI 4.1.4, 4.3.4, 9.1.2, 9.1.3)*	
04.	**Applications of ML in Communication Systems** ***Learning Objective/s:*** *Review concepts of communication systems that invite intervention of ML techniques and establish connections between requirements in communication & machine learning.*	**04**
	Suggested Tasks:	

	Present a case-study on one of the following topics identifying resources, exhibiting self-learning abilities, and team-spirit following IEEE formats of presentation and ethics (Ref: IEEE - IEEE Code of Ethics) 1. Applications of supervised learning in modulation classification, adaptive modulation, and coding mechanisms for wireless systems 2. Use of principal component analysis in massive MIMO system design, auto encoders in wireless communication transceiver design.	
	Learning Outcomes: *A learner will be able to* *LO 4.1: Identify relevant resources for the case-study, appreciate impact of ML in sustainable design of communication systems and highlight that in the group presentation effectively. (PI 6.4.1, 8.3.1, 9.1.2)* *LO 4.2: Follow ethical standards expected from an Engineer and convey the relevance in the presented case-study (PI 7.1.1, 7.2.1)*	
05.	**Course Project** ***Learning Objective/s:*** *To provide a guided channel for learner to employ the techniques learnt in the course to a problem related to engineering applications.*	**06**
	Learners are expected to complete and demonstrate a group course project on a topic relevant to society/Industry; from the suggested (but not limited to) list and approved by the instructor and present the course project report in IEEE format following all ethical practices.	
	Suggested List of Course Projects: 1. **Service Industry**: Customer churn prediction using Decision tree & Random Forest 2. **Bike Sharing**: Prediction of bike rental count hourly or daily based on the environmental and seasonal settings 3. **Stock Prediction**: Time Series Forecasting (LSTM) and Prediction Curve 4. **Healthcare**: ECG Heart Beat Analysis, Visualization and Heart Beat Classification using ANN 5. **Governance**: Classification of Devanagari Handwritten Characters using Neural Network 6. **Security: and surveillance**: Face Recognition using Computer Vision and Deep Learning 7. **Marketing strategies**: Amazon Product Reviews Sentiment Analysis 8. **Delivery agent services**: Zomato Restaurant Reviews Sentiment Analysis using LSTM 9. **Societal application**: Object detection and tracking using YOLO 10. **Banking sector**: Developing Conventional Chatbot using Open AI, GPT	
	Learning Outcomes:	
	A learner will be able to *LO 5.1: Define a problem, its scope, and importance for purposes of investigation and examine relevant tools for carrying out the project. (PI 4.1.1, 4.1.2)* *LO 5.2: Design and develop an experimental approach to solve the use-case establishing its relevance and impact in the larger interest of society. (PI 4.2.1, 6.1.1)* *LO 5.3: Analyze the data trends, correlations and represent the data graphically leading to choice of appropriate techniques. (PI 4.3.2, 4.3.3)* *LO 5.4: Apply principles of preventive engineering to propose efficient solutions through time-and-power-efficient programming method. (PI 5.1.2, 6.4.1)*	

	LO 5.5: Carry out the project work with equitable distribution of work highlighting individual contributions and resolving conflicts if any. (PI 8.1.2) *LO 5.6: Adhere to ethical practices in carrying out literature survey, programming and presentation with proper citation and credits. (PI 7.1.1, 7.1.2)* *LO 5.7: Present the course project findings in a well-documented report as well as oral presentation in an effective manner. (PI 9.1.1, 9.1.2, 9.1.3, 9.2.2)*	
	Course Conclusion	**01**
	Total	**30**
One experiment on Module-I, one on each of part A, B & C from Module-II, two experiments from Module-II, one case-study and one course project has to be completed by the learner in the allotted time frame.		

Performance Indicators:

P.I. No.	**P.I. Statement**
1.1.3	Apply advanced mathematical techniques of computer and information science to describe/solve/construct a mathematical model of a system.
1.4.1	Apply concepts of electronics and communication engineering and **allied disciplines** to solve engineering problems.
2.1.1	Articulate problem statements and identify primary objectives and key constraints.
4.1.1	Define a problem, its scope, and importance for purposes of investigation
4.1.2	Examine the relevant methods, tools, and techniques of experiment, design, system calibration, data acquisition, analysis and presentation
4.2.1	Apply the concepts of statistical design of experiments and choose an appropriate experimental design plan based on the study objectives.
4.3.1	Use appropriate procedures, tools, and techniques to conduct experiments and collect data
4.3.4	Synthesize information and knowledge about the problem from the data to reach appropriate conclusions
5.1.1	Identify modern hardware and software engineering tools, techniques and resources for engineering activities.
5.1.2	Use/adapt/modify/create tools and techniques to solve engineering problems
5.2.2	Demonstrate proficiency in using discipline-specific tools.
5.3.1	Recognize the assumptions and simplifications used in a model or a simulation and their impact on the results.
5.3.3	Recognize sources of error in measurements, modelling or simulations and verify credibility of results.
6.1.1	Identify social, environmental and health related issues with the help of relevant viewpoints from stakeholders and propose engineering solutions.
6.4.1	Describe and evaluate engineering principles for sustainable development pertaining to relevant fields.
7.1.1	Follow ethical practices to create a document.
7.1.2	Recognize and articulate the issues involved in ethical case studies.

8.1.2 Implement the norms of practice (e.g. rules, roles, charters, agendas, etc.) of effective teamwork, to accomplish a goal.

8.3.1 Present results as a team, with smooth integration of contributions from all individual efforts

9.1.1 Read, understand and interpret technical and/or non-technical information.
9.1.2 Create clear, well-constructed, and well-supported written engineering documents and/or presentation.
9.1.3 Create flow in a document or presentation - a logical progression of ideas so that the main point is clear.

Course Outcomes: A learner will be able to -

1. Write Python programs for evaluation of performance of a ML algorithm. *(LO 1.1, 1.2)*
2. Use modern libraries in Python to implement supervised/unsupervised learning algorithms in an efficient manner on given datasets. *(LO 2.1, 3.1)*
3. Interpret results correctly and establish relationship with underlying algorithm/s in a well-documented lab report following ethical principles. *(LO 1.2, 2.2, 2.3, 3.2)*
4. Exhibit competency to identify relevant resources for case studies in ML for sustainable design of communication systems and present the same as a collective effort. *(LO 4.1, 4.2)*
5. Perform and present a group project related to application of ML in societal/health-care/public safety area following engineering ethics. *(LO 5.1 to 5.7)*

CO-PO Mapping Table with Correlation Level

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CEMDM604.1	3	-	-	-	3	-	2	-	3	-	-
CEMDM604.2	2	3	-	-	3	-	-	-	-	-	-
CEMDM604.3	-	-	-	3	-	-	3	-	3	-	-
CEMDM604.4	-	-	-	-	-	2	3	2	2	-	-
CEMDM604.5	-	-	-	3	3	3	3	2	3	-	-
Average	**3**	**3**	**-**	**3**	**3**	**3**	**3**	**2**	**3**	**-**	**-**

Text Books :

1. Andreas C. Muller and Sarah Guido, Introduction to Machine Learning with Python, O'Raily Publication
2. Francois Chollet, Deep Learning with Python (Second Edition), Manning Pubns Co.
3. Ian Goodfellow, Yoshua Bengio,Aaron Courville, Deep Learning, MIT Press, 2016

Reference Books:

1. Christopher M Bishop, Neural Networks For Pattern Recognition, Oxford University Press

Other Resources:

1. NPTEL course: Introduction to Machine Learning (in Hindi) by Prof. Anubha Gupta, IIT Delhi, available at: NPTEL

A. CONTINUOUS ASSESSMENT (25 MARKS)

Suggested breakup of distribution

a. Practical Exercises & Case study- 10 Marks

Students will be given number of experiments as mentioned in the syllabus. Each experiment carries 10 Marks. Average will be taken of all experiments. Students are expected to follow instructions in the write-up and perform the assigned tasks in the experiments as per requirement. They are expected to prepare a brief report of each experiment, which includes the task description, program, results obtained, discussion on results and conclusion.

b. Course Project Presentation: 15 marks

The course project will have to be demonstrated and will be evaluated as per the following guideline.

- Group assessment:

i. Quality of literature survey and ethics: 20%

ii. Clarity in conceptual knowledge and approach: 20%

iii. Presentation skills & ethics: 10%

iv. Technical competency and working project demonstrated as a group: 30%

- Individual Assessment:

i. Clearly stated individual contribution: 10%

ii. Responding to questions: 10%

B. END SEMESTER ASSESSMENT (25 Marks)

Practical Examination: 15 marks

A laboratory examination based upon on the tasks performed in Module 1, 2 and 3. Inter-linkage of tasks from different modules can be asked as the problem statement in ESE.

Example: Given a dataset of brain-tumour images, perform classification of the images as “malignant” and “benign” using appropriate ML technique. Evaluate performance of the algorithm using suitable metric.

Viva-voce on the course-modules: 10 marks

Two examiners, one Internal and one External will do the ESE evaluation.