DAYANANDA SAGAR UNIVERSITY

SCHOOL OF ENGINEERING

SCHEME & SYLLABUS

FOR

BACHELOR OF TECHNOLOGY (B.Tech) – I YEAR

COMMON TO ALL BRANCHES

(With effect from 2024-25)

Devarahaggalahalli , Harohalli,
Kanakapura Main Road,
Bengaluru South – 562 112

SCHEME - B.TECH – 2024-25 ONWARDS

I SEM - CHEMISTRY CYCLE

SL	COURSE CODE	COURSE TITLE	SCHEME OF TEACHING					Duration in Hrs Exam	Examination (Maximum Marks)		
			L	T	P	J	C		CIE	SEE	TM
1	24EN1101	LINEAR ALGEBRA AND DIFFERENTIAL EQUATIONS	3	0	0	0	3	3	60	40	100
2	24EN1102	OBJECT ORIENTED PROGRAMMING	2	1	2	0	4	3	60	40	100
3	24EN1103	COGNITIVE AND TECHNICAL SKILLS – I	0	0	4	0	2	1	100	-	100
4	24EN1104	ENGINEERING CHEMISTRY	2	0	2	0	3	3	60	40	100
5	24EN1105	INTRODUCTION TO MECHANICAL ENGINEERING	2	0	2	0	3	3	60	40	100
6	24EN1106	INTRODUCTION TO ELECTRICAL ENGINEERING	2	0	0	0	2	3	60	40	100
7	24EN1107	ENGINEERING MECHANICS	2	0	0	0	2	3	60	40	100
8	24EN1108	TECHNICAL ENGLISH	2	0	0	0	2	1	100	-	100
9	24EN1109	ENVIRONMENTAL SCIENCE	1	0	0	0	0	1	50	-	50
10	24EN1110	KANNADA KALI / MANASU	1	0	0	0	0	1	50	-	50
			17	1	10	0	21		660	240	900

L – Lecture, T – Tutorial, P – Practical, J – Project, C – No. of Credits, CIE – Continuous Internal Evaluation,
SEE- Semester End Examinations, TM – Total Marks.

SCHEME - B.TECH – 2024-25 ONWARDS

I SEM - PHYSICS CYCLE

SL	COURSE CODE	COURSE TITLE	SCHEME OF TEACHING					Duration in Hrs Exam	Examination (Maximum Marks)		
			L	T	P	J	C		CIE	SEE	TM
1	24EN1101	LINEAR ALGEBRA AND DIFFERENTIAL EQUATIONS	3	0	0	0	3	3	60	40	100
2	24EN1102	OBJECT ORIENTED PROGRAMMING	2	1	2	0	4	3	60	40	100
3	24EN1103	COGNITIVE AND TECHNICAL SKILLS – I	0	0	4	0	2	1	100	-	100
4	24EN1111	ENGINEERING PHYSICS	3	0	2	0	4	3	60	40	100
5	24EN1112	INTRODUCTION TO ELECTRONICS ENGINEERING	3	0	0	0	3	3	60	40	100
6	24EN1113	ENGINEERING GRAPHICS AND DESIGN THINKING	2	0	2	0	3	2	60	40	100
7	24EN1114	BIOLOGY FOR ENGINEERS	2	0	0	0	2	1	100	-	100
8	24EN1115	CONSTITUTION OF INDIA AND PROFESSIONAL ETHICS	1	0	0	0	0	1	50	-	50
			16	1	10	0	21		550	200	750

L – Lecture, T – Tutorial, P – Practical, J – Project, C – No. of Credits, CIE – Continuous Internal Evaluation,
SEE- Semester End Examinations, TM – Total Marks.

SCHEME - B.TECH – 2024-25 ONWARDS

II SEM - CHEMISTRY CYCLE

SL	COURSE CODE	COURSE TITLE	SCHEME OF TEACHING					Duration in Hrs Exam	Examination (Maximum Marks)		
			L	T	P	J	C		CIE	SEE	TM
1	24EN1201	SINGLE AND MULTIVARIATE CALCULUS	3	0	0	0	3	3	60	40	100
2	24EN1202	C PROGRAMMING FOR PROBLEM SOLVING	2	1	2	0	4	3	60	40	100
3	24EN1203	COGNITIVE AND TECHNICAL SKILLS – II	0	0	4	0	2	1	100	-	100
4	24EN1104	ENGINEERING CHEMISTRY	2	0	2	0	3	3	60	40	100
5	24EN1105	INTRODUCTION TO MECHANICAL ENGINEERING	2	0	2	0	3	3	60	40	100
6	24EN1106	INTRODUCTION TO ELECTRICAL ENGINEERING	2	0	0	0	2	3	60	40	100
7	24EN1107	ENGINEERING MECHANICS	2	0	0	0	2	3	60	40	100
8	24EN1108	TECHNICAL ENGLISH	2	0	0	0	2	1	100	-	100
9	24EN1109	ENVIRONMENTAL SCIENCE	1	0	0	0	0	1	50	-	50
10	24EN1110	KANNADA KALI / MANASU	1	0	0	0	0	1	50	-	50
			17	1	10	0	21		660	240	900

L – Lecture, T – Tutorial, P – Practical, J – Project, C – No. of Credits, CIE – Continuous Internal Evaluation,
SEE- Semester End Examinations, TM – Total Marks.

SCHEME - B.TECH – 2024-25 ONWARDS

II SEM - PHYSICS CYCLE

SL	COURSE CODE	COURSE TITLE	SCHEME OF TEACHING					Duration in Hrs Exam	Examination (Maximum Marks)		
			L	T	P	J	C		CIE	SEE	TM
1	24EN1201	SINGLE AND MULTIVARIATE CALCULUS	3	0	0	0	3	3	60	40	100
2	24EN1202	C PROGRAMMING FOR PROBLEM SOLVING	2	1	2	0	4	3	60	40	100
3	24EN1203	COGNITIVE AND TECHNICAL SKILLS – II	0	0	4	0	2	1	100	-	100
4	24EN1111	ENGINEERING PHYSICS	3	0	2	0	4	3	60	40	100
5	24EN1112	INTRODUCTION TO ELECTRONICS ENGINEERING	3	0	0	0	3	3	60	40	100
6	24EN1113	ENGINEERING GRAPHICS AND DESIGN THINKING	2	0	2	0	3	2	60	40	100
7	24EN1114	BIOLOGY FOR ENGINEERS	2	0	0	0	2	1	100	-	100
8	24EN1115	CONSTITUTION OF INDIA AND PROFESSIONAL ETHICS	1	0	0	0	0	1	50	-	50
			16	1	10	0	21		550	200	750

L – Lecture, T – Tutorial, P – Practical, J – Project, C – No. of Credits, CIE – Continuous Internal Evaluation,
SEE- Semester End Examinations, TM – Total Marks.

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

SCHEME & SYLLABUS FOR BACHELOR OF TECHNOLOGY (B. Tech)

COMPUTER SCIENCE & ENGINEERING (Artificial Intelligence and Data Science)

(3rd to 4th Semester)

With effect from 2024

B.Tech. PROGRAMME– 2024-28 BATCH

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

III-SEMESTER													
S.NO	Course Type	Course Code	Course Title	Teaching Department	Teaching Hours / Week				Examination				Credits
					Lecture	Tutorial	Practical	Project	Contact Hours	CIE Marks	SEE Marks	Total Marks	
					L	T	P	J					
1	BSC	24AD2301	PROBABILITY AND STATISTICS	MAT	3	0	0	0	3	60	40	100	**3**
2	IPCC	24AD2302	DATA STRUCTURES	CSE (AI&DS)	3	0	2	0	5	60	40	100	**4**
3	IPCC	24AD2303	DIGITAL LOGIC DESIGN	ECE	3	0	2	0	5	60	40	100	**4**
4	IPCC	24AD2304	DATABASE MANAGEMENT SYSTEMS	CSE (AI&DS)	3	0	2	0	5	60	40	100	**4**
5	PCC	24AD2305	DATA ANALYTICS AND VISUALIZATION	CSE (AI&DS)	3	0	0	0	3	60	40	100	**3**
6	PCC	24AD2306	ARTIFICIAL INTELLIGENCE	CSE (AI&DS)	3	0	0	0	3	60	40	100	**3**
7	SEC	24AD23XX	SKILL ENHANCEMENT COURSE – I	CSE (AI&DS)	1	0	2	0	3	100	--	100	**2**
8	TPC	24AD2307	COGNITIVE AND TECHNICAL SKILLS – III	TPO	-	-	-	-	-	-	-	-	-
		TOTAL			**19**	**0**	**8**	**0**	**27**	**460**	**240**	**700**	**23**

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

Skill Enhancement Course I	
Course Code	**Course Name**
24AD2308	MATLAB FOR DATA SCIENCE
24AD2309	LINUX PROGRAMMING AND SCRIPTING
24AD2310	BI USING TABLEAU / POWER BI

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

IV-SEMESTER													
S. No	Course Type	Course Code	Course Title	Teaching Department	Teaching Hours / Week				Examination				Credits
					Lecture	Tutorial	Practical	Project	Contact Hours	CIE Marks	SEE Marks	Total Marks	
					L	T	P	J					
1	BSC	24AD2401	TRANSFORMS AND NUMERICAL TECHNIQUES	MAT	3	0	0	0	3	60	40	100	**3**
2	IPCC	24AD2402	DESIGN AND ANALYSIS OF ALGORITHMS	CSE (AI&DS)	3	0	2	0	5	60	40	100	**4**
3	IPCC	24AD2403	OPERATING SYSTEMS	CSE (AI&DS)	3	0	2	0	5	60	40	100	**4**
4	PCC	24AD2404	APPLIED MACHINE LEARNING	CSE (AI&DS)	3	0	2	0	5	60	40	100	**4**
5	IPCC	24AD2405	COMPUTER ORGANIZATION AND ARCHITECTURE	CSE (AI&DS)	3	0	0	0	3	60	40	100	**3**
6	PCC	24AD2406	BUSINESS ESSENTIALS FOR ENGINEERS	CSE (AI&DS)	2	0	0	0	2	60	40	100	**2**
7	AEC	24LSXXXX	LIBERAL STUDIES - I	AEC	1	0	0	0	1	50	--	50	**1**
8	SEC	24AD24XX	SKILL ENHANCEMENT COURSE - II	CSE (AI&DS)	1	0	2	0	3	100	--	100	**2**
9	TPC	24AD2407	COGNITIVE AND TECHNICAL SKILLS – IV	TPO	–	–	–	–	–	–	–	–	–
		TOTAL			**19**	**0**	**8**	**0**	**27**	**510**	**240**	**750**	**23**

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

Skill Enhancement Course II	
Course Code	**Course Name**
24AD2408	R PROGRAMMING
24AD2409	DJANGO FRAMEWORK
24AD2410	PROGRAMMING IN JAVA

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

PROBABILITY AND STATISTICS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Course Code	: 24AD2301	**Credits**	: 03
Hours / Week	: 03 Hours	**Total Hours**	: 39 Hours
L–T–P–J	: 3–0–0–0		

Course Learning Objectives:

This Course will enable students to:

1. **Apply** statistical principles and probability concepts to solve complex problems in real-world scenarios involving uncertainty and randomness.
2. **Evaluate** and select appropriate probability distributions and statistical techniques to analyze and interpret data accurately in various applications.
3. **Justify** the use of estimation methods and hypothesis testing techniques for drawing meaningful inferences about population parameters.
4. **Analyze** and interpret sample test results for different statistical relationships, such as means, variances, correlation coefficients, regression coefficients, goodness of fit, and independence, to make informed decisions.
5. **Identify** sample tests using appropriate statistical procedures to investigate the significance of observed data and communicate findings effectively.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture method, but different *type of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

MODULE – I: Probability	**09 Hours**

Definitions of Probability, Addition Theorem, Conditional Probability, Multiplication Theorem, Bayes' Theorem of Probability	
MODULE – II: Random Variables and their Properties and Probability Distributions	**09 Hours**
Discrete Random Variable, Continuous Random Variable, Joint Probability Distributions Their Properties, Probability Distributions: Discrete Distributions: Binomial, Poisson Distributions and their Properties; Continuous Distributions: Exponential, Normal, Distributions and their Properties.	
MODULE – III: Estimation and testing of hypothesis	**06 Hours**
Sample, Populations, Statistic, Parameter, Sampling Distribution, Standard Error, Un-Biasedness, Efficiency, Maximum Likelihood Estimator, Notion & Interval Estimation.	
MODULE – IV: Sample Tests-1	**07 Hours**
Large Sample Tests Based on Normal Distribution, Small Sample Tests: Testing Equality of Means, Testing Equality of Variances, Test of Correlation Coefficient	
MODULE – V: Sample Tests-2	**08 Hours**
Test for Regression Coefficient; Coefficient of Association, 2 – Test for Goodness of Fit, Test for Independence.	

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Apply** the principles of probability to solve complex problems in various real-world scenarios.	L2 & L3
2	**Solve** and **compare** different probability distributions, including discrete and continuous random variables, in order to make informed decisions and predictions.	L2 & L3
3	**Apply** statistical estimation techniques, such as maximum likelihood estimation and interval estimation, to draw meaningful inferences about population parameters from sample data.	L3
4	**Examine** hypothesis testing methods, including large and small sample tests, to assess the significance of observed data and draw valid conclusions.	L4
5	**Analyze** statistical relationships and perform sample tests to assess the Equality of means in different populations, Correlation coefficients between variables to determine the strength and direction of the relationship. Independence of variables using appropriate statistical tests to assess the absence of any relationship.	L4

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	3	2	2		2				1					
CO2	3	2	2		2				1					
CO3	3	2	2						1					
CO4	3	2	2		2				1					
CO5	3	2	2		2				1					

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Probability & Statistics for Engineers and Scientists, Walpole, Myers, Myers, Ye. Pearson Education.

REFERENCE BOOKS:

1. Probability, Statistics and Random Processes T. Veerarajan Tata McGraw – Hill
2. Probability & Statistics with Reliability, Queuing and Computer Applications, Kishor S. Trivedi, Prentice Hall of India ,1999

E-Resources:

1. https://nptel.ac.in/courses/106104233
2. https://nptel.ac.in/courses/117103067
3. https://nptel.ac.in/courses/103106120
4. https://www.coursera.org/learn/probability-intro#syllabus
5. https://nptel.ac.in/courses/111104073

Activity Based Learning (Suggested Activities in Class)

1. Tools like Python programming, R programming can be used which helps student to develop a skill to analyze the problem and providing solution.
2. Regular Chapter wise assignments/ Activity/Case studies can help students to have critical thinking, developing an expert mind set, problem-solving and teamwork.

Following are Assignments/ Activities Can be carried out using either R programming language or Python Programming or excel solver.

1. There are n people gathered in a room. What is the probability that at least 2 of them will have the same birthday? (Use excel solver, R Programming, Python Programming)
   a. Use simulation to estimate this for various n., and Produce Simulation Graph.
   b. Find the smallest value of n for which the probability of a match is greater than 0.5.
   c. Explore how the number of trials in the simulation affects the variability of our estimates.
2. **Case Study 1: Customer Arrivals at a Coffee Shop**
   a. A coffee shop wants to analyze the number of customer arrivals during its morning rush hour (7:00 AM to 9:00 AM). The shop has been recording the number of customer arrivals every 15 minutes for the past month.
   b. Data: The data consists of the number of customer arrivals recorded at the coffee shop during each 15-minute interval for the past month.

c. Here is a sample of the data:

Time Interval	Customer Arrivals
00 AM - 7:15 AM	**6**
15 AM - 7:30 AM	**4**
30 AM - 7:45 AM	**9**
45 AM - 8:00 AM	**7**
00 AM - 8:15 AM	**5**
15 AM - 8:30 AM	**8**
30 AM - 8:45 AM	**10**
45 AM - 9:00 AM	**6**

analyze the customer arrivals and determine the probability distribution that best fits the data. Specifically, explore both discrete and continuous probability distributions, including the binomial, Poisson, exponential, and normal distributions.

3. **Case Study 2: Comparing the Performance of Two Groups**

a. Suppose you are a data analyst working for a company that manufactures a new energy drink. The marketing team conducted a promotional campaign in two different cities (City A and City B) to determine the effectiveness of the campaign in increasing sales. The sales data for a random sample of customers in each city was collected over a week. Your task is to compare the average sales between the two cities and test whether there is a significant difference in the variance of sales.

b. **Data**: Let's assume the following sample data for the number of energy drinks sold in each city:

City A: [30, 28, 32, 29, 31, 33, 34, 28, 30, 32]
City B: [25, 24, 26, 23, 22, 27, 29, 30, 26, 24]

perform a two-sample t-test to test the equality of means and a test for equality of variances using Python's SciPy library.

1. **case study 3:** testing independence between two categorical variables.

a. Data: Sample of 100 employees, and each employee is classified as either Male or Female. They were asked to rate their job satisfaction on a scale of 1 to 5, where 1 represents low satisfaction and 5 represents high satisfaction. The data is as follows:

Employee	Gender	Job Satisfaction
1	Male	4
2	Female	3
3	Male	2
4	Female	5
...	...	...
100	Female	4

b. Test for independence between gender and job satisfaction, use the chi-squared test in R.

****

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

DATA STRUCTURES

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Course Code	: 24AD2302	**Credits**	: 04
Hours / Week	: 05 Hours	**Total Hours**	: 39(Th)+26(P) Hours
L–T–P-J	: 3–0–2-0		

Prerequisites:

Proficiency in a C programming language.

Course Learning Objectives:

This Course will enable students to:

1. **Understand** the basic approaches for analysing and designing data structures.
2. **Introduce** dynamic memory allocation and C language concepts required for building data structures
3. **Develop** essential skills to construct data structures to store and retrieve data quickly and **efficiently**.
4. **Utilize** different data structures that support different sets of operations which are suitable for various applications.
5. **Explore & Implement** how to insert, delete, search and modify data in any data structure-Stack, Queues, Lists, Trees.
6. **Develop** applications using the available data structure as part of the course for mini-project.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.

3. Show ***Video/Animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

MODULE – I	**08 Hours**

INTRODUCTION:

Introduction to Data Structure, Classification, C Structure and Union, Array Definition, Representation, Operations (Insertion, Deletion, Search and Traversal), Two/Multidimensional Arrays, sparse matrix, C Pointers

MODULE – II	**09 Hours**

INTRODUCTION TO ADT:

Stack: Definition, Array Representation of Stack, Operations on Stacks.

Applications of Stack: Expression evaluation, Conversion of Infix to Postfix, Infix to Prefix Recursion, Tower of Hanoi

Queue: Definition, Representation of Queues, Operations of Queues, Circular Queue.

Applications of Queue: Job Scheduling, A Maze Problem

MODULE – III	**09 Hours**

DYNAMIC DATA STRUCTURES:

Linked List: Types, Representation of Linked Lists in Memory. Traversing, Searching, Insertion & Deletion from Linked List. Circular List, Doubly Linked List, Operations on Doubly Linked List (Insertion, Deletion, Traversal).

Applications: Stack & Queue Implementation using Linked Lists.

Case Study: Josephus problem.

MODULE – IV	**08 Hours**

TREES:

Basic Terminology, Binary Trees and their representation, Complete Binary Trees, Binary Search Trees, Threaded Binary Trees, Operations on Binary Trees (Insertion, Deletion, Search & Traversal).

Applications: Expression Evaluation

Case Study: Game Tree

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

MODULE – V	**05 Hours**
Efficient Binary Search Trees: Optimal Binary Search Trees, AVL Trees, Red Black Trees, Splay Trees. **Case Study**: B Trees	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Demonstrate** the key C programming concepts such as pointers, structures, unions and arrays data structures to perform operations such as insertion, deletion, searching, sorting, and traversing.	L3
2	**Utilize** the fundamental concepts of stacks and queues to solve the standard applications like tower of Hanoi, conversion and evaluation of expressions, job scheduling and maze.	L3
3	**Implement** Singly Linked List, Doubly Linked List, Circular Linked Lists, stacks and queues using linked list**.**	L3
4	**Develop** critical thinking and problem-solving skills by designing and implementing efficient algorithms for Non-linear tree data structure and perform insertion, deletion, search and traversal **operations** on it.	L3
5	**Apply** advanced techniques, such as balancing algorithms for AVL trees, Splay trees and Red-Black trees to maintain the balance and efficiency of binary trees.	L3

Table: Mapping Levels of COs to POs / PSOs														
COs	**Program Outcomes (POs)**											**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO1	3	2	2		2			1				3	2	2
CO2	3	3	3	2	2	1		1			2	3	3	2
CO3	3	3	3	2	3						2	3	2	3
CO4	3	3	3	3	3	1	1	1	1	1	2	3	3	2
CO5	3	3	3	3	3			1				3	2	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. A.M. Tannenbaum, Y Langsam, M J Augentien "Data Structures using C", 1st Edition, Pearson, 2019.
2. Ellis Horowitz, Susan Anderson-Freed, and Sartaj Sahni, "Fundamentals of Data structures in C", 2nd Edition, Orient Longman, 2008.

REFERENCE BOOKS:

1. Brian. W. Kernighan, Dennis. M. Ritchie, "The C Programming Language", 2nd Edition, Prentice-Hall, 1988.
2. Gilbert & Forouzan, "Data Structures: A Pseudo-code approach with C", 2nd Edition, Cengage Learning,2014.
3. Jean-Paul Tremblay & Paul G. Sorenson, "An Introduction to Data Structures with Applications", 2nd Edition, McGraw Hill, 2013.
4. R.L. Kruse, B.P. Learly, C.L. Tondo, "Data Structure and Program design in C", 5th Edition, PHI ,2009.

E-Resources:

1. https://nptel.ac.in/courses/106102064
2. https://www.coursera.org/learn/data-structures?specialization=data-structures- algorithms
3. https://www.udemy.com/topic/data-structures/free/
4. https://www.mygreatlearning.com/academy/learn-for-free/courses/data-structures
5. https://cse01-iiith.vlabs.ac.in/

6. https://kremlin.cc/k&r.pdf

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving using group discussion.
2. Role play E.g., Stack, Queue, etc.,
3. Demonstration of solution to a problem through programming.
4. Flip class activity E.g., arrays, pointers, dynamic memory allocation, etc.,

LABORATORY EXPERIMENTS

Following are experiments to be carried out using either C programming language

1. To Implement C programs with concepts of pointers, structures.
2. To implement multidimensional array Matrix Multiplication.
3. To search elements in data structure with different search methods.
4. To implement stack, queue and their variations using arrays.
5. To implement stack, queue and their variations using singly linked lists
6. To implement conversion & evaluation of expression using stacks.
7. To implement doubly circular Linked Lists and variations and use them to store data and perform operations on it.
8. To implement Addition/multiplication of 2 polynomial using linked lists
9. To implement binary tree traversal techniques.

OPEN-ENDED EXPERIMENTS

1. A man in an automobile search for another man who is located at some point of a certain road. He starts at a given point and knows in advance the probability that the second man is at any given point of the road. Since the man being sought might be in either direction from the starting point, the searcher will, in general, must turn around many times before finding his target. How does he search to minimize the expected distance travelled? When can this minimum expectation be achieved?
2. The computing resources of a cloud are pooled and allocated according to customer demand. This has led to increased use of energy on the part of the service providers due to the need to maintain the computing infrastructure. What data structure will you use

for allocating resources which addresses the issue of energy saving? Why? Design the solution.

3. Mini-Project on applying suitable data structure to a given real-world problem.

****

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

DIGITAL LOGIC DESIGN

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Course Code	: 24AD2303	**Credits**	: 04
Hours / Week	: 05 Hours	**Total Hours**	: 39(Th) + 26(P) Hours
L–T–P-J	: 3–0–2-0		

Course Learning Objectives:

This Course will enable students to:

1. **Translate** the elements of digital logic functions to digital system abstractions using Verilog.
2. **Illustrate** simplification of Boolean expressions using Karnaugh
3. **Model** combinational logic circuits for arithmetic operations and logical operations
4. **Analyse** and model sequential elements flip-flops, counter, shift registers.
5. **Outline** the concept of Mealy Model, Moore Model and apply FSM to solve a given design problem.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture method, but different *type of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.

8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

MODULE – I	**08 Hours**

INTRODUCTION:

Number System- Binary, Hexa, Decimal, Octal and its conversion. Canonical Notation - SOP & POS forms, Minimization of SOP and POS forms.

ARITHMETIC CIRCUITS AND VERILOG MODELLING

Adders: Half adder, full adder, Ripple carry adder, parallel adder /subtractor, fast adders-CLA, comparator- 2 bit. Simplification using K-Maps

Introduction to Verilog, Syntax of Verilog coding, Modelling styles in Verilog, Verilog Operators, Test bench for simulation

MODULE – II	**07 Hours**

Combinational Circuit Building Multiplexers 4:1, 8:1, decoders 3:8, 2:4, demultiplexers 1:4, encoders 8:3, 4:2, code converters- B to G and G to B- Simplification using K-Maps

Verilog for combinational circuits, if else, case-casex, casez, for loop, generate.

MODULE – III	**08 Hours**

Sequential Circuits-1

Basic Latch, Gated latches, Flip Flops SR, D, JK, T, master-slave flip-flops JK, Characteristic equations, 0's and 1's Catching Problem, Race round condition, Switch debounce, shift registers- SISO, SIPO, PISO, PIPO, Setup time, Hold time, Propagation Delay

MODULE – IV	**08 Hours**

Sequential Circuits-2

Binary counters – asynchronous and synchronous, mod-n counter, ripple counter- 4 bit.

Verilog blocking and non-blocking,

Mealy Model, Moore Model, State machine notation, Construction of Finite State Machine.

MODULE – V	**08 Hours**

Introduction to Electronic Design Automation:

FPGA Design Flow, ASIC Design flow, architectural design, logic design, simulation, verification and testing, 3000 Series FPGA architecture.

Applications:

Design 4 Bit ALU, 7 Segment display, Vending Machine, 3 Pipeline.

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

Laboratory Experiments	
Experiments are conducted using Verilog tool /Kits	
1.	Introduction to Xilinx tool, FPGA flow
2.	Adder – HA, FA using data flow and behaviour modelling styles
3.	Adder – HA, FA using structural modelling style
4.	Combinational designs – I (blocking and non-blocking/looping examples) **a.** Multiplexer: 4:1, 8:1 MUX. **b.** De Multiplexer: 1:4, 1:8 DEMUX.
5.	Combinational designs – II (different types of case statements) **c.** Encoder with and without Priority: 8:3 and 4:2. **d.** Decoder: 3:8 and 2:4.
6.	Design of 4-bit ALU
7.	Flip Flop: D FF, T FF, JK FF
8.	Design of Mod – n Up/Down Counter with Synchronous reset
9.	Design of Mod – n Up/Down Counter with Asynchronous reset.
10.	Design of Universal shift Register using FSM

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Interpret Boolean Expressions of digital design in simplified form	L2
2	Build the various elements of digital logic system with Verilog	L3
3	Construct Combinational and Sequential logic circuits	L3
4	Analyse the hardware model of a digital system at different levels of abstraction in Verilog	L4
5	Evaluate the functionality of digital design by implementing on FPGA kits	L5
6	Design digital systems using FSM	L3

Table: Mapping Levels of COs to POs / PSOs														
Cos	Program Outcomes (POs)											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	3	-	1	-	-	-	-	-	-	-	1	1	-	-
CO2	3	2	1	2	3	-	-	-	1	-	1	2	1	-
CO3	3	2	3	1	2	-	-	1	1	-	1	2	1	-
CO4	3	3	2	3	3	1	-	1	-	1	2	2	2	1
CO5	3	3	2	3	3	1	-	-	-	1	-	2	2	1
CO6	3	3	3	3	3	2	-	1	2	2	2	2	1	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. M. Morris Mano Michael D. Ciletti , "Digital Design with an Introduction to the Verilog HDL", 6th Edition, Pearson Education, 2014.

2. Stephen Brown, Zvonko Vranesic, "Fundamentals of Digital Logic with Verilog design", McGraw Hill, 2014.

3. Nazein M. Botros, "HDL programming (VHDL and Verilog)", Dreamtech Press, 2006.

4. Douglas J Smith, "HDL Chip Design", Doone publications 1996.

REFERENCE BOOKS:

1. John M Yarbrough, "Digital Logic Applications and Design", Thomson Learning, 2014.

2. Donald D. Givone, "Digital Principles and Design", McGraw Hill, 2015.

3. Samir Palnitkar, "Verilog HDL: A Guide to Digital Design and Synthesis", Pearson Education, 2016.

E-Resources:

1. https://archive.nptel.ac.in/courses/106/105/106105165/
2. https://nptel.ac.in/courses/117105080

Activity Based Learning (Suggested Activities in Class)

1. Design problem solving and Programming using group discussion. E.g., Traffic light controller, Digital Clock, Elevator.
2. Demonstration of solution to a problem through simulation.

****

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

DATABASE MANAGEMENT SYSTEMS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Course Code: 24AD2304	**Credits**: 04
Hours /Week: 05 Hours	**Total Hours**: 39 (L)+ 26(P) Hours
L–T–P–J: 3-0-2-0	

Course Learning Objectives:

This Course will enable students to:

1. Acquire the concept of databases and Relational Model., Entity-Relationship Model and relational model for creating and designing databases for the real-world scenario.
2. Develop queries to extract data from the databases using a structured query language.
3. Draw ER Diagrams and Optimize the Database design using Normalization Concepts.
4. Understand the fundamental concepts of NoSQL databases and their advantages over traditional relational RDBMS
5. Learn to design, implement, and manage scalable data solutions using MongoDB

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture method, but different *type of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyze information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps

improve the students' understanding.

Module 1: RELATIONAL DATABASES	**09 Hours**
Purpose of Database System, Views of data, Introduction to relational databases, Structure of Relational Databases, Database Schema, Keys, Schema Diagrams, RDBMS- Overview of the SQL Query Language, SQL Data Definition, Basic Structure of SQL, Basic Operations of DDL, DML.	
Module 2: RELATIONAL QUERY LANGUAGE	**08 Hours**
Aggregate Functions , Nested Subqueries , Join Expressions , Views , Transactions , SQL Data Types and Schemas, Authorization, Accessing SQL from a Programming Language, Functions and Procedures ,Triggers, Cursor.	
Module 3: DATABASE DESIGN	**06 Hours**
Entity, Relationship model, E-R Diagrams, Complex Attribute, Mapping Cardinalities, ER-to-Relational Mapping, Features of Good Relational Designs, Decomposition Using Functional Dependencies, Non loss Decomposition – First, Second, Third Normal Forms, Dependency Preservation, Boyce/Codd Normal Form, Multi-valued Dependencies and Fourth Normal Form, Join Dependencies and Fifth Normal Form.	
Module 4: INTRODUCTION TO NOSQL, MONGODB	**09 Hours**
Fundamentals of NoSQL, Definition and evolution of NoSQL databases, Differences between NoSQL and SQL databases, Types of NoSQL databases: Key-Value, Document, Column-Family, Graph, CAP Theorem (Consistency, Availability, Partition Tolerance), Overview of MongoDB , Introduction to MongoDB and its features, CRUD Operations in MongoDB, Creating, reading, updating, and deleting documents, Querying data with filters, projections, and sorting.	
Module 5: ADVANCED MONGODB FEATURES	**08 Hours**
Data Replication and Sharding-Setting up replica sets for high availability, Implementing sharding for horizontal scaling, Data distribution and shard key selection, Hands-on: Configuring replication and sharding in MongoDB, Aggregation Framework - Introduction to MongoDB Aggregation Pipeline, Aggregation stages: $match, $group, $sort, $project, $lookup.	

At the end of the Course the Students will be able to

Course Outcome	Description	Bloom's Taxonomy Level
1	Understand the basic concepts of database management system and Relational Model	L2

2	Utilize SQL concepts to build and manipulate relational databases for a given schema.	L3
3	Apply normalization techniques in designing the relational database	L3
4	**Demonstrate** proficiency in performing CRUD operations, indexing, and basic data modelling using MongoDB	L3
5	**Analyze** advanced MongoDB functionalities such as aggregation pipelines, indexing strategies, replication, and sharding	L4

Table: Mapping Levels of COs to POs / PSOs

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	1	-	-	-	-	-	-	-	-	-	-	-	2	-
CO2	1	2	3	-	3	-	-	-	2	1	-	-	3	-
CO3	1	1	2	-	-	-	-	-	2	1	-	-	3	-
CO4	1	1	-	-	-	-	-	-	-	-	-	-	1	-
CO5	1	2	3	-	3	-	-	-	2	1	-	-	3	-

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

Textbooks:

1. Abraham Silberschatz, Henry F. Korth, S. Sudharshan, “Database System Concepts”, Seventh Editi on, McGraw Hill, 2020.
2. MongoDB: The Definitive Guide" by Shannon Bradshaw, Eoin Brazil, and Kristina Chodorow, O'Reilly Media, ISBN: 978-1491954462

Reference Books:

1. Raghu Ramakrishnan and Johannes Gehrke, “Database Management Systems”, Third Edition, McGraw-Hill, 2003.
2. C.J. Date, A. Kannan, S. Swamynatham: “An Introduction to Database Systems”, Eight Edition, Pearson Education, 2012.

Activity Based Learning:

1. Real world problem solving and puzzles using group discussion.
2. Demonstration of solution to a problem through programming.

E Resources:

1. https://www.ibm.com/docs/en/zos-basic-skills?topic=zos-what-is-database-management-system
2. https://www.mongodb.com/resources/basics/database-management-system
3. https://www.oracle.com/in/database/what-is-database/

4. https://onlinecourses.swayam2.ac.in/cec19_cs05/preview
5. https://onlinecourses.nptel.ac.in/noc19_cs46/preview

Laboratory Experiments:

1. Database Table Creation and Manipulation with key constraints - Create a database table and add constraints such as primary key, unique, check, and not null. Insert, update, and delete rows using SQL DDL and DML commands.
2. Querying with Conditions and Aggregates - Query the database tables using various WHERE clause conditions and implement aggregate functions like SUM, AVG, COUNT, etc.
3. Subqueries and Joins - Perform queries involving subqueries and simple join operations to retrieve data from multiple tables also implement natural joins, equijoins, and outer joins in your queries.
4. User Defined Functions and Stored Procedures - Write and execute user defined functions and stored procedures in SQL for various operations.
5. Transactions and Control Commands - Execute complex transactions and understand the use of Data Control Language (DCL) and Transaction Control Language (TCL) commands.
6. SQL Triggers - Write SQL triggers for handling insert, delete, and update operations on a database table.
7. Develop a program to implement cursor
8. NoSQL Database Operations - Create and manage document-based data using CRUD and Aggregation commands using MongoDB.
9. Implement a Replication and Sharding Setup using MongoDB

****

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

DATA ANALYTICS AND VISUALIZATION

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Course Code	**:**	**24AD2305**	**Credits**	**:**	**03**
Hours / Week	**:**	**03 Hours**	**Total Hours**	**:**	**39 Hours**
L–T–P–S	**:**	**3–0–0–0**			

Course Learning Objectives:

This course will enable students to:

1. Understand the fundamental differences and relationships among data science concepts such as statistics, data mining, analytics, and data science itself.
2. Familiarize with Python-based data analytics libraries like NumPy, SciPy, Pandas, and visualization tools such as Matplotlib, Plotly, and Bokeh.
3. Apply advanced visualization methodologies such as dendrograms, linked views, and projection-based techniques for exploratory data analysis.
4. Visualize clustering techniques and complex structures like mixture models, contingency tables, and kernel-based representations.

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture method, but different *type of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyze information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

MODULE – I Introduction to Data Analytics	**08 Hours**
Statistics vs Data Mining vs Data Analytics vs Data Science; Understanding data: Introduction, Types of Data, Types of Data Distribution: Normal distribution, Poisson Distribution, Binomial Distribution, Uniform distribution, Data Architecture and its components, Structured and Unstructured data. Steps of data analysis, Python Packages for Data Analysis: Numpy, Scipy, Matplotlib, Plotyly/Bokeh. DataFrames, Usage of frames analytical roles, File handling and reading data for processing	
MODULE – II Visualization in high-dimensional data	**08 Hours**
Introduction, Computational Statistics and Data Visualization; A Brief History of Data Visualization, Good Graphics, Static Graphics, Data Visualization Through Their Graph Representations. Graph-theoretic Graphics, High-dimensional Data Visualization, Multivariate Data Glyphs: Principles and Practice, Linked Views for Visual Exploration, Linked Data Views, Visualizing Trees and Forests.	
MODULE – III Visualization Methodologies	**08 Hours**
Interactive Linked Micromap Plots for the Display of Geographically Referenced Statistical Data, Grand Tours, Projection Pursuit Guided Tours and Manual Controls, Multidimensional Scaling, Huge Multidimensional Data Visualization: Back to the Virtue of Principal Coordinates and Dendrograms in the New Computer Age, Multivariate Visualization by Density Estimation.	

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

MODULE – IV Visualization of Cluster Analysis	**08 Hours**
Structured Sets of Graphs, Regression by Parts: Fitting Visually Interpretable Models with GUIDE, Structural Adaptive Smoothing by Propagation–Separation Methods, Smoothing Techniques for Visualization, Data Visualization via Kernel Machines, Visualizing Cluster Analysis and Finite Mixture Models, Visualizing Contingency Tables.	
MODULE – V Applications	**07 Hours**
Visualization for Genetic Network Reconstruction, Reconstruction, Visualization and Analysis of Medical Images, Exploratory Graphics of a Financial Dataset, Graphical Data Representation in Bankruptcy Analysis, and Visualizing Functional Data with an Application to eBay's Online Auctions, Visualization Tools for Insurance Risk Processes.	

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Demonstrate a clear understanding of the various types of data, distributions, and the overall analytics process.	L2
2	Utilize Python libraries to perform data analysis and generate fundamental visualizations.	L3
3	Analyze and interpret high-dimensional data using advanced visualization techniques including graph-theoretic and glyph-based methods.	L4
4	Evaluate and compare different visualization methodologies for cluster analysis, regression, smoothing, and kernel machines.	L5
5	Investigate real-world applications of data visualization in various domains like medical imaging, finance, and insurance, etc.	L4

Table: Mapping Levels of COs to POs / PSOs

COs	**Program Outcomes (POs)**											**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO1	2				3				2		3	1	1	2
CO2	2				3				2		3	2	2	2
CO3	2	3	3	1	3				2		3	2	2	3
CO4	2	3	3	1	3				2		3	2	2	2
CO5	2	3	3	1	3				2		3	2	2	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Text Books

1. Bharti Motwani, Data Analytics using Python (1 ed.), Wiley, 2020. ISBN 8126502959
2. JosDirksen, "Expert Data Visualization", Packt Publishing, 2017.

References

1. Chun-houh Chen, Wolfgang Härdle, Antony Unwin, "Handbook of Data Visualization", Springer- Verlag Berlin Heidelberg, Germany, 2008.
2. Andy Kirk, "Data Visualization: A Handbook for Data Driven Design", SAGE publishing, 2016.
3. Julie Steele and Noah Iliinsky, "Designing Data Visualizations: Representing Informational Relationships", O'REILLY, 2011.

E-Resources

https://nptel.ac.in/courses/106/106/106106179
https://nptel.ac.in/courses/110/106/110106064
https://nptel.ac.in/courses/106/101/106101234
https://nptel.ac.in/courses/106/108/106108222
https://nptel.ac.in/courses/110/107/110107093

Activity Based Learning (Suggested Activities in Class)

1. Students choose a real-world dataset, clean it, and create interactive dashboards to uncover a trend.
2. Provide a messy dataset with missing values, duplicates, outliers, and inconsistent formats and preprocess it.

****

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

ARTIFICIAL INTELLIGENCE

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Course Code	:	24AD2306	**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	39 Hours
L–T–P-J	:	3–0–0-0			

Course Learning Objectives:

This Course will enable students to:

1. Understand the basic concepts and techniques in Artificial Intelligence.
2. Introducing the concepts of neural networks and advanced neural networks
3. Understand the concepts of genetic algorithms
4. Establish basic knowledge about optimization techniques in soft computing

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

MODULE – I Introduction to AI and Intelligent Agents	**08 Hours**
Introduction to AI, Foundations of Artificial Intelligence, Risks and Benefits of AI INTELLIGENT AGENTS: Agents and Environments, Good Behavior: The Concept of Rationality, The Nature of Environments, The Structure of Agents, SOLVING PROBLEMS BY SEARCHING: Problem-Solving Agents, Example Problems LOGICAL AGENTS**:** Knowledge-Based Agents, Logic, Propositional Logic: A Very Simple Logic, Fuzzy Logic Basics	
MODULE – II Neural Networks	**08 Hours**

Mathematical model of neuron, ANN architectures, Learning rules, Learning Paradigms. Perceptron network, Backpropagation network, Backpropagation learning and its applications, Variants of BPA	
MODULE – III Advanced Neural Networks	**08 Hours**
Associative Memory: Auto correlation, Hetero Correlation, Exponential BAM, Applications. Adaptive Resonance Theory: Vector Quantization, ART1, ART2, applications, Kohonen's Self Organizing Map	
MODULE – IV Genetic Algorithms	**08 Hours**
Introduction to Evolutionary Computation: Biological and artificial evolution, evolutionary computation, a simple genetic algorithm. Search Operators: Crossover, mutation, crossover and mutation rates, Selection Schemes: Fitness proportional selection and fitness scaling, ranking, tournament selection, selection pressure and its impact on evolutionary search.	
MODULE – V Optimization and Decision Support Techniques	**07 Hours**
Swarm Intelligence Introduction--Ant Colony Optimization-Particle Swarm Optimization-Applications, scatter search and other search techniques, simulated annealing	

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	**Explain** the foundational concepts of Artificial Intelligence, the structure and behavior of intelligent agents, and various types of environments.	L2
2	**Apply** neural network architectures, learning algorithms and demonstrating their use in solving classification and pattern recognition problems.	L3
3	**Analyze** advanced neural network models such as associative memory, ART, and Kohonen's SOM to solve complex learning and clustering tasks.	L4
4	**Implement** simple Genetic Algorithms using selection, crossover, and mutation operations to solve optimization problems.	L5
5	**Evaluate** the effectiveness of optimization and decision support techniques for real-world AI applications.	L5

Table: Mapping Levels of COs to POs / PSOs

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

COs	Program Outcomes (POs)											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	3	2	2		2			1				3	2	2
CO2	3	3	3	2	2	1		1			2	3	3	2
CO3	3	3	3	2	3						2	3	2	3
CO4	3	3	3	3	3	1	1	1	1	1	2	3	3	2
CO5	3	3	3	3	3			1				3	2	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Stuart Russell and Peter Norvig, Artificial Intelligence – A Modern Approach, Pearson Education Press, Fourth Edition.
2. S. Rajasekaran and G.A.V. Pai, "Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis and Applications", PHI, 2017.

REFERENCE BOOKS:

1. T. J. Ross, "Fuzzy Logic with Engineering Applications", John Wiley & Sons, 3rd Edition, 2016.
2. S.N Sivanandam, S N Deepa, " Principles of Soft Computing", Wiley India, 2nd Edition, 2011.
3. Davis E. Goldberg, "Genetic Algorithms: Search, Optimization and Machine Learning", Pearson Education, Fourth impression 2009.

E-Resources:

https://nptel.ac.in/courses/106106147
https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-034-artificial-intelligence-fall-2010/
https://nptel.ac.in/courses/106106139

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving using group discussion.
2. Demonstration of solutions to a problem through programming.

****

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

TRANSFORMS AND NUMERIAL TECHNIQUES

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Course Code:	24AD2401	**Credits**	: 03
Hours / Week:	03 Hours	**Total Hours**	: 39 Hours
L–T–P–J:	3–0–0–0		

Course Learning Objectives:

This Course will enable students to:

1. **Apply** their knowledge of Laplace transforms and inverse Laplace transforms to proficiently solve linear ordinary differential equations with constant coefficients, facilitating the analysis and modelling of complex systems.
2. **Analyze** periodic functions using Fourier series, assessing the convergence properties and precision of the series expansion, thereby enhancing their ability to understand and manipulate periodic phenomena.
3. **Utilize** complex exponential form, Fourier transforms of basic functions, and Fourier sine and cosine transforms to solve problems involving Fourier integrals, developing proficiency in applying these techniques to various mathematical scenarios.
4. **Employ** numerical methods, including Euler's Method, Runge-Kutta 4th order, Adams-Bashforth, and Adams-Moulton Methods, to solve differential equations and effectively analyze dynamic systems, enabling them to model real-world phenomena and make accurate predictions**.**
5. **Apply** finite difference methods, including the Crank-Nicolson method and appropriate techniques for hyperbolic PDEs, to effectively solve different types of partial differential equations (PDEs) such as elliptic, parabolic, and hyperbolic equations, enhancing their problem-solving skills in the context of differential equations and their applications.

MODULE – I: Laplace Transform and Inverse Laplace Transform	**09 Hours**
Laplace Transforms of Elementary functions (without proof), Laplace Transforms of $e^{at}f(t)$, $t^n f(t)$ and $\frac{f(t)}{t}$, Periodic functions, Module step function and impulse functions Inverse Laplace Transforms- By the method of Partial Fractions, Logarithmic and Trigonometric functions, Convolution Theorem, Inverse Laplace transform using Convolution Theorem Solution to Differential Equations by Laplace Transform.	
MODULE – II: Fourier Series	**09 Hours**
Periodic Functions, Trigonometric Series, Fourier series Standard function, Functions of any Period 2L, Even and Odd functions, Half-range Expansions. Practical Harmonic analysis (calculate average power and RMS values of periodic	

waveforms)	
MODULE – III: Fourier Transform	**06 Hours**
Calculation of Fourier integrals using complex exponential form, Fourier transform of basic functions Fourier sine and cosine transforms.	
MODULE – IV: Numerical Methods for Solving Ordinary Differential Equations	**07 Hours**
Euler's Method-Basic principles of Euler's method for solving first-order ODEs, Runge-Kutta 4th order, Multistep Methods-Explanation of multistep methods (Adams-Bashforth, Adams–Moulton Methods), Second-Order ODE. Mass–Spring System (Euler Method, Runge–Kutta Methods)	
MODULE – V: Numerical Methods for Partial Differential Equations	**08 Hours**
Classification of PDEs (elliptic, parabolic, hyperbolic), Difference Methods (Laplace and Poisson Equations), Derivation of finite difference approximations, Crank–Nicolson Method ,Method for Hyperbolic PDEs	

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
C01	3		2										2	
C02	3		3		1								2	
C03	3	3	1										2	
C04	3	3	1										2	
C05	3	3	1										2	

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

TEXT BOOKS:

1. Erwin Kreyszig, Advanced Engineering Mathematics, 2015, 10th Edition, Wiley India.

REFERENCE BOOKS:

1.Higher Engineering Mathematics, B.S. Grewal, 2015, 43rd Edition, Khanna Publishers.
2. Higher Engineering Mathematics, John Bird, 2017, 6 th Edition, Elsevier Limited.

E-Resources:

1. https://nptel.ac.in/courses/111106139
2. https://nptel.ac.in/courses/111101164
3. https://nptel.ac.in/courses/111105038

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

DESIGN AND ANALYSIS OF ALGORITHMS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Course Code	:	24AD2402	**Credits**	:	04
Hours / Week	:	05 Hours	**Total Hours**	:	39(L)+26(P) Hours
L–T–P-J	:	3–0–2-0			

Course Learning Objectives:

This Course will enable students to:

1. **Analyze** the non-recursive and recursive algorithms and to represent efficiency of these algorithmsin terms of the standard Asymptotic notations.
2. **Devise** the Brute Force and Divide and Conquer techniques to design the algorithms and apply these methods in designing algorithms to solve a given problem.
3. **Explain** the Decrease and Conquer, Transform and Conquer algorithm design techniques, and Timeversus Space Trade-offs.
4. **Get the idea** of Greedy method and dynamic programming methods and apply these methods indesigning algorithms to solve a given problem.
5. **Describe** and illustrate the idea of Backtracking and Branch and Bound algorithm design techniques to solve a given problem.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

MODULE – I	**08 Hours**

INTRODUCTION: What is an Algorithm? Fundamentals of Algorithmic Problem Solving. ***(Text Book-1: Chapter 1: 1.1 to 1.2)*** **FUNDAMENTALS OF THE ALGORITHMS EFFICIENCY:** Analysis Framework, Asymptotic Notations and Standard notations and common functions ***(Text Book-2:Chapter 3: 3.1, 3.2)***, Mathematical Analysis of Non-recursive and Recursive Algorithms, ***(Text Book-1: Chapter 2: 2.1, 2.3, 2.4,)***	
MODULE – II	**08 Hours**
BRUTE FORCE: Background, Selection Sort, Brute-Force String Matching. ***(Text Book-1: Chapter 3: 3.1, 3.2)*** **DIVIDE AND CONQUER:** General method, Recurrences: The recursion-tree method, The master method. ***(Text Book-2: Chapter 4: 4.4, 4.5***), Merge sort, Quick sort, Binary Search, Multiplication of large integers, **Case study:** Strassen's Matrix Multiplication. ***(Text Book-1: Chapter 4: 4.1 to 4.3, 4.5)***	
MODULE – III	**06 Hours**
DECREASE & CONQUER: General method, Insertion Sort, Graph algorithms: Depth First Search, Breadth First Search, Topological Sorting **TRANSFORM AND CONQUER:** **Case study:** Heaps and Heap sort. **TIME AND SPACE TRADEOFFS:** Input Enhancement in String Matching: Horspool's algorithm, Hashing: Open and Closed hashing. ***(Text Book-1: Chapter 5: 5.1 to 5.3, Chapter 6: 6.3 to 6.4, Chapter 7: 7.2 to 7.3)***	
MODULE – IV	**09 Hours**
GREEDY TECHNIQUE: General method of Greedy technique, Single-Source Shortest Paths: General method, The Bellman-Fordalgorithm, Single-Source Shortest Paths in DAGs, Dijkstra's Algorithm ***(Text Book-2: Chapter 24: 24.1 to 24.3).*** Minimum Spanning Trees: Prim's Algorithm, Optimal Tree problem: Huffman Trees; **Case study:** Kruskal's Algorithm. ***(Text Book-1: Chapter 9: 9.1, 9.2, 9.4).*** **DYNAMIC PROGRAMMING:** General method, The Floyd-Warshall Algorithm, Johnson's algorithm for sparse graphs ***(Text Book-2: Chapter 25: 25.1 to 25.3)***, The Knapsack problem ***(Text Book-1: Chapter 8: 8.4).***	
MODULE – V	**08 Hours**
LIMITATIONS OF ALGORITHMIC POWER P, NP and NP-complete problems ***(Text Book-1: Chapter 11: 11.3***) **BACKTRACKING:** General method, N-Queens problem, Subset-sum problem. ***(Text Book-1: Chapter 12: 12.1***)	

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

BRANCH AND BOUND:
General method, Travelling Salesman problem, Approximation algorithms for TSP.
Case study: Knapsack Problem.
(Text Book-1: Chapter 12: 12.2, 12.3)

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Analyze the given recursive and non-recursive algorithms for time using step count, substitution method, recurrence tree method and mathematical analysis method. Represent the complexity of the algorithm using asymptotic notation.	L3
2	Solve sorting, searching, matrix multiplication problems basedon divide and conquer design technique and implement in C programming language.	L3
3	Apply algorithms for graph-based problems (DFS, BFS and Topological sorting) using decrease and conquer design techniques. Distinguish the trade-offs between space and time complexity.	L3
4	Apply algorithms for finding the shortest path and minimum spanning tree for a given graph using greedy and dynamic programming techniques and implement the 0/1 knapsack problem in C programming language.	L3
5	Apply an efficient algorithm to solve N-Queens problem, Subset-sum problem, Knapsack and Traveling salesman problem using branch and bound and backtracking design technique. Describe the limitation of algorithmic power in terms of P, NP, NP hard and NP complete categorizations.	L3
6	Implement the graph-based algorithms including DFS, BFS, Warshall's Algorithm, Floyd's Algorithm, Kruskal's Algorithm, and Dijkstra's Algorithm in C programming language. (**Lab Experiments**)	L3

Table: Mapping Levels of Cos to POs /PSOs

COs	Program Outcomes (POs)											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	PSO1	PSO2	PSO3
CO1	2	3	-	-	-	-	-	-	-	1	-	1	1	-
CO2	2	3	-	3	-	-	-	-	-	1	-	1	1	-
CO3	1	2	3	-	-	-	-	-	-	1	-	1	2	-
CO4	1	2	3	3	-	-	-	-	-	1	-	1	2	-
CO5	1	2	3	-	-	-	-	-	-	1	-	1	2	-
CO6	1	2	3	3	-	-	-	-	-	1	-	1	2	-

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Anany Levitin, "Introduction to the Design & Analysis of Algorithms", 2nd Edition, Pearson Education, 2011.
2. Thomas H. Cormen, Charles E.Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction toAlgorithms", 3rd Edition, PHI, 2014.

REFERENCE BOOKS:

1. Horowitz E., Sahni S., Rajasekaran S, "Computer Algorithms", Galgotia Publications, 2001.
2. R.C.T. Lee, S.S. Tseng, R.C. Chang & Y.T.Tsai, "Introduction to the Design and Analysis ofAlgorithms: A Strategic Approach", Tata McGraw Hill, 2005.

E-Resource

1. https://nptel.ac.in/courses/106/101/106101060/
2. http://cse01-iiith.vlabs.ac.in/
3. http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=IntroToAlgorithms
4. https://www.coursera.org/specializations/algorithms

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving and puzzles using group discussion. E.g., Fake coinidentification, Cabbage puzzle, Konigsberg bridge puzzle etc.,
2. Demonstration of solutions to a problem through programming.

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

OPERATING SYSTEMS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Course Code	: 24AD2403	**Credits**	:	04
Hours / Week	: 05 Hours	**Total Hours**	:	39(L)+26(P) Hours
L–T–P–S	: 3–0–2–0			

Course Learning Objectives:

This course will enable students to:

1. To understand the basic concepts and functions of operating systems.
2. To understand Processes and Threads
3. To analyze Scheduling algorithms.
4. To understand the concept of Deadlocks.
5. To analyze various Memory and Virtual memory management, File system and storage techniques.
6. To discuss the goals and principles of protection in a modern computer system.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods that teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only the traditional lecture method but a different *type of teaching method* that may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note-taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain the functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher-order Thinking questions in the class.
6. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the student's understanding.

MODULE – I	**10 Hours**

OS Overview and System Structure

Introduction to operating systems, System structures: What operating systems do?; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and Security; Distributed system; Computing environments.

Operating System Services: User - Operating System interface; System calls; Types of system calls; System programs; Operating system design and implementation; Operating System structure; Virtual machines.

MODULE – II	**12 Hours**

Process Management:
Process Management: Process concept; Process scheduling; Operations on processes. Multi-threaded Programming: Overview; Multithreading models; Threading issues. Process Scheduling: Basic concepts; Scheduling Criteria; Scheduling Algorithms

MODULE – III	**10 Hours**
Process Coordination Process Synchronization: The critical section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors. Deadlocks: Deadlocks; System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock.	
MODULE – IV	**10 Hours**
Memory Management Memory Management Strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation. Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing	
MODULE – V	**10 Hours**
File System and Secondary Storage Structure File System, Implementation of File System: File system: File concept; Access methods; Directory structure; File system mounting; File sharing. Protection: Implementing File system: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management. Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management; Swap space management. Protection and Security: Protection: Goals of protection, Principles of protection, System Security: The Security Problem, Program Threats, System and Network Threats.	

Course Outcomes:

At the end of the course the student will be able to:

1. **Summarize** basic concepts of operating systems and identify its variants.
2. **Apply** FIFO, SJF, SRT, Priority, Round Robin algorithms to solve process scheduling.
3. **Summarize** process coordination and apply Banker's algorithm to prevent inter-process deadlock.
4. **Distinguish** contiguous and virtual memory management.
5. **Apply** the page replacement algorithms to detect page fault appear in virtual memory.
6. **Apply** FCFS, SSTF, SCAN, C-SCAN, Look and C-Look algorithms to solve I/O scheduling.

Table: Mapping Levels of COs to POs / PSOs														
COs	**Program Outcomes (POs)**											**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO1	3	1	0	0	1	0	0	0	0	0	0	2		0
CO2	3	2	1	0	2	0	0	0	0	0	0	2	0	0
CO3	3	2	1	0	2	0	0	0	0	0	0	2	0	0
CO4	3	2	1	0	2	0	0	0	0	0	0	2	0	0
CO5	3	2	1	0	2	0	0	0	0	0	0	2	0	0

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

TEXT BOOKS:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 8th edition, WileyIndia, 2010.

REFERENCES:

1. Operating Systems-Internals and Design Principles, William Stallings, 6th Edition, Pearson Education, 2009.
2. Operating Systems: A Modern Perspective, Gary J. Nutt, Addison-Wesley, 1997.

E-Resources:

1. https://onlinecourses.nptel.ac.in/noc24_cs80/preview
2. https://onlinecourses.nptel.ac.in/noc24_cs108/preview
3. https://www.coursera.org/learn/codio-intro-to-operating-systems-2-memory-management
4. Modern Operating Systems by Andrew S. Tanenbaum - Known for its comprehensive coverage of modern operating systems principles and design.
5. Operating System Fundamentals - Course (nptel.ac.in)
6. Introduction to Operating Systems - Course (nptel.ac.in)
7. Intro to Operating Systems 1: Virtualization | Coursera
8. Intro to Operating Systems 3: Concurrency | Coursera
9. Intro to Operating Systems 2: Memory Management | Coursera

Activity Based Learning (Suggested Activities in Class)

1. Implementing Operating System concepts by doing Mini projects.
2. Case study to compare various Operating Systems and their performance.
3. Present real-world case studies or problems related to operating systems design and performance optimization, encouraging students to apply theoretical knowledge to analyze and propose solutions.

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

LABORATORY EXPERIMENTS

1. Write a C program to create a new process that exec a new program using system calls fork(), execlp() & wait().
2. Write a C program to display PID and PPID using system calls getpid () & getppid ().
3. Write a C program using I/O system calls open(), read() & write() to copy contents of one file to another file.
4. Write a C program to implement multithreaded program using pthreads.
5. Write C program to simulate the following CPU scheduling algorithms.

a. FCFS
b. SJF
c. Priority
d. Round Robin

6. Write a C program to simulate producer-consumer problem using semaphores.
7. Write a C program to simulate Banker's algorithm for the purpose of deadlock avoidance.
8. Write a C program to simulate deadlock detection.
9. Write a C program to simulate paging technique of memory management.
10. Write a C program to simulate page replacement algorithms.

a. FIFO b. LRU c. LFU

11. Write a C program to simulate the following file organization techniques,

a. Single level directory b. Two level directory

12. Write a C program to simulate the following file allocation strategies,

a. Sequential b. Indexed

Open-ended experiments

1. Write a program to simulate dynamic memory allocation using malloc() and free().
2. Set up and manage virtual machines using VirtualBox or VMware.
3. Write a program to navigate and manipulate file directories.

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

APPLIED MACHINE LEARNING

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Course Code:	24AD2404	**Credits**	:	04
Hours / Week	: 5 Hours	**Total Hours**	:	39(L)+26(P) Hours
L–T–P–S	: 3–0–2–0			

Course Learning Objectives:

1. Introduce the fundamental concepts of machine learning, its types, and evaluating models using appropriate performance metrics.
2. To provide an in-depth understanding of regression techniques and regularization methods, enabling students to build accurate predictive models and avoid overfitting.
3. To familiarize students with key classification algorithms and Bayesian methods, along with techniques for model inference, decision-making, and margin optimization.
4. To equip students with knowledge of clustering techniques, ensemble methods, dimensionality reduction techniques, and sequence modeling using Hidden Markov Models for real-world applications.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods that teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only the traditional lecture method but a different *type of teaching method* that may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note-taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain the functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher-order Thinking questions in the class.
6. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the student's understanding.

MODULE – I: Introduction	**07 Hours**
Introduction to Machine Learning, Types of Learning – Supervised, Unsupervised, Semi Supervised, Reinforcement. Training and testing classifier models, Model evaluation- Performance Measures - Confusion Matrix – Binary, Multi Classification - (precision, recall, F1-score, accuracy, ROC etc.), Bias, Variance. Understanding data with statistics and Visualization, Data preparation, Data feature selection.	
MODULE – II: Regression and Regularization	**08 Hours**

Regression – Linear, Non-linear, Simple and Multi-variable regression, Regression model and calculate accuracy, Application of regression. Regularization - Lasso Regression, Ridge Regression, Gradient Descent Method for adjusting the weights, Stochastic Gradient Descent, Cross-validation.	
MODULE – III: Classification	**08 Hours**
Logistic Regression, Linear Regression vs Logistic Regression, K-Nearest Neighbor, Decision Tree Learning, Generative Vs Discriminative models, Bayes Rule, Maximum A Posteriori (MAP) Vs Maximum Likelihood (ML), Bayesian Networks: Building BNs, Inferencing, Bayes Decision theory – Binary features, Bayesian Belief Network, Maximum margin classifiers (Support Vector Machine).	
MODULE – IV: Clustering and Ensemble Methods	**08 Hours**
Cluster Analysis - Proximity Matrices, Dissimilarities Based on Attributes, Clustering Algorithms, Expectation Maximization, Convergence, K-means, Application of K-means, Gaussian Mixtures as Soft K-means Clustering, Hierarchical Clustering. **ENSEMBLE METHODS:** Ensemble – Voting, Bagging, Boosting, AdaBoosting, Gradient Boosting, Random Forest.	
MODULE – V: Dimensionality Problem, and HMM	**08 Hours**
Dimensionality Problem: Dimensionality Reduction: Principal Component Analysis (PCA) for feature reduction: Maths behind PCA., Linear Discriminant Analysis (LDA), Hidden Markov Model (HMM). Real life use cases of machine learning algorithms.	

Course Outcomes	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Understand the types of machine learning techniques and comprehend the parameters to evaluate the performance of various models.	L2
2	Develop solutions for regression problems using supervised learning algorithms with the given dataset.	L3
3	Develop solutions for classification problems using supervised learning algorithms with the given dataset.	L3
4	Apply clustering algorithms and ensemble methods to solve problems.	L3
5	Develop models in python to solve complex problems using ML algorithms and analyze the model performance.	L4

Table: Mapping Levels of COs to POs / PSOs														
COs	Program Outcomes (POs)											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	1	1												3
CO2	2	2	2											
CO3	3	2	3											
CO4	2	2	2											
CO5	3	3	3	2	3				3	3	2	2	3	3

3:Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Ethem Alpaydin, Introduction to Machine Learning (Adaptive Computation and Machine Learning series), The MIT Press, Third Edition.
2. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, Third Edition, 2019.
3. Trevor Hastie, Robert Tibshirani, Jerome H. Friedman, The Elements of Statistical Learning, Second Edition, Springer.

REFERENCE BOOKS:

1. Christopher M Bishop. Pattern recognition and Machine Learning, Springer

2. Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong, Mathematics for Machine Learning, Cambridge University Press.

3. Thomas M. Mitchell, Machine Learning, McGraw- Hill, Inc. New York

E-Resources:

1. https://www.kaggle.com/learn/intro-to-machine-learning
2. https://learn.microsoft.com/en-us/training/Module_s/introduction-to-machine-learning/
3. https://onlinecourses.nptel.ac.in/noc22_cs29/preview

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving using group discussion.
2. Demonstration of solutions to a problem through programming.

Laboratory Experiments

1. Data Collection and Preprocessing
2. Exploratory Data Analytics
3. Univariate and Multivariate Linear Regression
4. Logistic Regression

5. Decision Tree Classification
6. Naïve Bayes Classification
7. K Nearest Neighbour Classification
8. SVM
9. K Means Clustering
10. Random Forest
11. XGBoost
12. Mini Project

****

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

COMPUTER ORGANIZATION AND ARCHITECTURE [As per Choice Based Credit System (CBCS) scheme] **SEMESTER – IV**			
Course Code: 24AD2405		**Credits**	: 03
Hours / Week	: 03	**Total Hours**	: 39 Hours
L–T–P-J	: 3–0–0-0		

Course Learning Objectives:

This Course will enable students to:

1. **Understand** the Architecture and programming of ARM microprocessor.
2. **Develop** program using Arm instruction set and appreciate the advanced features provided in the ARM
3. **Understand** the exception handling techniques.
4. **Study in** detail the concept of instruction level parallelism and concepts of pipelining.
5. **Understand** various cache memory mapping techniques and memory Organization.

Teaching-Learning Process

1. ***Lecture method*** along with traditional lecture method, different *type of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *incorporating* brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Showing ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.
6. Showing the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

MODULE – I	**05 Hours**
An Overview of Computing Systems: History of Computers, The Computing Device, **The ARM7TDMI Programmers' Model:** Introduction, Data types, Processor Modes, Registers, Program Status Registers, The vector Table. **Assembler Rules and Directives:** Structure of Assembly Language Module s, Registers, Directives and Macros. **Loads, Stores and Addressing:** LODS and STORES instructions, Operand Addressing, ENDIANNES ***Text Book-1: 1.1 to 1.3; 2.1 to 2.6 ; 4; 5.3, 5.4, 5.5***	
MODULE – II	**05 Hours**
Constants and Literal Pools: The ARM Rotation Scheme, Loading Constants and address into Registers **Logic and Arithmetic:** Flags and their Use, Compare instructions, Data Processing Instructions **Loops and Branches:** Branching, Looping, Conditional Execution, Straight-Line Coding **Subroutines and Stacks:** Stack, Subroutines, Passing parameters to subroutines, The ARM APCS. ***(Text Book-1: 6.1 to 6.4; 7.1 to 7.4; 8.2 to 8.6; 10.1 to 10.5***	
MODULE – III	**05 Hours**
Mixing C and Assembly Language: Inline Assembler Embedded Assembler, Calling Between C and Assembly. **Exception Handling:** Interrupts, Error Conditions, Processor Exception Sequence, The Vector Table, Exception Handlers, Exception Priorities, Procedures for Handling Exceptions. ***(Text Book-1: 11.1 to 11.8; 14.1 to 14.4***	
MODULE – IV	**12 Hours**
Pipelining: Basic and Intermediate Concepts Introduction, The Major Hurdle of Pipelining, How Pipelining Implemented, What makes Pipelining hard to Implement, Extending the MIPS Pipeline to Handle Multicycle Operations, The MIPS R4000 Pipeline, Crosscutting Issues. ***Text Book-2: C.1 to C.7***	
MODULE – V	**12 Hours**
Memory Hierarchy: Introduction, Cache Performance, Six basic cache Optimizations, Virtual Memory, Protection andexamples of Virtual Memory, Fallacies and Pitfalls. ***Text Book-2: B.1 to B.6***	

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	**Apply** knowledge of the internal architecture and organization of ARM microprocessors to utilize their	L3

	components and functionalities.	
2	**Apply** the instruction set of ARM Microprocessor by writing Assembly language programs.	L3
3	**Analyze and compare** the various exception handling techniques.	L4
4	**Examine** the concept of instruction-level parallelism and **analyze** the principles of Pipelining techniques.	L4
5	**Compare** and **Contrast** memory hierarchy and its impact on computer cost/performance.	L4

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
C01	3		2									2		
C02	3		3		1							2		
C03	3	3	1									2		
C04	3	3	1									2		
C05	3	3	1									2		

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. William Hohl, "ARM Assembly Language", 2nd Edition, CRC Press, 2009.
2. John L Hennessy, David A Patterson, "Computer Architecture, A Quantitative Approach", 5th Edition, Morgan Kaufmann publishers, 2012.

REFERENCE BOOKS:

1. David A Patterson, John L Hennessy, "Computer Organization and Design", 4th Edition, Morgan Kaufmann publishers, 2010.
2. Steve Furber, "ARM System-on-chip Architecture", 2nd Edition, Pearson Publications, 2000.
3. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, "Computer Organization", 5th Edition, Tata McGraw Hill, 2002.

E-Resources:

1. https://www.udemy.com/topic/arm-cortex-m/
2. https://www.edx.org/school/armeducation
3. https://onlinecourses.nptel.ac.in/noc22_cs93/preview

Activity Based Learning (Suggested Activities in Class)

1. Mini project implementation using Assembly Language Programming.
2. Demonstration of solutions to a problem through programming.

****

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

BUSINESS ESSENTIALS FOR ENGINEERS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Course Code: 24AD2406	**Credits** : 02
Hours / Week : 02	**Total Hours** : 26 Hours
L–T–P-J : 2-0-0-0	

Course Learning Objectives:

This Course will enable students to:

1. Develop a Comprehensive Understanding of Management Principles.
2. Enhance People and Marketing Management Competencies.
3. Build Financial and Project Management Expertise.
4. Strengthen Strategic Thinking and Quality Management Skills

Teaching-Learning Process

1. ***Lecture method*** along with traditional lecture method, different *type of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *incorporating* brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Showing ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.
6. Showing the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

MODULE – I	**05 Hours**
Introduction: The evolution of management theory, Business functions and their roles, Organizations and types, Levels of management, Types of markets and pillars of management-planning, organizing, leading and controlling. **People Management:** Catalysts for organizational performance – Motivation & Leadership, Organization culture & Change, Human resource functions in a dynamic business environment and evolving dynamics in Industrial Relations.	
MODULE – II	**05 Hours**
Marketing Management: Nature and scope of marketing, Company's orientation towards market place, Importance of marketing concept, Marketing environment, 4p's of marketing, market segmentation, target market selection and positioning.	
MODULE – III	**05 Hours**
Financial Management: Financial accounting, Financial statements and analysis for decisions, Financial planning, Capital, Working capital, Capital structure and Sources of corporate finance, Investment decisions.	
MODULE – IV	**06 Hours**
Project Management: Project screening and Selection, Techniques, Structuring concepts and Tools (WBS, OBS, and LRC, RACE). Project life cycle analysis. Appraisal of a project, Project Planning: Techniques, CPM, PERT- GAN - Time Cost Trade-off and Crashing Procedure, Project Monitoring: Monitoring Techniques and time control System, EVA Analysis	
MODULE – V	**05 Hours**
Quality & Strategy: Quality, Principles, Quality Awards, Standards of Quality culture, Quality metrics programs, Strategy, Vision and Mission, Porter's 5–forces, McKinsey's 7S Model, BCG Matrix, Competitive advantage - Value chain analysis & Resource based view.	

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Understand the basic concepts of management and its functions	L2
2	Apply the functions of management for taking effective decisions	L3
3	Analyze factors influencing management in a competitive business environment	L4
4	Identify business opportunities and challenges	L3
5	Integrate functions of management for building a better organization	L5

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	1					1		2	2	3	2	1		2
CO2	1	1				1		2	2	3	2	2		2
CO3		2				2			1	2	2	2		2
CO4		2	1			2			2	2	2	2	1	2
CO5			1			2			1	2	3	1	1	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Ronald J. Ebert, Ricky W. Griffin, Business Essentials, Pearson, 2019, 12th Edition
2. Harold Koontz, Heinz Weihrich, Mark V. Cannice, Essentials of Management, McGraw hill, 2020, 11th Edition

REFERENCE BOOKS:

1. G. Shainesh Philip Kotler, Kevin lane Keller, Alexander Chernev, Jagdish N. Sheth, Marketing Management, Pearson, 2022, 16th Edition
2. Dessler, G., & Varkkey, B, Human Resource Management, Pearson Education, 2024, 17th Edition
3. Prasanna Chandra, Financial Management: Theory & Practice, Mc Graw Hill, 2022, 11th Edition
4. Poornima M Charantimath, Total Quality Management, Pearson, 2022, 4th Edition
5. IM Pandey, Financial Management, Vikas Publications, 2021, 12th Edition

6. Jack R. Meredith, Mantel, Project Management - A Managerial Approach, John Wiley, 2021,11th Edition

E-Resources:

1. https://nptel.ac.in/courses/110106050
2. https://nptel.ac.in/courses/110105146
3. https://nptel.ac.in/courses/110105069
4. https://nptel.ac.in/courses/110104068
5. https://ocw.mit.edu/courses/15-535-business-analysis-using-financial-statements-spring-2003/
6. https://ocw.mit.edu/courses/15-810-marketing-management-fall-2010/

Activity Based Learning (Suggested Activities in Class)

1. Analyze a case study and present group findings.
2. Each student is given a virtual salary and asked to plan a monthly budget, investment, and savings strategy using Excel/Google Sheets.

****

SCHEME & SYLLABUS FOR BACHELOR OF TECHNOLOGY (B. Tech)

COMPUTER SCIENCE & ENGINEERING (Artificial Intelligence and Data Science)

(5th to 6th Semester)

With effect from 2024

B.Tech. PROGRAMME– 2024-28 BATCH

B Tech CoOrdinator

ChairPerson

Chairperson
Department of CSE (AI & Data Science)
School of Engineering
Dayananda Sagar University
Devarakaggalahalli, Harohalli, Kanakapura Road,
Bengaluru South Dt.-562 112, Karnataka, India

V-SEMESTER											
S.NO	Course Type	Course Code	Course Title	Teaching Hours / Week				Examination			credits
				Lecture	Tutorial	Practical	Project	CIE Marks	SEE Marks	Total Marks	
				L	T	P	J				
1	PCC	24AD3501	COMPUTER NETWORKS	3	0	0	0	60	40	100	3
2	IPCC	24AD3502	DATA MINING AND PREDICTIVE MODELLING	3	0	2	0	60	40	100	4
3	IPCC	24AD3503	DEEP LEARNING	3	0	2	0	60	40	100	4
4	PCC	24AD3504	IOT FUNDAMENTALS AND ITS APPLICATIONS	3	0	0	0	60	40	100	3
5	IPCC	24AD3505	TIME SERIES ANALYSIS	3	0	2	0	60	40	100	4
6	PEC	24AD35XX	PROFESSIONAL ELECTIVE - I	3	0	0	0	60	40	100	3
7	TPC	24AD3506	COGNITIVE AND TECHNICAL SKILLS – V	-	-	-	-	-	-	-	0
TOTAL				18	0	6	0	360	240	600	21

B Tech CoOrdinator

ChairPerson

Chairperson
Department of CSE (AI & Data Science)
School of Engineering
Dayananda Sagar University
Devarakaggalahalli, Harohalli, Kanakapura Road,
Bengaluru South Dt.-562 112, Karnataka, India

Professional Elective Courses - I	
Course Code	**Course Name**
24AD3507	BIG DATA COMPUTING
24AD3508	DEVOPS
24AD3509	KNOWLEDGE ENGINEERING
24AD3510	ACOUSTICS, IMAGE AND VIDEO PROCESSING

B Tech CoOrdinator

ChairPerson

Chairperson
Department of CSE (AI & Data Science)
School of Engineering
Dayananda Sagar University
Devarakaggalahalli, Harohalli, Kanakapura Road,
Bengaluru South Dt.-562 112, Karnataka, India

VI-SEMESTER											
S. NO	Course Type	Course Code	Course Title	Teaching Hours / Week				Examination			Credits
				Lecture	Tutorial	Practical	Project	CIE Marks	SEE Marks	Total Marks	
				L	T	P	J				
1	PCC	24AD3601	RESPONSIBLE AI, ETHICS & POLICY	2	0	0	0	60	40	100	**2**
2	IPCC	24AD3602	APPLIED NLP AND GENERATIVE AI	3	0	2	0	60	40	100	**4**
3	IPCC	24AD3603	HIGH-PERFORMANCE COMPUTING	3	0	2	0	60	40	100	**4**
4	PCC	24AD3604	EDGE COMPUTING	3	0	0	0	60	40	100	**3**
5	PEC	24AD36XX	PROFESSIONAL ELECTIVE – II/MOOC	3	0	0	0	60	40	100	**3**
6	PROJ	24AD3605	MINI PROJECT (RESEARCH METHODOLOGY AND TOOLS)	0	0	0	4	100	--	100	**2**
7	OEC		OPEN ELECTIVE - I	3	0	0	0	60	40	100	**3**
8	TPC	24AD3606	COGNITIVE AND TECHNICAL SKILLS – VI	-	-	-	-	-	-	-	**0**
TOTAL				**17**	**0**	**4**	**4**	**460**	**240**	**700**	**21**

B Tech CoOrdinator

ChairPerson

Chairperson
Department of CSE (AI & Data Science)
School of Engineering
Dayananda Sagar University
Devarakaggalahalli, Harohalli, Kanakapura Road,
Bengaluru South Dt.-562 112, Karnataka, India

COMPUTER NETWORKS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Course Code	:	24AD3501	**Credits**	:	03
Hours /Week	:	03 Hours	**Total Hours**	:	45 Hours
L–T–P-J	:	3–0–0-0			

Course Learning Objectives: This Course will enable students to:

1. Understand the fundamentals of computer networks, network architectures, and communication models.
2. Explain the operation of data link, network, transport, and application layer protocols.
3. Analyse routing, flow control, congestion control, and error control mechanisms in computer networks.
4. Understand Internet protocols, addressing techniques, and network communication services.
5. Apply network security concepts and cryptographic techniques for secure communication systems.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods that teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and role-playing.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three higher-order thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

MODULE – I : Overview of Computer Networks	09 Hours
OVERVIEW OF NETWORKS: Network components, network structure, classification of networks: LAN, MAN, WAN, network protocols and standards, data representation and data flow, OSI reference model, TCP/IP reference model, transmission media: wired and wireless, transmission modes, analog and digital signals, line coding schemes, network topologies, switching methods: circuit switching and packet switching, Ethernet protocol basics, Wi-Fi protocol basics- Introduction to open-source network simulation and analysis tools such as Wireshark, GNS3, and ns 3. **(TEXT BOOK: CHAPTER 1, CHAPTER 2, CHAPTER 3, CHAPTER 4, CHAPTER 7, CHAPTER 8)**	
MODULE – II: Data Link Layer Protocols	**09 Hours**
DATA LINK LAYER PROTOCOLS: Data link layer services, MAC and LLC sublayers, framing techniques and protocols, flow control protocols: Stop-and-Wait and Sliding Window, error detection and correction using CRC and Hamming Code, retransmission techniques, packet analysis using Wireshark and tcpdump, basic network troubleshooting using ping, traceroute, netstat and ss, network performance testing using iperf, multiple access protocols: ALOHA, Slotted ALOHA, CSMA/CD, CSMA/CA, Ethernet protocol, PPP protocol, HDLC protocol, VLAN basics, connecting devices and their basic functions: hub, bridge, switch, router. **(TEXT BOOK: CHAPTER 10, CHAPTER 11, CHAPTER 12, CHAPTER 13, CHAPTER 14 CHAPTER 15)**	
MODULE – III: Network Layer Protocols	**09 Hours**
NETWORK LAYER PROTOCOLS: Network layer design issues, routing and forwarding, routing protocols: Distance Vector and Link State routing, Dijkstra's algorithm, IP addressing and packet format, IPv4 and IPv6 protocols, ARP protocol, DHCP protocol, ICMP protocol, subnetting and CIDR, basics of congestion control. **(TEXT BOOK: CHAPTER 19, CHAPTER 20, CHAPTER 21, CHAPTER 22)**	
MODULE – IV: Transport Layer and Application Layer Protocols	**09 Hours**
TRANSPORT LAYER AND APPLICATION LAYER PROTOCOLS : Transport layer services, multiplexing and demultiplexing, TCP protocol, UDP protocol, TCP segment structure, connection establishment and termination, congestion control: Slow Start and AIMD basics, comparison of TCP and UDP, application layer services, HTTP and HTTPS protocols, FTP protocol basics, e-mail protocols: SMTP and POP3, DNS protocol, World Wide Web services. **(TEXT BOOK: CHAPTER 23 –, CHAPTER 24, CHAPTER 25, CHAPTER 26, CHAPTER 27, CHAPTER 28, CHAPTER 29)**	
MODULE – V: Network Security Protocols	**09 Hours**
NETWORK SECURITY PROTOCOLS : Fundamentals of network security, security attacks and threats, cryptography basics, symmetric key encryption, public key encryption, authentication protocols, digital signatures, hashing techniques, firewalls and firewall architecture, SSL and TLS protocols, secure socket communication, Virtual Private Networks, network security protocols, malware, phishing and cyber attacks, intrusion detection and prevention systems, basics of cybersecurity in communication networks. **(TEXT BOOK: CHAPTER 30, CHAPTER 31, CHAPTER 32)**	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Explain the fundamentals of computer networks, network models, transmission media, and communication techniques.	L2
2	Analyze data link layer protocols, framing, flow control, error control, and LAN communication mechanisms.	L4
3	Apply routing algorithms, IP addressing, subnetting, and network layer protocols for efficient data communication.	L3
4	Understand transport and application layer protocols including TCP/IP, UDP, HTTP, DNS, and related services.	L2
5	Demonstrate knowledge of network security principles, cryptography, authentication, and secure communication protocols	L2

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	3	2		1	2				1		2	2	1	1
CO2	2	3	1	2	2				1		2	2	2	1
CO3	2	3	2	1	3				1	1	2	2	3	1
CO4	2	2	1	2	2				2		2	2	2	1
CO5	2	3	2		2	2	3	1	1		3	3	2	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Text Book

1. Behrouz A. Forouzan, *Data Communications and Networking*, McGraw Hill Education, 5th Edition, 2013.

Reference Books

1. Computer Networks — Andrew S. Tanenbaum and David J. Wetherall, Pearson Education.
2. Computer Networking: A Top-Down Approach — Pearson Education.
3. TCP/IP Illustrated, Volume 1 — Addison-Wesley Professional.
4. Cryptography and Network Security — Pearson Education.
5. Internetworking with TCP/IP — Pearson Education.

DATA MINING AND PREDICTIVE MODELLING

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Course Code	:	24AD3502	**Credits**	:	04
Hours / Week	:	03 Hours (Theory) + 02 Hours (Lab)	**Total Hours**	:	45(T) + 15(L)
L–T–P-J	:	3–0–2-0			

Course Learning Objectives: This Course will enable students to:

1. Understand data mining and knowledge discovery processes.
2. Analyse and pre-process real-world datasets.
3. Build predictive models using statistical and machine learning techniques.
4. Evaluate model performance and uncertainty.
5. Use AI-assisted tools responsibly in data-driven systems.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

MODULE – I	09 Hours
DATA MINING FUNDAMENTALS AND PRE-PROCESSING: Introduction to data mining and knowledge discovery; evolution of data mining techniques; KDD process and its stages; data types including numerical, categorical, binary and time-series data; dataset characteristics such as dimensionality, sparsity and distribution; data quality issues including missing values, noise and outliers; noise analysis and data cleaning techniques; data pre-processing methods including aggregation and sampling; dimensionality reduction techniques; feature selection approaches; data transformation methods such as normalization and standardization; discretization techniques for continuous data. **(TEXT BOOK 3: CHAPTER 1 , 2, 3; TEXT BOOK 1: CHAPTER 1, 2; TEXT BOOK 2: CHAPTER 1)**	
MODULE – II	**09 Hours**
SIMILARITY MEASURES AND ASSOCIATION RULE MINING: Concept of similarity and dissimilarity in data analysis; distance measures including Euclidean and Manhattan distance; set-based similarity measures including Jaccard index and Dice coefficient; rule-based reasoning and decision-making systems; association rule mining concepts including support, confidence and lift; Apriori algorithm for frequent itemset mining; pattern mining techniques including frequent and sequential pattern mining; model selection techniques; evaluation metrics; error analysis including bias-variance trade-off, overfitting and underfitting. **(TEXT BOOK 3: CHAPTER 5, CHAPTER 6, 7; TEXT BOOK 1: CHAPTER 3, 4; TEXT BOOK 2: CHAPTER 2, 3)**	
MODULE – III	**09 Hours**
EXPLORATORY DATA ANALYSIS AND PREDICTIVE MODELLING: EDA concepts and importance; data summarization and visualization (histograms, scatter plots, box plots); correlation and covariance analysis; clustering techniques (K-means, hierarchical); outlier detection methods; regression analysis (simple and multiple linear regression); model assumptions and diagnostics; ANOVA; parametric and non-parametric models; classification basics; frequent pattern mining concepts. ***(TEXT BOOK 2: CHAPTERS 4, 5, 6; TEXT BOOK 3: CHAPTERS 10, 4; TEXT BOOK 1: CHAPTERS 5, 6, 7)***	
MODULE – IV	**09 Hours**
ADVANCED PREDICTIVE MODELLING: Discriminant analysis for classification problems and decision boundaries; time-series analysis concepts including trend, seasonality and stationarity; ARIMA models for forecasting; autocorrelation function (ACF) and partial autocorrelation (PACF); autoregressive (AR) and moving average (MA) models; factor analysis for dimensionality reduction and latent variable identification; matrix factorization techniques for data representation; recommendation systems including collaborative filtering and content-based filtering approaches. **(TEXT BOOK 2: CHAPTER 7, 9; TEXT BOOK 3: CHAPTER 8, 9; TEXT BOOK 1: CHAPTER 8)**	
MODULE – V	**09 Hours**
DATA MINING APPLICATIONS AND ADVANCED TOPICS: Text mining concepts including text preprocessing, tokenization, TF-IDF and sentiment analysis; spatial and temporal data mining for location-based and time-dependent patterns; privacy and security issues in data analytics including data protection and secure processing; ethical considerations such as bias, fairness, transparency and responsible use of AI systems. **(Text Book 2: Chapter 8; Text Book 3: Chapter 12; Text Book 1: Chapter 9)**	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Explain data mining concepts and pre-processing techniques.	L2
2	Apply similarity measures and pattern mining techniques.	L3
3	Develop predictive models using statistical methods.	L3
4	Evaluate model performance and data-driven decisions.	L5
5	Critically assess AI-generated outputs in data analysis.	L5

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	1	1										2	3	1
CO2	2	2	1									2	3	1
CO3	3	2	2	1								3	3	1
CO4	2	2	2	2							1	3	3	1
CO5	2	2	2	2							1	3	3	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Textbooks

1. Bruce Ratner – Statistical and Machine Learning Data Mining
2. Dursun Delen – *Predictive Analytics*
3. Data Mining: Concepts and Techniques by Jiawei Han, Micheline Kamber, Jian Pei

Reference Textbooks

1. Introduction to Data Mining by Pang-Ning Tan, Michael Steinbach, Vipin Kumar
2. An Introduction to Statistical Learning by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani
3. Python for Data Analysis by Wes McKinney
4. Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow by Aurélien Géron
5. Mining of Massive Datasets by Jure Leskovec, Anand Rajaraman, Jeffrey Ullman

DATA MINING AND PREDICTIVE MODELLING

LAB EXPERIMENT

Total Contact Hours: 15

Following are experiments to be carried out using Python:

1. Implement data pre-processing techniques such as handling missing values, normalization, and encoding for a given dataset.
2. Perform feature selection and ranking on a dataset and analyse the impact on model performance.
3. Develop an association rule mining system using the Apriori algorithm and evaluate support and confidence.
4. Compute similarity measures (Euclidean, Jaccard, Dice) for a given dataset and compare results.
5. Perform clustering analysis on a dataset and identify meaningful patterns and outliers.
6. Build a regression-based predictive model and evaluate its performance using appropriate metrics.
7. Implement discriminant analysis for classification and analyze decision boundaries.
8. Apply factor analysis on a dataset and interpret latent factors.
9. Develop a recommendation system using collaborative filtering techniques.
10. Analyze a real-world dataset and build a complete predictive modelling pipeline (EDA → pre-processing → model → evaluation).

DEEP LEARNING

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Course Code	:	24AD3503	**Credits**	:	04
Hours / Week	:	03 Hours (Theory) + 02 Hours (Lab)	**Total Hours**	:	45 (T) + 15 (L)

L–T–P–J: 3–0–2–0

Course Learning Objectives: This Course will enable students to:

1. Describe the fundamental concepts of artificial neural networks, including perceptrons, multilayer perceptrons, activation functions, and evaluation techniques used in deep learning.
2. Illustrate the learning mechanisms of neural networks, including forward and backward propagation, gradient descent variants, and optimization techniques used in training models.
3. Apply regularization techniques, weight initialization methods, and hyperparameter tuning strategies for building well-generalized deep learning models.
4. Analyze major deep learning architectures such as CNNs, RNNs, LSTMs, GRUs, and attention-based models for vision and sequential data applications.
5. Examine advanced deep learning approaches including autoencoders, graph neural networks (GNNs), and explainable AI techniques with awareness of ethical and responsible AI principles

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

MODULE – I Introduction to Neural Networks and Deep Learning	**08 Hours**
Introduction to Artificial Intelligence, Machine Learning, and Deep Learning: Differences and Relationships. Foundations of Neural Networks and Deep Learning: Neural Networks, Activation functions, Loss Functions, Hyperparameter, The Perceptron, Multilayer perceptron (MLP). **Engineering Applications**: Image classification and object detection, speech recognition, predictive maintenance and fault diagnosis in industrial systems, medical diagnosis and healthcare analytics, autonomous systems and robotics.	
MODULE – II Mathematical Background, Data Preparation and Training Techniques	**10 Hours**
Data representation using tensors: Scalars, vectors, matrices, tensor shapes, data batches, image data, and introduction to sequence/time-series data representation. The gears of neural networks: Tensor operations—Element-wise operations, Broadcasting, Tensor product, and Tensor reshaping. Training the Model:Forward and Backward Propagation:forward propagation; backpropagation using the chain rule;Training Loop - epochs and batches, forward pass, loss computation, backpropagation, parameter update and validation steps;Gradient descent methods: batch, stochastic, and mini-batch gradient descent;Optimization algorithms: Momentum and Adam. Model Optimization and Generalization: Weight initialization: overview of Xavier/Glorot and He initialization; Regularization techniques: L2 weight decay, dropout, early stopping based on validation performance, and basic concepts of batch normalization; Data augmentation; Hyperparameter tuning using validation data. **Engineering Applications**:Time-series forecasting and demand prediction, anomaly detection in cybersecurity and network monitoring, energy load forecasting in smart grids, traffic flow prediction and intelligent transportation systems, signal processing and sensor data analysis.	
MODULE – III Convolutional Neural Networks (CNNs)	**09 Hours**
Biological inspiration of CNNs,Convolution operation: kernels/filters, feature maps, stride, and padding (with simple numerical examples). Convolutional layers: parameter sharing, local connectivity, and multiple filters/channels. Pooling layers: max pooling and average pooling; role of batch normalization and dropout in CNNs. CNN architecture: overall structure and working of convolutional neural networks, including input, convolutional, pooling, and fully connected layers. Popular CNN architectures (conceptual overview): LeNet-5 (architecture and idea of early CNN); AlexNet (representation learning—brief—and architecture); VGG (VGG blocks and network design); ResNet (residual blocks and model architecture). Transfer learning: use of pre-trained models (VGG16, ResNet50) for feature extraction and fine-tuning. Object detection (introduction): bounding boxes and YOLO (concept and intuition).	

Engineering Applications: Autonomous driving and ADAS, remote sensing and satellite imagery, biometric authentication (fingerprint/iris recognition), agricultural monitoring using drones, and CNN-based access control systems (Case Study:"Personalized Doggy Door").

MODULE – IV Recurrent Neural Networks, LSTMs and Attention Mechanism	**09 Hours**

Need for Sequential Models: limitations of MLP and CNN for sequential data (text, time-series, audio).

Recurrent Neural Networks (RNNs): architecture, hidden state, and unrolling through time. Vanishing Gradient Problem: challenges in learning long-term dependencies in RNNs. Long Short-Term Memory (LSTM): cell state, input, forget, and output gates. Gated Recurrent Unit (GRU): simplified gating mechanism and comparison with LSTM.

Attention Mechanism: limitations of fixed context vectors in Seq2Seq models; queries, keys, and values; Bahdanau (additive) attention-conceptual understanding.

Engineering Applications: Machine translation systems, predictive maintenance using sensor sequence data, anomaly detection in financial transactions, human activity recognition using wearable sensors, network traffic analysis and intrusion detection.

MODULE – V Unsupervised Deep Learning and Applications	**09 Hours**

Unsupervised Pretrained Networks (UPNs): motivation and learning from unlabelled data.

Autoencoders: encoder–decoder architecture, bottleneck/latent space, applications in compression and denoising, Variational Autoencoders, Introduction to Generative Adversarial Networks (GANs)

Graph Neural Networks (GNNs): basic concepts of graph-structured data, learning on graphs, and applications.

Explainable AI (XAI): importance of model interpretability; Grad-CAM, LIME, and SHAP concepts. Responsible AI and Ethics: bias in datasets, fairness, data privacy, and societal impact of AI.

Engineering Applications: Graph-based analysis for smart city infrastructure and networked systems, data compression and denoising in communication and signal processing, explainable AI for reliable decision-making in critical systems, and development of fair, unbiased, and privacy-preserving AI solutions.

LAB EXPERIMENT

Total Contact Hours: 15

1. Create tensors and perform basic operations such as reshaping, indexing, and element-wise computations using a deep learning framework. Demonstrate simple forward computation without training.
2. Implement linear and logistic regression models using gradient descent. Build a basic training loop with forward pass, loss computation, and parameter updates. Evaluate model performance using appropriate metrics.
3. Training a Basic Neural Network

Train a neural network for image classification using the MNIST dataset and evaluate performance using loss and accuracy. Implement the model using NumPy/PyTorch and analyze training behavior using loss curves.

4. Convolutional Neural Network (CNN)

Design and train a Convolutional Neural Network (CNN) for image classification using MNIST or American Sign Language dataset. Compare performance in terms of accuracy and feature learning.

5. Data Augmentation

Apply data augmentation techniques such as rotation, flipping, and scaling to enhance the dataset and evaluate their impact on model generalization and performance.

6. Optimization and Regularization

Train a neural network using different optimizers (SGD, Adam) and apply regularization techniques such as dropout and batch normalization. Compare overfitting using training and validation curves.

7. Using Pre-trained Models

Implement an image classification system using a pre-trained model (e.g., ResNet). Perform inference on new images and understand how classification layers work in deep learning models.

8. Transfer Learning

Fine-tune a pre-trained model on a custom dataset and compare performance with training from scratch. Analyze convergence speed and generalization.

9. Sequence Modeling using RNN/LSTM

Train an RNN or LSTM model for sequence prediction (text or time-series). Evaluate performance and understand handling of sequential data.

10. Natural Language Processing using Pre-trained Models

Implement an NLP task such as text classification using a pre-trained language model. Perform tokenization and text segmentation during preprocessing and understand text masking in masked language modeling. Evaluate model performance using accuracy and loss. (e.g., sequence modeling / attention-based models, machine translation systems, BERT-based NLP models)

11. GNN

Implement a Graph Neural Network (GNN) for node classification on a graph dataset. Represent data using nodes and edges, train the model efficiently, and evaluate performance. Analyze the impact of graph structure and discuss real-world applications.

Project Component (Indicative, not limited to)

1. Develop an image classification model that interprets color images using deep learning techniques (CNN-based approach).
2. Design and implement a data generator with augmentation techniques to effectively train models on small datasets and improve generalization.
3. Build an efficient image classification system using transfer learning and feature extraction with pre-trained models (e.g., ResNet), and analyze improvements in training speed and accuracy.
4. Implement a deep learning solution that combines data augmentation and transfer learning to optimize model performance on limited data scenarios.
5. Explore and compare advanced neural network architectures (e.g., CNN variants, transformers) for image-based tasks and analyze their performance.
6. Design an end-to-end deep learning pipeline for image-based applications, including data preparation, model training, and performance evaluation.
7. Investigate recent advancements in deep learning architectures and apply one technique to improve model performance on a selected task.

Course Outcome		
Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Understand the basic concepts of Deep Learning, including ANN, activation functions, loss functions, and optimization techniques.	L2
2	Apply CNN architecture to solve image classification and computer vision problems.	L3
3	Apply RNN architecture to solve NLP and time series data analytics.	L3
4	Analyze the working of Transformer architecture and Large Language Models (LLMs).	L4
5	Design generative and time series models using autoencoders, Variational Autoencoders (VAEs), and Generative Adversarial Networks (GANs).	L6

Table: Mapping Levels of Cos to Pos / PSOs														
COs	Program Outcomes (Pos)											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO 1	3	2	1	1	2	–	–	1	2	1	2	3	3	3
CO 2	3	3	2	1	3	-	–	1	2	2	3	3	3	3
CO 3	2	3	3	2	3	-	1	2	2	2	3	3	3	3
CO 4	2	3	2	2	3	2	-	2	2	2	3	3	3	3
CO 5	1	2	2	2	2	3	3	2	2	2	3	2	3	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Text Books:

1. Aston Zhang, Zack C. Lipton, Mu Li, Alex J. Smola, "Dive into Deep Learning", Cambridge University Press, ISBN-10**:** 1009389432,2023**.**
2. Josh Patterson and Adam Gibson, "Deep Learning: A Practitioner's Approach", O'Reilly Media, ISBN:978-1-4919-1425-0, 2017.
3. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning (Adaptive Computation and Machine Learning series", MIT Press, 2017.

Additional Resources

1. Aurelien Geron, "Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow", O'Reilly, 3rd Edition, 2022.
2. Francois Chollet, "Deep Learning with Python", Manning Publications, 2nd Edition, 2021.
3. Simon Haykin, "Neural Networks: A Comprehensive Foundation", 2nd Edition, Prentice Hall, 2003.
4. Nikhil Buduma, "Fundamentals of Deep Learning", O'Reilly Media, 2017.

IOT FUNDAMENTALS AND ITS APPLICATIONS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Course Code	:	24AD3504	**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hours
L–T–P–J : 3–0–0–0					

Course Learning Objectives: This Course will enable students to:

1. Understand the fundamental concepts, evolution, and architecture of IoT systems and their role in real-world applications.
2. Identify and explain the function of sensors, actuators, microcontrollers, and embedded devices used in IoT systems.
3. Explain communication protocols and data flow mechanisms including Wi-Fi, Bluetooth, and MQTT in IoT environments.
4. Explore and analyze IoT applications across domains such as smart homes, healthcare, agriculture, and smart cities using case studies.
5. Develop a basic understanding of IoT system design, cloud integration, data analytics, and security considerations.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

MODULE – I	09 Hours
INTRODUCTION TO IoT: Definition and characteristics of IoT; evolution of IoT; IoT in daily life (smart homes, wearables); basic IoT architecture (3-layer model: perception, network, and application layers); benefits and challenges of IoT adoption; IoT ecosystem overview; AI-assisted visualization of IoT architecture and real-world deployment scenarios. (Text Book 1: Chapter 1; Text Book 2: Chapter 1)	
MODULE – II	**09 Hours**
IoT DEVICES – SENSORS AND ACTUATORS: Types of sensors (temperature, motion, light, humidity, pressure); types of actuators (LED, motor, buzzer, relay); microcontrollers and embedded platforms (Arduino overview); Input–Process–Output (IPO) model; basic interfacing concepts between sensors and controllers; real-world applications including environmental monitoring and smart lighting; AI-assisted identification and classification of IoT sensors and devices. (Text Book 1: Chapters 2, 3; Text Book 2: Chapter 2)	
MODULE – III	**09 Hours**
COMMUNICATION IN IoT: Basics of IoT communication; wired vs wireless communication technologies; Wi Fi and Bluetooth protocols in IoT; LoRa and Zigbee communication technologies; Internet based communication and IP addressing; MQTT protocol (publish subscribe model, broker, topics); data transmission flow in IoT systems from device to cloud to user; networking fundamentals for IoT. (Text Book 1: Chapter 4; Text Book 2: Chapters 3, 4)	
MODULE – IV	**09 Hours**
IoT DATA AND CLOUD BASICS: IoT data generation and lifecycle; introduction to cloud computing and cloud service models; IoT dashboards and data visualization tools; edge computing vs. cloud computing; basics of IoT data analytics and pattern recognition; popular IoT cloud platforms (conceptual overview); AI-based dashboard generation and data visualization for IoT monitoring systems. (Text Book 1: Chapters 5, 6; Text Book 2: Chapter 5)	
MODULE – V	**09 Hours**
IoT APPLICATIONS AND SECURITY: IoT applications across domains: smart home, smart city, smart agriculture, healthcare IoT, and industrial IoT; benefits and limitations of IoT deployment; security challenges in IoT (privacy, data integrity, device authentication); future trends in IoT (5G, AI integration, digital twins); case studies: smart irrigation system and healthcare wearable monitoring; AI-based case study analysis and ethical reflection on IoT data usage. (Text Book 1: Chapters 7, 8; Text Book 2: Chapters 6, 7)	

Course Outcome		
Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Understand the basic concepts, evolution, and architecture of IoT systems.	L2
2	Identify IoT components such as sensors, actuators, and embedded devices and explain their functioning.	L2
3	Explain communication mechanisms and data flow in IoT systems including Wi-Fi, Bluetooth, and MQTT.	L2
4	Analyze IoT applications across domains using real-world case studies.	L4
5	Design simple IoT-based solutions applying the Input–Process–Output model and system design principles.	L5

Table: Mapping Levels of COs to POs / PSOs														
COs	**Program Outcomes (POs)**											**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO1	3	3							1			2	1	
CO2	3	2	1		2							3	2	1
CO3	2	3	1		2				1			2	2	1
CO4	2	3	2	1	2	2	1		2			2	2	3
CO5	2	2	3	2	3	2	1	2	2	1	2	3	3	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Note: This course requires no prior programming or electronics background and is designed for learners from all disciplines. Students are encouraged to explore IoT simulation platforms such as Tinkercad and freely available NPTEL IoT course materials to supplement classroom learning.

Textbooks

1. Arshdeep Bahga & Vijay Madisetti – Internet of Things: A Hands-on Approach
2. Raj Kamal – Internet of Things: Architecture and Design

Reference Textbooks

1. Olivier Hersent, David Boswarthick & Omar Elloumi – The Internet of Things: Key Applications and Protocols
2. Honbo Zhou – The Internet of Things in the Cloud: A Middleware Perspective
3. NPTEL IoT Courses – nptel.ac.in
4. Industry IoT documentation and tutorials (Arduino, ThingSpeak, AWS IoT Core)

TIME SERIES ANALYSIS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Course Code	:	24AD3505	**Credits**	:	04
Hours / Week	:	03 Hours (Theory) + 02 Hours (Lab)	**Total Hours**	:	45(L) + 15(P) Hours
L–T–P-J	:	3–0–2-0			

Course Learning Objectives: This Course will enable students to:

1. To introduce the fundamental concepts, components, and graphical representation of time series data.
2. To develop an understanding of smoothing, decomposition, and exponential smoothing techniques for forecasting.
3. To analyse stationary and non-stationary time series using correlation and spectral analysis methods.
4. To apply AR, MA, ARMA, ARIMA, and state-space models for time series modelling and forecasting.
5. To enable students to solve real-world engineering, business, and scientific forecasting problems using time series techniques.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

MODULE – I : INTRODUCTION TO TIME SERIES & FORECASTING FUNDAMENTALS	09 Hours
Definition and scope of time series, classification of time series, components of time series including trend, seasonal, cyclical and irregular components, additive and multiplicative models, forecasting methods versus forecasting models, single-step and multi-step forecasting, point and interval forecasting, white noise process. Time series graphics including time plots, seasonal plots, seasonal subseries plots, scatter plots, lag plots, autocorrelation function (ACF) plots and partial autocorrelation function (PACF) plots. Simple forecasting methods including mean method, naïve method, seasonal naïve method and drift method. Forecast accuracy measures including MAE, RMSE, MAPE, scale-dependent and scale-independent measures, and interval forecast accuracy measures. Applications in demand forecasting, stock market analysis, weather prediction, production planning and inventory control. **(TEXT BOOK 1: CHAPTER 1, CHAPTER 2; TEXT BOOK 2: CHAPTER 1, CHAPTER 2, CHAPTER 3, CHAPTER 5)**	
MODULE – II: SMOOTHING, DECOMPOSITION & EXPONENTIAL SMOOTHING METHODS	**09 Hours**
Moving average smoothing, simple exponential smoothing, Holt's linear trend method, Holt–Winters trend and seasonal methods, ETS (Error-Trend-Seasonal) framework, taxonomy of exponential smoothing models, innovation state-space models, classical decomposition techniques, estimation and elimination of trend and seasonal components, additive and multiplicative decomposition models. Forecasting techniques including ARAR algorithm, Holt–Winter algorithm and Holt–Winter seasonal algorithm. Applications in signal smoothing, sales forecasting, energy demand prediction and load forecasting. **(TEXT BOOK 1: CHAPTER 3, CHAPTER 4; TEXT BOOK 2: CHAPTER 8, CHAPTER 9)**	
MODULE – III : STATIONARY PROCESSES, CORRELATION & SPECTRAL ANALYSIS	**09 Hours**
STATIONARY PROCESSES, CORRELATION & SPECTRAL ANALYSIS: Stationarity and differencing, MODULE roots in time series, stationary stochastic processes, linear processes, properties of sample mean and autocorrelation, autocorrelation function (ACF), partial autocorrelation function (PACF), forecasting stationary time series. Spectral analysis including spectral density functions, time-invariant linear filters and spectral density of ARMA processes. Applications in noise filtering, system stability analysis, financial time series modelling and signal processing. **(TEXT BOOK 1: CHAPTER 5, CHAPTER 6; TEXT BOOK 2: CHAPTER 9, CHAPTER 10)**	
MODULE – IV:ARIMA, SEASONAL MODELS & STATE-SPACE METHODS	**09 Hours**
ARIMA, SEASONAL MODELS & STATE-SPACE METHODS: Autoregressive (AR) models, moving average (MA) models, ARMA models, ARIMA models, Seasonal ARIMA (SARIMA) models, regression with ARMA errors, Box–Jenkins methodology including model identification, parameter estimation, diagnostic checking and forecasting, order selection techniques, preliminary estimation methods and maximum likelihood estimation. State-space models including state-space representations, basic structural models, state-space representation of ARIMA models, Kalman filtering and recursions, and estimation in state-space models. Applications in economic forecasting, predictive maintenance, traffic flow modelling and climate modelling. **(TEXT BOOK 1: CHAPTER 7, CHAPTER 8, CHAPTER 9; TEXT BOOK 2: CHAPTER 10, CHAPTER 11)**	

MODULE – V: ADVANCED TIME SERIES METHODS & APPLICATIONS	09 Hours
Multivariate time series analysis including second-order properties of multivariate time series, estimation of mean and covariance, multivariate ARMA models, best linear predictors, modeling and forecasting multivariate series, cointegration and Granger causality. Machine learning approaches for time series forecasting including regression-based forecasting, tree-based forecasting models, neural network approaches, Long Short Term Memory (LSTM) Networks for Time Series Forecasting, and hybrid forecasting models. Probabilistic forecasting including parametric and non-parametric models. **(TEXT BOOK 1: CHAPTER 10; TEXT BOOK 2: CHAPTER 12, CHAPTER 13)**	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Understand the basic concepts, components, graphical methods, and forecasting fundamentals of time series data.	L2
2	Apply smoothing, decomposition, and exponential smoothing techniques for trend and seasonal forecasting.	L3
3	Analyze stationary and non-stationary processes using ACF, PACF, differencing, and spectral analysis techniques.	L4
4	Develop and evaluate AR, MA, ARMA, ARIMA, SARIMA, and state-space models for forecasting applications.	L5
5	Apply advanced time series techniques including multivariate analysis, machine learning methods, probabilistic forecasting, and fuzzy forecasting to real-world applications.	L3

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	3	2		1	2				1		2	2	1	1
CO2	2	3	2	2	3				1	1	2	2	2	1
CO3	2	3	1	3	3				1		2	3	2	1
CO4	3	3	2	3	3	1		1	1	1	3	3	3	2
CO5	3	3	3	3	3	2	1	1	1	1	3	3	3	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Note: To achieve a thorough comprehension and in-depth knowledge of theoretical principles for topics on regression, classification and neural network, students are encouraged to refer the textbook authored by Christopher M Bishop. (Text book 1 in Reference books)

Text Books

1. Peter J. Brockwell and Richard A. Davis, Introduction to Time Series and Forecasting, Springer, 3rd Edition, 2016.
2. Rob J. Hyndman and George Athanasopoulos, Forecasting: Principles and Practice, OTexts, 3rd Edition, 2021.

Reference Books

3. Douglas C. Montgomery, Cheryl L. Jennings and Murat Kulahci, Introduction to Time Series Analysis and Forecasting, Wiley, 2nd Edition, 2015.
4. Ruey S. Tsay, Analysis of Financial Time Series, Wiley, 3rd Edition, 2010.
5. Robert H. Shumway and David S. Stoffer, Time Series Analysis and Its Applications with R Examples, Springer, 4th Edition, 2017.

LAB EXPERIMENT

Total Contact Hours: 30

Following are experiments to be carried out using Python:

1. Implementation of time series data visualization and identification of components (trend, seasonal, cyclical, irregular) using real datasets.
2. Application of moving average and simple exponential smoothing techniques for forecasting time series data.
3. Implementation of Holt's linear trend method and Holt–Winters seasonal forecasting method.
4. Time series decomposition using additive and multiplicative decomposition models.
5. Stationarity testing and differencing of time series data using statistical methods.
6. Computation and analysis of Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots.
7. Building and analysis of Autoregressive (AR), Moving Average (MA), and ARMA models using statistical tools.
8. Implementation of ARIMA and Seasonal ARIMA (SARIMA) models for forecasting applications.
9. Implementation of LSTM Based Time Series Forecasting using Python.
10. Forecast evaluation using performance measures such as MAE, RMSE, and MAPE.
11. Mini project on real-world time series forecasting applications using machine learning or multivariate time series techniques.

RESPONSIBLE AI, ETHICS & POLICY

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Course Code	:	24AD3601	**Credits**	:	02
Hours / Week	:	02 Hours	**Total Hours**	:	30 Hours
L–T–P–J	:	2–0–0–0			

Course Learning Objectives: The course will enable students to:

1. Understand ethical frameworks and principles relevant to the design and deployment of artificial intelligence technologies.
2. Analyze the societal, legal, and policy implications of AI adoption across different sectors.
3. Apply the principles of Responsible AI in the development and evaluation of AI systems.
4. Evaluate and design AI solutions that are ethical, transparent, and socially responsible.
5. Design responsible AI strategies and governance mechanisms aligned with global standards and regulatory frameworks.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. **Lecture method** means it includes not only traditional lecture method, but different type of teaching methods may be adopted to develop the course outcomes.
2. **Interactive Teaching:** Adopt the Active learning that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show **Video/animation** films to explain functioning of various concepts.
4. Encourage **Collaborative** (Group Learning) Learning in the class.
5. To make **Critical thinking**, ask at least three Higher order Thinking questions in the class.
6. Adopt **Problem Based Learning**, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyze information rather than simply recall it.
7. Show the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every **concept can be applied to the real world** - and when that possible, it helps improve the students' understanding.

MODULE – I	06 Hours
Foundation of Responsible and Authentic Intelligence **Topics:** Concept of Responsible AI and Authentic Intelligence, Principles of Authentic Intelligence, Human-centred AI, trust, transparency, fairness, accountability, and inclusivity and its societal impact, Ethical frameworks for AI, Human values and cultural contexts in AI systems.	
MODULE – II	**06 Hours**
Bias, Fairness, Transparency and Human-Aligned AI **Topics:** Sources of bias in AI, data bias, algorithmic bias, societal bias, Fairness metrics and bias mitigation techniques, Algorithmic transparency and explainability, Explainable AI (XAI) models and interpretability methods.	
MODULE – III	**06 Hours**
Privacy, Safety and Trustworthy AI Systems **Topics:** Privacy challenges in AI-driven systems, Privacy-preserving machine learning, differential privacy, federated learning, AI security risks and adversarial attacks, Robustness, safety, and reliability of AI models, Trustworthy AI lifecycle frameworks.	
MODULE – IV	**06 Hours**
AI Governance, Regulation and Responsible Innovation **Topics:** AI governance and regulatory frameworks, Algorithmic accountability and auditing mechanisms, AI risk assessment and impact evaluation models, Global AI policy initiatives and standards, Responsible innovation in AI development.	
MODULE – V	**06 Hours**
Designing Authentic and Responsible AI Systems **Topics:** Human-centered AI design methodologies, Responsible AI development lifecycle, Social and economic implications of emerging AI technologies, Sustainable and inclusive AI development	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Explain key concepts and principles of Responsible AI, including fairness, transparency, accountability, and privacy	L2
2	Analyze ethical challenges and societal impacts associated with AI technologies	L4
3	Evaluate AI systems for bias, reliability, and ethical risks using appropriate frameworks	L5
4	Interpret AI governance models, regulatory guidelines, and policy frameworks related to responsible AI for ethical governance	L3
5	Apply responsible AI principles in the design and development of trustworthy AI solutions	L3

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	3	2	1	-	-	3	1	3	-	-	-	1	3	3
CO2	2	3	2	2	-	3	2	3	-	-	-	1	2	2
CO3	3	3	2	2	3	3	1	3	1	-	-	1	3	3
CO4	2	3	2	2	2	3	2	3	1	-	-	2	2	2
CO5	3	2	3	3	3	3	2	3	2	-	2	2	3	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXTBOOKS:

1. M. Coeckelbergh - AI ethics, 2020, MIT press, ISBN: 978-0262538190.
2. S. Agarwal & S. Mishra - Responsible AI Implementing Ethical and Unbiased Algorithms, 2021, Springer, https://doi.org/10.1007/978-3-030-76860-7.

REFERENCE BOOKS:

1. S. Voeneky, P. Kellmeyer, O. Mueller, and W. Burgard - The Cambridge handbook of responsible artificial intelligence: interdisciplinary perspectives, eds., 2022, Cambridge University Press, ISBN: 9781009207867 / 9781009207898, DOI: 10.1017/9781009207898.

E Resources:

1. Virginia Dignum – *Responsible AI.* https://people.cs.umu.se/virginia/docs/responsibleAI2018.pdf
2. Luciano Floridi – *The Ethics of AI*. https://www.researchgate.net/publication/385157886_The_Ethics_of_Artificial_Intelligence

****

APPLIED NLP AND GENERATIVE AI

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Course Code	:	24AD3602	**Credits**	:	04
Hours / Week	:	03 Hours (Theory) + 02 Hours (Lab)	**Total Hours**	:	45 (Th) + 15 (P) Hours
L–T–P–J : 3–0–2–0					

Course Learning Objectives: This Course will enable students to:

1. Understand fundamental concepts of NLP and language processing.
2. Apply text representation and machine learning techniques.
3. Build NLP applications using deep learning models.
4. Use transformer-based models and generative AI tools.
5. Critically evaluate AI-generated outputs and ethical implications.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

MODULE – I	09 Hours
FOUNDATIONS OF NATURAL LANGUAGE PROCESSING: Introduction to NLP and real-world applications; NLP pipeline and workflow; text preprocessing (noise removal, lowercasing, punctuation handling); tokenization, stop word removal, stemming and lemmatization; POS tagging and parsing basics; N-grams and language modeling; Bag of Words and TF-IDF; word embeddings — Word2Vec (CBOW, Skip-gram), GloVe, FastText; Similarity and Distance Measures: cosine, Euclidean, Jaccard, edit distance (Levenshtein), semantic similarity using embeddings; applications in information retrieval, semantic search, text clustering, and RAG foundations	
MODULE – II	**09 Hours**
CLASSICAL NLP — APPLICATIONS & TEXT CLASSIFICATION: Text classification: pipeline, feature extraction, and model training (Naive Bayes, Logistic Regression, SVM); sentiment analysis: lexicon-based and ML-based approaches, polarity detection; Named Entity Recognition (NER): entity types, sequence labeling concepts; information extraction: relation extraction and event detection basics; topic modeling: Latent Dirichlet Allocation (LDA) — intuition and usage; machine translation basics: rule-based vs statistical approaches (overview); spell correction: edit-distance-based correction;	
MODULE – III	**09 Hours**
TRANSFORMERS, LLMS & HUGGING FACE ECOSYSTEM: Limitations of RNNs and LSTMs; attention mechanism and scaled dot-product attention; self-attention, multi-head attention, and positional encoding; Transformer architecture — encoder and decoder stacks, feed-forward layers; encoder-only models (BERT) — masked language modeling, pre-training, use cases; decoder-only models (GPT family) — autoregressive language modeling; encoder-decoder models (T5, BART) — sequence-to-sequence tasks; evolution of LLMs: GPT-3, GPT-4, LLaMA, Mistral, and the open-source landscape; transfer learning — pre-training and fine-tuning paradigm; Hugging Face Transformers — pipeline API, AutoModel, AutoTokenizer; NLP tasks: text classification, summarization, question answering, NER, zero-shot classification, semantic search, and text generation.	
MODULE – IV	**09 Hours**
PROMPT ENGINEERING & LLM APPLICATION DEVELOPMENT: Introduction to Generative AI — foundation models and the LLM ecosystem; open-source vs closed-source LLMs: GPT-4, Claude, Gemini, LLaMA 3.1, Mistral; GenAI use cases: coding assistants, content generation, summarization, customer support; prompt design principles — clarity, context, specificity, and output format; prompting strategies: zero-shot, few-shot, chain-of-thought (CoT), role prompting, and system prompts; hallucination — causes, detection, and reduction; OpenAI API — chat completions and key parameters; LangChain — LCEL, Runnables, chains, and conversational memory.	

MODULE – V	09 Hours
RAG, ADVANCED GENAI & RESPONSIBLE AI: Retrieval-Augmented Generation (RAG) — architecture and use cases; sentence transformers and semantic embeddings for retrieval; vector databases — FAISS, ChromaDB, Pinecone; document chunking and indexing; semantic search pipeline and nearest neighbor retrieval; context injection and knowledge-grounded generation; RAG pipeline development with LangChain; fine-tuning basics — when to fine-tune vs RAG; parameter-efficient fine-tuning: LoRA, QLoRA (conceptual); multimodal AI — CLIP and image-text tasks; diffusion models and image generation (overview); LLM inference and quantization (INT8, GGUF);	
LAB EXPERIMENT	**30 Hours**
1. Perform text preprocessing using Python libraries such as NLTK and spaCy, including tokenization, stopword removal, stemming, and lemmatization. 2. Implement n-gram language models and analyze word frequency distributions on a given text corpus. 3. Develop a Part-of-Speech (POS) tagging system using NLP libraries and evaluate tagging accuracy. 4. Implement text representation techniques such as Bag of Words (BoW) and TF-IDF for document analysis. 5. Design and implement a sentiment analysis system using machine learning models such as Naive Bayes or Logistic Regression. 6. Develop a document similarity system using cosine similarity and vector representations. 7. Generate and visualize word embeddings using Word2Vec, GloVe, or fastText models. 8. Implement sequence models using RNN/LSTM for text classification or sequence prediction tasks. 9. Use transformer-based models from Hugging Face Transformers for text generation and summarization. 10. Develop a Retrieval Augmented Generation (RAG) Based Question Answering System using LLMs and Vector Databases. 11. Design prompts using zero-shot, one-shot, and few-shot prompting techniques and compare model outputs. **Open-Ended Experiments:** *The following open-ended experiments encourage students to explore, innovate, and apply Applied NLP and Generative AI concepts to solve real-world problems using modern AI tools and frameworks.* 1. AI-Based Resume Screening System: Design an NLP-based system that analyzes resumes and matches them with job descriptions using similarity measures and keyword extraction techniques. Expected exploration: TF-IDF / embeddings, cosine similarity, resume ranking.	

Course Outcome		
Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Explain Applied NLP concepts, preprocessing techniques, and NLP pipelines.	L1, L2
2	Apply text representation methods and machine learning models for NLP tasks.	L2, L3
3	Develop NLP-based systems such as classification and similarity models.	L3, L4
4	Use Generative AI models, transformer architectures, and prompt engineering techniques effectively.	L3, L4
5	Evaluate ethical, reliability, and fairness aspects of AI and Generative AI systems.	L4, L5

Table: Mapping Levels of COs to POs / PSOs														
COs	**Program Outcomes (POs)**											**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO1	3	2			2			1			1	2	1	
CO2	3	3	2	1	3			1	1		1	3	2	1
CO3	3	3	3	2	3				1		2	3	3	1
CO4	2	3	3	2	3	1	1	1	2	1	2	3	3	2
CO5	2	2	2	2	2	2	3	2	2	1	3	2	2	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Primary Textbooks

- Daniel Jurafsky & James H. Martin – Speech and Language Processing.
- Jacob Eisenstein – Introduction to Natural Language Processing.
- Ian Goodfellow, Yoshua Bengio & Aaron Courville – Deep Learning.
- David Foster – Generative Deep Learning.

Suggested Textbooks

- Steven Bird, Ewan Klein & Edward Loper – Natural Language Processing with Python.
- Yoav Goldberg – Neural Network Methods in Natural Language Processing.
- Aurélien Géron – Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow.
- James Phoenix & Mike Taylor – Prompt Engineering for Generative AI.

Additional Resources

- Hugging Face documentation and tutorials.
- OpenAI documentation and API guides.
- Stanford University NLP course materials.

HIGH-PERFORMANCE COMPUTING

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Course Code	: 24AD3603	**Credits**	: 03
Hours/Week	: 03 Hours	**Total Hours**	: 45 Hours
L–T–P–J	: 3–0–0–0		

Course Learning Objectives: This course will enable students to:

1. Understand the fundamentals of parallel and distributed computing architectures used in high-performance systems.
2. Apply parallel programming models and tools to develop efficient high-performance applications.
3. Analyze and optimize algorithms for scalability, load balancing, and improved computational performance.
4. Use HPC techniques to solve complex problems in scientific, engineering, and data-intensive applications.
5. Evaluate emerging HPC technologies, including GPU computing, cloud HPC, and exascale computing systems.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture method, but different *type of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding

MODULE – I	**09 Hours**

Topics: Era of Computing, Parallel Computing, Multiprocessors and Multicomputer Architectures, Scalar VS Vector Processing, Multisector and Superscalar Machines, Pipelined Processors, SIMD Computers, Conditions of parallelism, Program flow mechanisms, Types of Parallelism – ILP, PLP, LLP, Program

Partitioning and Scheduling. Weather forecasting, climate modeling, cryptography and molecular dynamics. **Case Study:** HPC for accurate forecasting and climate simulations.	
MODULE – II	**09 Hours**
Topics: Era of Computing, Scalable Parallel Computer Architectures, towards low-cost computing, Network of Workstations project by Berkeley, Cluster Computing Architecture, Components, Cluster Middleware and SSI, Need of Resource Management and Scheduling, Programming Environments. Computational fluid dynamics, finite element analysis, robotics and autonomous systems. **Case Study:** HPC-driven digital twins to simulate factory operations	
MODULE – III	**09 Hours**
Topics: Clustering Models, Clustering Architectures, Clustering Architectures key factors, types of clusters, Mission critical Vs Business Critical Applications, Fault Detection and Masking Algorithms, Check pointing, Heartbeats, Watchdog Timers, Fault recovery through Failover and Failback Concepts. Bioinformatics, Big Data Processing, Web and Cloud Services. **Case Study:**GPU clusters for training deep learning models	
MODULE – IV	**08 Hours**
Topics: Introduction to High-Speed Networks, Lightweight Messaging Systems, Xpress Transport Protocol, Software RAID and Parallel File systems, Load Balancing Over Networks – Algorithms and Applications, Job Scheduling approaches and Resource Management in Cluster. Cloud and Data Center Computing, real-Time Systems and IoT, Telecommunications and Network Routing.	
MODULE – V	**10 Hours**
Topics: Introduction to CUDA architecture for parallel processing, CUDA Parallelism Model, Foundations of Shared Memory, Introduction to CUDA-C, Parallel programming in CUDA-C, Thread Cooperation and Execution Efficiency, Constants memory and events, memory management, CUDA C on multiple GPUs, Hashing and Natural Parallelism, Scheduling and Work Distribution, Atomics, Barriers and Progress, Transactional Memory. Deep Learning and AI in Engineering, Image and Signal Processing. **Case Study:** CUDA-based acceleration is used in MRI and CT image reconstruction, enabling faster processing and improved diagnostic imaging.	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1.	Explain the fundamentals of parallel computing, parallel architectures, and performance metrics used in High Performance Computing systems.	L1 & L2
2.	Apply parallel programming models and message passing techniques for solving computational problems.	L2 & L3
3.	Design and implement cluster computing and distributed computing solutions for scalable applications.	L3 & L4
4.	Analyze scheduling, synchronization, load balancing, and resource management techniques in HPC environments.	L4
5.	Evaluate GPU computing and CUDA-based optimization techniques for high-performance applications.	L4 & L5

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	2	2	1		2							3	2	1
CO2	3	3	2		2							3	2	1
CO3	3	3	3	2	3				1			3	3	2
CO4	2	3	3	2	3	1			1			3	3	2
CO5	2	2	3	3	3	2	1	2	2	1	2	3	3	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Textbooks

1. Rajkumar, High Performance Cluster Computing: Architectures and Systems, Vol. 1 Pearson Education.
2. Georg Hager and Gerhard Wellein, Introduction to High Performance Computing for Scientists and Engineers, CRC Press.

Additional Resources

1. Kai Hwang, Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw Hill International Editions.
2. NPTEL Course Name: High Performance Computing. Link: https://nptel.ac.in/courses/106/108/106108055/.
3. Course Name: High Performance Computing Architecture
4. Link: https://nptel.ac.in/courses/106/105/106105033
5. https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+T-AC-01+V1 (DLI)
6. https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-AC-10+V1 (DLI)

LAB EXPERIMENT

Total Contact Hours: 15

1. Matrix vector multiplication using OpenMP PARALLEL directive.
2. Sum of elements of one-dimensional real array A using OpenMP PARALLEL DO directive.
3. Compute the value of PI function by Numerical Integration using OpenMP PARALLEL section.
4. Using OpenMP, design, develop and run a multi-threaded program to generate and print Fibonacci Series. One thread must generate the numbers up to the specified limit and another thread must print them. Ensure proper synchronization.
5. Recursive computation using OpenMP CRITICAL section for the following code.

```
do i = 2, N

   a(i) = (a(i-1) + a(i))/2

end do
```

6. Finding largest element in a list of numbers using OpenMP CRITICAL section.
7. To calculate the sum of given numbers in parallel using MPI
8. Design a program that implements MPI Collective Communications
9. Design a MPI program that uses blocking send/receive routines.
10. Design a MPI program that uses nonblocking send/receive routines.
11. Design CUDA program to perform GPU-based matrix multiplication with many thread blocks.
12. Design CUDA program on GPU-based matrix multiplication using shared memory.

EDGE COMPUTING

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Course Code	: 24AD3604	**Credits**	: 03
Hours/Week	: 03 Hours	**Total Hours**	: 45 Hours
L–T–P–J	: 3–0–0–0		

Course Learning Objectives: This course will enable students to:

1. To understand the fundamentals of cloud and edge computing paradigms.
2. To analyze edge computing architectures, deployment models, and use cases.
3. To explore integration of edge computing with 5G and MEC.
4. To develop AI-based applications on edge devices.
5. To deploy and optimize intelligent systems using edge computing frameworks.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture method, but different *type of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

MODULE – I	09 Hours
Cloud Computing Basics & Edge Computing Introduction: Cloud Computing Basics (IaaS, PaaS, SaaS), Limitations of Cloud (Latency, Bandwidth, Privacy), Introduction to Edge Computing, Edge vs Cloud vs Fog Computing, Edge Architecture. **Engineering Applications:** Real-time IoT systems, smart devices, latency-sensitive applications.	
MODULE – II	**09 Hours**
Edge Computing Fundamentals: Edge Computing Basics, Edge Use-Cases, Benefits, Types of Edge (Device Edge, Local Edge, Cloud Edge), Edge Deployment Modes, Resource Management at Edge- **Engineering Applications:** Smart cities, industrial IoT, healthcare monitoring systems.	
MODULE – III	**09 Hours**
Edge Computing in 5G and MEC: 5G Overview, Edge in 5G Networks, Multi-access Edge Computing (MEC), Mobile Edge Computing, Network Slicing, Edge Orchestration- **Engineering Applications:** Autonomous vehicles, AR/VR systems, video streaming.	
MODULE – IV	**08 Hours**
Edge Intelligence and Real-Time Processing: Introduction to Edge Intelligence, Constraints of Edge Devices (compute, memory, power), Model Optimization for Edge (Quantization, Pruning, TensorRT basics), Lightweight Models for Edge (MobileNet, EfficientNet-lite – conceptual), Real-Time Data Processing at Edge, On-device Inference, Edge Analytics Pipelines, Event-driven Processing, Comparison of Cloud vs Edge AI Execution, Latency and Throughput Analysis.- **Engineering Applications:** Real-time video analytics, smart surveillance systems, healthcare monitoring devices, autonomous edge systems, industrial automation	
MODULE – V	**10 Hours**
Edge Deployment and Systems: Jetson Nano Setup, Docker and Containerization, NGC Setup, Riva Pipeline Deployment, Speech AI (ASR), Model Fine-tuning- **Engineering Applications:** Voice assistants, smart surveillance, AI-enabled robotics.	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Explain cloud and edge computing concepts and architectures.	L1
2	Analyze edge computing use cases, deployment modes, and benefits.	L3
3	Apply edge computing concepts to design AI-based real-time systems.	L4
4	Implement and deploy applications on edge devices using modern tools.	L5
5	Develop optimized, scalable edge solutions integrating AI and networking.	L5

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	3	1	1	-	2	-	-	-	-	-	1	2	1	1
CO2	3	3	2	2	2	-	-	-	1	-	1	2	2	-
CO3	3	3	3	2	3	-	-	-	1	-	1	3	3	-
CO4	3	2	3	3	3	1	-	-	1	1	2	3	3	1
CO5	3	3	3	3	3	1	1	1	1	1	-	-	-	1

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Textbooks

1. Weisong Shi, Schahram Dustdar – The Promise of Edge Computing, Computer, IEEE, 2016
2. Rajkumar Buyya, James Broberg, Andrzej Goscinski – Cloud Computing: Principles and Paradigms, Wiley, 2011
3. Perry Lea – Internet of Things for Architects, Packt Publishing, 2018.

Additional Resources

1. *David Jensen, JetsonHacks* – NVIDIA Jetson Nano Developer Kit Guide, NVIDIA Developer Documentation*, 2020*
2. *NVIDIA Deep Learning Institute* – Getting Started with AI on Jetson Nano, NVIDIA*, 2021*
3. *Kai Hwang, Geoffrey C. Fox, Jack J. Dongarra* – Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Morgan Kaufmann, 2012
4. *Satyanarayanan Mahadev* – Edge Computing: Vision and Challenges, IEEE Internet Computing, 2017

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

PROFESSIONAL ELECTIVE COURSES

5th Semester

BIG DATA COMPUTING					
[As per Choice Based Credit System (CBCS) scheme]					
SEMESTER – V					
Course Code	:	24AD3507	**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hours
L–T–P–J : 3–0–0–0					

Course Learning Objectives: This Course will enable students to:

1. Understand the fundamentals and architecture of Big Data systems.
2. Learn distributed storage and processing using Hadoop ecosystem.
3. Explore modern Big Data tools and frameworks.
4. Understand NoSQL databases and scalable data management.
5. Analyze real-world Big Data applications and ethical considerations.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

MODULE – I	09 Hours
Topics: Big Data definition and characteristics (5Vs); Types of data; Big Data vs traditional systems; Data sources; Big Data architecture; Challenges in Big Data; Distributed computing; Batch vs real-time processing; Big Data file formats (JSON, Avro, Parquet – basic idea); Data preprocessing basics (data cleaning and transformation concepts). **Engineering Applications:** Social Media Analytics, IoT Data Systems **Case Study:** Social Media Data Analysis Platform **(TEXT BOOK 1: CHAPTER 1, CHAPTER 2, CHAPTER 3; TEXT BOOK 2: CHAPTER 1, CHAPTER 2, CHAPTER 3)**	
MODULE – II	**09 Hours**
Topics: Introduction to Apache Hadoop; Hadoop ecosystem; HDFS architecture – NameNode, DataNode; Data replication and fault tolerance; Basic HDFS operations; Hadoop YARN architecture (resource management); HDFS read/write mechanism; Introduction to Hadoop MapReduce; Map and Reduce phases; Limitations of MapReduce. **Engineering Applications:** Distributed Log Processing, Data Storage Systems **Case Study:** Large-scale Log Data Processing using Hadoop **(TEXT BOOK 1: CHAPTER 2, CHAPTER 3, CHAPTER 4, CHAPTER 6; TEXT BOOK 2: CHAPTER 4, CHAPTER 5)**	
MODULE – III	**09 Hours**
Topics: Apache Spark architecture; RDDs and DataFrames; Transformations and Actions; Spark execution model (lazy evaluation, DAG); Introduction to Spark SQL; Performance comparison: Spark vs Hadoop; Hive; Pig; Data ingestion tools (Sqoop, Flume); Stream processing using Apache Kafka. **Engineering Applications:** Real-time Data Analytics, Recommendation Systems **Case Study:** Real-time Streaming Analytics System **(TEXT BOOK 1: CHAPTER 16, CHAPTER 17, CHAPTER 19; TEXT BOOK 2: CHAPTER 6, CHAPTER 7, CHAPTER 9, CHAPTER 10)**	
MODULE – IV	**09 Hours**
Topics: Introduction to NoSQL databases; CAP theorem; Types of NoSQL; Basics of MongoDB; Basics of Apache Cassandra; Data modeling in NoSQL; Consistency models (strong vs eventual consistency); Basic querying in MongoDB (CRUD operations – conceptual); NoSQL vs relational databases; Use cases. **Engineering Applications:** E-commerce Systems, Content Management Systems **Case Study:** Product Recommendation Database System **(TEXT BOOK 2: CHAPTER 11, CHAPTER 12, CHAPTER 13)**	

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

MODULE – V	09 Hours
Topics: Big Data analytics lifecycle; Machine learning overview; Real-time analytics; Data visualization concepts; Introduction to data pipelines (ETL concept); Dashboarding tools overview (Tableau/Power BI – conceptual); Cloud-based Big Data platforms (AWS/GCP overview); Applications (healthcare, finance, IoT); Case studies; Ethical issues, data privacy, and governance. **Engineering Applications:** Fraud Detection, Healthcare Analytics **Case Study:** Healthcare Big Data Analytics System **(TEXT BOOK 2: CHAPTER 17, CHAPTER 18)**	

Course Outcome		
Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Explain the fundamental concepts of Big Data, its characteristics, and distributed computing models.	L1, L2
2	Describe and interpret the architecture of Apache Hadoop and distributed storage systems.	L2, L3
3	Apply Big Data processing frameworks such as Apache Spark for data analysis tasks.	L3
4	Analyze different NoSQL databases and design appropriate data models for scalable systems.	L4
5	Evaluate Big Data applications, analytics workflows, and ethical considerations in real-world scenarios.	L5

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO1	3	2						1				2	1	
CO2	3	3	1		2							3	2	1
CO3	2	3	2		3				1			3	2	1
CO4	2	3	3	1	2	2	1					2	2	2
CO5	2	2	3	2	3	2	1	2	2	1	2	3	3	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Note: This course requires only basic programming and database concepts as prerequisites. Students are encouraged to explore the recommended tools and platforms listed below.

Textbooks

1. Tom White – Hadoop: The Definitive Guide
2. Seema Acharya – Big Data and Analytics

Additional Resources

1. NPTEL Big Data Courses
2. Coursera Big Data Specialization

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

DEVOPS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Course Code	:	24AD3508	**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hours

L–T–P–J : 3–0–0–0

Course Learning Objectives: This Course will enable students to:

1. 1To learn the key ideas and techniques used in DevOps to bring development and operations together to produce higher-quality software and deliver it more quickly
2. To gain a thorough understanding of Git, enabling them to efficiently manage source code changes, collaborate seamlessly with team members, and maintain a clean and organized project history
3. To design, implement, and manage Continuous Integration and Continuous Delivery (CI/CD) pipelines using Jenkins
4. To develop the ability to efficiently containerize applications using Docker and orchestrate these containers with Kubernetes.
5. To automate infrastructure provisioning, configuration management, and application deployment using Ansible, alongside automating web application testing using Selenium

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps

improve the students' understanding.	
MODULE – I	**09 Hours**
UNDERSTANDING DEVOPS CONCEPTS: Introduction to SDLC- People Process and Tools- DevOps Primer - The history and evolution of DevOps - The DevOps Process - DevOps Goals - The benefits of DevOps - Agile vs. DevOps - Implementing Continuous Build, Inspection, Continuous Integration/Continuous Delivery/deployment, Continuous monitoring- Infrastructure as Code - Automation Testing Version Control - Tools we use in DevOps. (Text Book-1: Chapter 2)	
MODULE – II	**09 Hours**
SOURCE CODE MANAGEMENT: Definition and importance of SCM - What is Version Control? Version Control Tools - Git Introduction - Git Installation - Commonly used commands in Git - Branching and merging in Git - Merge Conflicts - Stashing, Rebasing-- Git Workflows - Working with Remote repository. Introduction to Build tools- Maven, Gradle, Ant, Unit Testing. Code quality tools-Sonarqube and Selenium. (Text Book-2: Chapter 2, 3)	
MODULE – III	**09 Hours**
CONTINUOUS INTEGRATION / CONTINUOUS DELIVERY AND CONTINUOUS DEPLOYMENT: CI/CD principles - Using Jenkins for CI/CD implementation - Jenkins Architecture Jenkins installation - Plugin Management in Jenkins - Overview of Maven - Maven project structure - Maven plugins - Project Object Model (POM) fundamental unit of work in Maven project - Maven build lifecycle - Adding external dependencies to maven pom.xml - Maven build and test project - Jenkins Security Management - Jenkins Deployment: Tomcat installation and configuration - Integration with GIT. (Ref Text Book-2: Chapter 2, 3, 4, 5, 7)	
MODULE – IV	**09 Hours**
CONTAINERIZATION AND CONTAINER ORCHESTRATION: What is a Docker - Dockers vs. Virtualization - Docker Architecture Docker installation - Working with container: How to Share and copy a container - Container Life Cycle - How to use Base Image and customize - Creation of Docker File - How to Publish Image on Docker Hub. Introduction to container orchestration - Kubernetes basics: architecture, components - Deploying applications on Kubernetes - Managing Kubernetes clusters. (Text Book-1: Chapter 10 & Ref Text Book-3: Chapter 1, 2, 3 & Ref Text Book-4: Chapter 9, 10)	

MODULE – V	**09 Hours**
CONFIGURATION MANAGEMENT AND TESTING: Introduction to configuration management - Infrastructure as Code - Ansible Architecture - Ansible Playbooks: About Playbooks - Playbook Language Example YAML - How to Write Playbooks - Tasks in Playbooks - Learn about using handlers, variables in the playbook - Ansible Roles - Ansible Galaxy. Manual Testing and its challenges - Introduction to Selenium - Selenium IDE - Selenium Remote Control - Selenium Webdriver - Selenium grid - What and why X-Path - Selenium Chrome Driver - Selenium Integration with Jenkins. (Ref Text Book-5: Chapter 1, 4, 5 & Ref Text Book-6: Chapter 4, 5, 9, 25)	

Course Outcome		
Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Analyze the need for DevOps and primary benefits of DevOps from perspective of business and IT teams to deliver quality software more rapidly.	L2
2	Judge and solve complex version control scenarios, contribute to collaborative projects, and maintain high quality codebases in any development environment.	L3
3	Model the automation of build, test, and deployment processes, ensuring that code changes are quickly and reliably integrated into production.	L3
4	Build Docker images, manage Docker containers, and use Docker Compose for multi container applications, ensuring consistent environments across development, testing, and production. Also set up and manage Kubernetes clusters, enabling efficient orchestration of containerized applications.	L4
5	Create Ansible playbooks for automating repetitive tasks, ensuring consistent and scalable environments. Also create Selenium scripts for automated browser testing, enhancing the reliability and efficiency of web applications.	L5

Table: Mapping Levels of COs to POs / PSOs														
COs	**Program Outcomes (POs)**											**PSOs**		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO1	2	2	1		2	1						2	2	1
CO2	3	3	2	1	2	1		1				3	3	2
CO3	3	3	3	2	2	1	1	2				3	3	3
CO4	3	3	2	2	2	1	1	2				3	2	3
CO5	2	3	2	1	3	2	2	1				3	2	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Textbooks

1. Osama Mustafa, A Complete Guide to DevOps with AWS: Deploy, Build, and Scale Services with AWS Tools and Techniques, Apress, 2023.
2. Scott Chacon and Ben Straub, Pro Git: Everything You Need to Know About Git, 2nd Edition, Apress, 2024.

References

1. Nigel Poulton and Pushkar Joglekar, The Kubernetes Book, 2022 Edition.
2. John Ferguson Smart, Jenkins: The Definitive Guide, O'Reilly Media, 2011.
3. James Turnbull, The Docker Book: Containerization Is the New Virtualization

Recommended Tools

1. Version Control: Git, GitHub, GitLab
2. CI/CD: Jenkins, GitHub Actions, GitLab CI
3. Containerization: Docker, Docker Compose
4. Orchestration: Kubernetes (Minikube / kind for local), Helm
5. Infrastructure as Code: Terraform, Ansible
6. Monitoring & Logging: Prometheus, Grafana, ELK Stack (Elasticsearch, Logstash, Kibana)
7. Distributed Tracing: Jaeger, OpenTelemetry
8. Security: OWASP Dependency-Check (SCA), SonarQube (SAST), HashiCorp Vault, Open Policy Agent (OPA)
9. Agile Project Tools: Jira, Trello, Azure Boards
10. AI Development Tools: GitHub Copilot, ChatGPT / Claude, Jupyter Notebook

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

KNOWLEDGE ENGINEERING

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Course Code	: 24AD3509	**Credits**	: 03
Hours / Week	: 03 Hours	**Total Hours**	: 45 Hours
L–T–P–J: 3–0–0–0			

Course Learning Objectives: This Course will enable students to:

1. Understand the principles of knowledge engineering including knowledge types, acquisition techniques, and the architecture of knowledge-based systems.
2. Represent knowledge using formal logic, semantic networks, frames, ontologies, OWL, RDF, and knowledge graphs for diverse application domains.
3. Apply reasoning mechanisms including forward and backward chaining, case-based reasoning, fuzzy logic, and Bayesian inference to solve domain-specific problems.
4. Design and analyze expert systems and knowledge-based systems, and apply NLP-based knowledge extraction and embedding techniques.
5. Explore semantic web technologies, knowledge graph construction pipelines, retrieval-augmented generation, and ethical responsibilities in knowledge engineering.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible,

it helps improve the students' understanding.	
MODULE – I	**09 Hours**
FOUNDATIONS OF KNOWLEDGE ENGINEERING: Introduction to knowledge engineering; knowledge types (declarative, procedural, heuristic); knowledge vs. data vs. information; knowledge acquisition techniques and sources; knowledge elicitation from domain experts (interviews, protocol analysis); knowledge lifecycle (creation, representation, use, and maintenance); knowledge-based systems overview and architecture; applications in expert systems, knowledge graphs, and chatbots; AI-assisted exploration of knowledge types and lifecycle diagrams. (Text Book 1: Chapters 1, 2)	
MODULE – II	**09 Hours**
KNOWLEDGE REPRESENTATION: Propositional logic: syntax, semantics, and inference rules; predicate logic: quantifiers and expressions; semantic networks: nodes, arcs, and inheritance; frames and scripts: structure, slots, and inheritance; production rules and rule-based systems; ontologies: definition, components, and types; OWL (Web Ontology Language): classes, properties, and restrictions; RDF, RDFS, and SPARQL basics; knowledge graphs and Neo4j introduction; AI-assisted ontology creation in Protégé and AI-generated RDF triples. (Text Book 1: Chapters 8, 9, 10; Text Book 2: Chapters 1, 2; Text Book 4: Chapters 1, 2, 3)	
MODULE – III	**09 Hours**
REASONING AND INFERENCE: Forward chaining: data-driven inference algorithm; backward chaining: goal-driven inference and Prolog basics; modus ponens and resolution-based inference; case-based reasoning (CBR): retrieve, reuse, revise, retain cycle; model-based reasoning for diagnosis and troubleshooting; uncertainty in knowledge systems: sources and representation; Bayesian reasoning: prior, likelihood, and posterior; fuzzy logic: fuzzy sets, membership functions, and linguistic variables; fuzzy inference systems (Mamdani and Sugeno methods); AI-assisted tracing of chaining inference and fuzzy membership function generation. (Text Book 1: Chapters 7, 11, 12, 13)	
MODULE – IV	**09 Hours**
EXPERT SYSTEMS AND KNOWLEDGE-BASED SYSTEMS: Expert system architecture: knowledge base, inference engine, explanation facility, and UI; introduction to CLIPS: facts, rules, agenda, and conflict resolution; classic expert systems: MYCIN, DENDRAL, and XCON case studies; knowledge acquisition bottleneck: challenges and solutions; NLP-based knowledge extraction: named entity recognition and relation extraction; knowledge graph embeddings: TransE and link prediction basics; question answering system architecture and knowledge integration; AI-assisted CLIPS rule generation and NLP-based knowledge extraction from text.	

(Text Book 2: Chapters 3, 4, 5, 6)	
MODULE – V	**09 Hours**
SEMANTIC WEB, KNOWLEDGE GRAPHS, AND EMERGING TRENDS: Semantic web architecture and linked data principles; DBpedia, Wikidata, and open knowledge graphs; SPARQL querying on real knowledge graphs; knowledge graph construction and population pipeline; GNN basics for knowledge graph reasoning; retrieval-augmented generation (RAG): integrating knowledge graphs with LLMs; ethical considerations in knowledge engineering (bias, privacy, accountability, responsible AI); AI-assisted SPARQL query generation and RAG pipeline design. (Text Book 3: Chapters 1, 2, 3; Text Book 4: Chapters 4, 5)	

Course Outcome		
Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Understand the fundamentals of knowledge engineering including knowledge types, acquisition techniques, knowledge lifecycle, and the architecture of knowledge-based systems.	L2
2	Represent knowledge using propositional logic, predicate logic, semantic networks, frames, ontologies (OWL), RDF, and knowledge graphs for diverse real-world domains.	L3
3	Apply reasoning mechanisms including forward and backward chaining, case-based reasoning, fuzzy inference systems, and Bayesian reasoning to solve domain-specific problems.	L3
4	Design and analyze expert systems and knowledge-based systems, and apply NLP-based knowledge extraction and knowledge graph embedding techniques.	L4
5	Evaluate semantic web technologies, knowledge graph construction pipelines, retrieval-augmented generation systems, and ethical considerations in knowledge engineering.	L5

Table: Mapping Levels of COs to POs / PSOs														
COs	**Program Outcomes (POs)**											**PSOs**		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO1	3	2	1		1							2	2	1
CO2	3	3	2	1	2			1				3	2	2
CO3	3	3	2	2	2	1		1				3	3	2
CO4	3	3	3	2	2	1	1	2				3	3	3
CO5	2	3	2	1	2	2	2	1				3	2	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Textbooks

1. Russell, S., & Norvig, P. (2022). Artificial Intelligence: A Modern Approach (4th ed.). Pearson. [s 1, 2, 3]
2. Giarratano, J. C., & Riley, G. (2005). Expert Systems: Principles and Programming (4th ed.). Course Technology. [4]
3. Hitzler, P., Krötzsch, M., & Rudolph, S. (2009). Foundations of Semantic Web Technologies. CRC Press. [5]
4. Allemang, D., & Hendler, J. (2011). Semantic Web for the Working Ontologist (2nd ed.). Morgan Kaufmann. [s 2, 5]

Additional Resources

1. Pan, J. Z. et al. (2023). Large Language Models and Knowledge Graphs: OpportMODULEies and Challenges. arXiv:2308.06374.
2. NVIDIA DLI: Building RAG Agents with LLMs (8 Hours) – developer.nvidia.com/dli
3. NVIDIA DLI: Augment Your LLM Using Retrieval Augmented Generation (1 Hour, Free)
4. NVIDIA DLI: Agentic AI Explained (1 Hour, Free)
5. Wikidata Query Service: https://query.wikidata.org/ | DBpedia SPARQL: https://dbpedia.org/sparql
6. Neo4j Documentation: https://neo4j.com/docs/ | Protégé: https://protege.stanford.edu/

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

ACOUSTICS, IMAGE AND VIDEO PROCESSING

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Course Code	: 24AD3510	**Credits**	: 03
Hours / Week	: 03 Hours	**Total Hours**	: 45 Hours
L–T–P–J : 3–0–0–0			

Course Learning Objectives: This Course will enable students to:

1. Understand fundamentals of acoustics and audio signal processing.
2. Apply image processing techniques for enhancement and analysis.
3. Analyze video data using advanced processing methods.
4. Explore feature extraction and pattern recognition in multimedia data.
5. Develop AI-based solutions for audio-visual applications.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

MODULE – I	**09 Hours**
Topics: Introduction to acoustics: sound waves, frequency, amplitude, decibels; Human auditory system; Digital audio fundamentals: sampling, quantization; Audio signal representation; Noise and filtering techniques; Introduction to speech processing. **Engineering Applications:** Speech recognition, audio enhancement, noise reduction. **Case Study:** Speech signal processing for voice assistants.	
MODULE – II	**09 Hours**
Topics: Digital image representation, image acquisition; Image enhancement: filtering, smoothing, sharpening; Histogram processing; Image transformations; Edge detection techniques. **Engineering Applications:** Medical imaging, satellite imaging, industrial inspection. **Case Study:** Image enhancement for medical diagnosis.	
MODULE – III	**09 Hours**
Topics: Image segmentation techniques: thresholding, region-based methods; Feature extraction: texture, shape, color features; Object detection basics; Introduction to pattern recognition. **Engineering Applications:** Object recognition, face detection, biometric systems. **Case Study:** Face detection system using image processing.	
MODULE – IV	**09 Hours**
Topics: Video fundamentals: frame representation, frame rate; Motion detection and estimation; Object tracking; Background subtraction; Video compression basics. **Engineering Applications:** Surveillance systems, autonomous driving, activity recognition. **Case Study:** Real-time object tracking in video streams.	
MODULE – V	**09 Hours**
Topics: Python libraries: OpenCV, Librosa, NumPy; Introduction to Deep Learning in multimedia; Audio Processing using NVIDIA libraries; Deep Learning for Image Analysis; Object Detection for Intelligent Video Analytics (IVA) (DLI). **Engineering Applications:** Multimedia analytics, smart surveillance, speech-to-text systems. **Case Study:** AI-based multimedia analytics system.	

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

Course Outcome		
Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Understand fundamentals of acoustics and audio signal processing techniques.	L2
2	Apply image processing techniques for image enhancement, filtering, and edge detection.	L3
3	Analyze audio and image signals using feature extraction and pattern recognition methods.	L4
4	Develop video processing algorithms for motion detection, object tracking, and video analytics.	L5
5	Build AI-based multimedia processing systems integrating audio, image, and video processing tools.	L6

Table: Mapping Levels of COs to POs / PSOs														
COs	**Program Outcomes (POs)**											**PSOs**		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO1	3	2	1		2							2	2	1
CO2	3	3	2	1	3							3	3	1
CO3	2	3	2	2	3			1				3	3	2
CO4	2	3	3	2	3	1		2	1			3	3	2
CO5	2	2	3	2	3	2		2	2	1		3	3	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Primary Textbooks

1. Gonzalez, R. C., & Woods, R. E. – Digital Image Processing, 4th Edition, Pearson, 2018.
2. Rabiner, L. R. – Fundamentals of Speech Recognition, 1st Edition, Prentice Hall, 1993.
3. Wang, Y., Ostermann, J., & Zhang, Y. Q. – Video Processing and Communications, 1st Edition, Prentice Hall, 2001.

Additional Resources

1. NVIDIA DALI Audio Processing: https://docs.nvidia.com/deeplearning/dali/user-guide/docs/examples/audio_processing/index.html (DLI)
2. NVIDIA Deep Learning for Healthcare Image Analysis: https://www.nvidia.com/content/dam/en-zz/Solutions/deep-learning/deep-learning-education/Deep-Learning-for-Healthcare-Image-Analysis.pdf (DLI)

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

PROFESSIONAL ELECTIVE COURSES

6th Semester

Cloud-Based AI Application Development

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Course Code	: 24AD3607	**Credits**	: 03
Hours / Week	: 03 Hours	**Total Hours**	: 45 hrs
L–T–P–J : 3–0–0–0			

Course Learning Objectives: This Course will enable students to:

1. 1. Understand the fundamentals of cloud computing and AWS architecture.
2. Explore AWS AI and Machine Learning services and their applications.
3. Learn techniques for managing and processing data in cloud environments.
4. Understand the concepts of building, training, and optimizing ML models on AWS.
5. Analyse methods for deploying scalable and secure AI applications in the cloud.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

MODULE – I	**09 Hours**
Introduction to Cloud Computing & AWS: Cloud Computing Fundamentals: IaaS, PaaS, SaaS, Benefits of Cloud Computing: Scalability, Elasticity, Cost-Efficiency, Overview of AWS Global Infrastructure, AWS Core Services: EC2, S3, IAM, VPC, Introduction to AWS Console and CLI	
MODULE – II	**09 Hours**
AI & ML Services in AWS: Overview of AWS AI/ML stack, Amazon SageMaker: Features and workflow, Pre-built AI services: Amazon Rekognition, Amazon Comprehend, Amazon Lex, Amazon Polly, Model Training vs inference on cloud.	
MODULE – III	**09 Hours**
Data Handling & Pre-processing on AWS: Data Storage using Amazon S3, Data pipelines with AWS Glue, Data querying using Amazon Athena, Data organisation and lifecycle policies, Handling large datasets.	
MODULE – IV	**09 Hours**
AI Model Design and Integration using AWS: Overview of AI model lifecycle in cloud environments, Selection of appropriate AI models for cloud deployment, Pre-built vs Custom ML Models, Concept of lightweight model design for cloud, Integration of trained/local ML with AWS services, Data input/output handling using S3.	
MODULE – V	**09 Hours**
Deployment of AI Models on AWS: Introduction to model deployment in cloud environment: Hosting AI models using ECI, Serverless deployment: AWS lambda, Amazon API Gateway, REST API for AI model inferences, Basic monitoring using Cloudwatch.	

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

Course Outcome		
Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Explain core cloud computing concepts and AWS services relevant to AI.	L2
2	Identify and describe AWS AI/ML services for different application scenarios.	L2
3	Apply knowledge of cloud-based data management for AI workflows.	L3
4	Analyze the process of training and optimizing machine learning models on AWS.	L4
5	Evaluate deployment strategies for scalable and secure cloud-based AI applications.	L5

Table: Mapping Levels of COs to POs / PSOs														
COs	**Program Outcomes (POs)**											**PSOs**		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO1	3	1	1	–	2	–	–	–	–	–	1	2	1	1
CO2	3	2	1	–	3	–	–	–	–	–	1	2	2	2
CO3	3	2	2	2	3	–	–	–	1	–	1	3	2	2
CO4	3	3	3	3	3	–	–	–	1	–	1	3	3	2
CO5	3	3	3	3	3	1	1	1	1	1	2	3	3	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Primary Textbooks

1. Rajkumar Buyya, Christian Vecchiola, Thamarai Selvi – *Mastering Cloud Computing*
2. Dan C. Marinescu – *Cloud Computing: Theory and Practice*

Additional Resources

1. NPTEL courses
2. Scopus-Indexed research articles
3. AWS Certification

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

MACHINE LEARNING OPERATIONS (MLOps)

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Course Code	:	24AD3608	**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hours

L–T–P–J : 3–0–0–0

Course Learning Objectives: This Course will enable students to:

1. Understand MLOps principles, the ML lifecycle, CI/CD/CT pipeline design, experiment tracking with MLflow, and data versioning with DVC for reproducible machine learning workflows.
2. Design and implement data pipelines with Apache Airflow, apply data validation using Great Expectations, build feature engineering workflows with Feast, and detect data drift in production.
3. Apply model evaluation strategies, manage model versions with MLflow registry, implement A/B testing, canary deployments, and design automated retraining triggers.
4. Deploy ML models as REST APIs using FastAPI, containerize with Docker, orchestrate with Kubernetes, and optimise models using ONNX and TensorRT for production inference.
5. Evaluate production ML systems for data and concept drift, design monitoring dashboards, apply DevSecOps practices, and assess responsible MLOps principles including fairness and sustainability.

Teaching-Learning Process (General Instructions)

These are sample pedagogical methods that teachers can adopt to accelerate attainment of various course outcomes. Lecture method includes not only traditional lectures but different types of teaching methods adopted to develop course outcomes. Interactive Teaching: Adopt Active Learning that includes brainstorming, group discussion, focused listening, formulating questions, note-taking, and collaborative problem-solving around real production ML failure scenarios. Show Video/animation demonstrations of MLOps pipelines, MLflow dashboards, Kubernetes deployments, and monitoring workflows. Encourage Collaborative (Group) Learning through end-to-end MLOps pipeline design workshops. For Critical Thinking, ask Higher Order Thinking questions such as "Why does an ML model that worked perfectly in the lab fail in production after two weeks?" to motivate drift detection, retraining, and the full MLOps lifecycle. Adopt Problem-Based Learning: start with real production ML failures before introducing pipeline components and tooling. Apply AI^2 Pedagogy: use AI tools to generate MLflow experiments, Airflow DAGs, Dockerfiles, and monitoring configurations, then guide students to critically evaluate the AI outputs for missing components, security gaps, and ethical implications. Discuss how every concept can be

applied to real-world production ML systems in fintech, healthcare, e-commerce, and SaaS.	
MODULE – I	**09 Hours**
FOUNDATIONS OF MLOps AND THE ML LIFECYCLE: MLOps principles and motivation; ML lifecycle and maturity model (Level 0, 1, 2); DevOps vs. DataOps vs. MLOps; CI/CD/CT pipeline design for machine learning; experiment tracking with MLflow (runs, metrics, artifacts); model registry (versioning, staging, and production promotion); DVC for data and model versioning; feature store concepts and Feast overview; AI-assisted exploration of MLOps pipeline architectures and AI-generated MLflow experiment logs. (Text Book 1: Chapters 1, 2, 3; Text Book 2: Chapters 1, 2)	
MODULE – II	**09 Hours**
DATA AND FEATURE ENGINEERING PIPELINES: Apache Airflow DAGs for ML pipeline orchestration (operators, sensors, XComs); data validation with Great Expectations (suites, checkpoints, expectations); feature engineering pipelines (batch and streaming); Feast feature store (entity definitions, feature views, materialisation); data drift detection: KS test, PSI, and Wasserstein distance; dataset versioning and data lineage tracking; ETL patterns for ML; production data quality monitoring and SLA design; AI-assisted Airflow DAG design and AI-generated Great Expectations validation suites. (Text Book 1: Chapters 4, 5; Text Book 2: Chapters 3, 4; Text Book 3: Chapters 1, 2)	
MODULE – III	**09 Hours**
MODEL LIFECYCLE MANAGEMENT AND DEPLOYMENT STRATEGIES: Model evaluation strategies and time-based splits; Optuna hyperparameter tuning with MLflow integration; MLflow Model Registry: lifecycle stages, access control, and webhooks; Weights & Biases for collaborative experiment tracking; A/B testing and statistical significance; canary releases and shadow deployment; champion-challenger model selection; automated retraining triggers (drift-based, schedule-based, performance-based); AI-supported A/B test design and AI-generated champion-challenger evaluation reports. (Text Book 1: Chapters 6, 7, 8; Text Book 2: Chapters 5, 6)	
MODULE – IV	**09 Hours**
MODEL DEPLOYMENT AND INFERENCE ENGINEERING: REST API deployment with FastAPI (endpoints, async request handling); Docker containerisation for ML (multi-stage Dockerfiles, non-root best practices); Kubernetes for ML (pods, services, deployments, Horizontal Pod Autoscaler); ONNX export and TensorRT INT8/FP16 model optimisation; batch vs. online inference patterns; serverless ML deployment; edge ML deployment basics (ONNX Runtime, TFLite); multi-model serving; AI-assisted Dockerfile and Kubernetes YAML generation and AI-generated FastAPI endpoint code. (Text Book 1: Chapters 9, 10)	

MODULE – V	**09 Hours**
MONITORING, GOVERNANCE, AND RESPONSIBLE MLOps: Model monitoring fundamentals (metrics, logs, traces); data drift and concept drift detection with Evidently AI; Prometheus and Grafana dashboards for ML observability; SHAP-based explainability in production; ML governance (model cards, audit trails, documentation standards); DevSecOps basics for ML (SAST, secrets management, dependency scanning); NVIDIA RAPIDS for GPU-accelerated data science (cuML, cuDF); responsible MLOps (fairness, sustainability, ethical AI deployment); AI-assisted Prometheus alert rule design and AI-generated model cards. (Text Book 1: Chapters 11, 12) **NVIDIA DLI Mapping:** Accelerating End-to-End Data Science Workflows; Best Practices in Feature Engineering for Tabular Data with GPU Acceleration.	

Course Outcome		
Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Understand MLOps principles, the ML lifecycle, CI/CD/CT pipeline concepts, experiment tracking with MLflow, and data versioning with DVC for reproducible machine learning workflows.	L2
2	Design and implement data pipelines with Apache Airflow, apply data validation using Great Expectations, build feature engineering workflows, and detect data drift in production.	L3
3	Apply model evaluation strategies, manage model versions with MLflow registry, implement A/B testing, canary deployments, and design automated retraining triggers.	L3
4	Deploy ML models as REST APIs using FastAPI, containerize with Docker, orchestrate with Kubernetes, and optimise models using ONNX and TensorRT for production inference.	L4
5	Evaluate production ML systems for data and concept drift, design monitoring dashboards, apply DevSecOps practices, and assess responsible MLOps principles including fairness and sustainability.	L5

Table: Mapping Levels of COs to POs / PSOs														
COs	**Program Outcomes (POs)**											**PSOs**		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO1	3	2	1		2	1						2	2	2
CO2	3	3	2	1	2	1		1				3	3	2
CO3	3	3	2	2	2	1	1	2				3	3	3
CO4	3	3	3	2	2	1	1	2				3	3	3
CO5	2	3	2	1	3	2	2	1				3	2	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Primary Textbooks

1. Huyen, C. (2022). Designing Machine Learning Systems. O'Reilly Media.
2. Alla, S., & Adari, S. K. (2021). Beginning MLOps with MLflow. Apress.
3. Kleppmann, M. (2017). Designing Data-Intensive Applications. O'Reilly Media. [2]

Additional Resources

1. Google Cloud MLOps Architecture: https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning
2. NVIDIA DLI: Accelerating End-to-End Data Science Workflows (8 Hours) – developer.nvidia.com/dli
3. NVIDIA DLI: Best Practices in Feature Engineering for Tabular Data with GPU Acceleration (2 Hours)
4. MLflow Documentation: https://mlflow.org/docs/ | DVC Documentation: https://dvc.org/doc
5. Evidently AI Documentation: https://docs.evidentlyai.com/ | Great Expectations: https://docs.greatexpectations.io/

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)
Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering
(Artificial Intelligence and Data Science)

SOFT COMPUTING TECHNIQUES

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Course Code	:	24AD3609	**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45

L–T–P–J: 3–0–0–0

Course Learning Objectives: This Course will enable students to:

1. Learn the various soft computing frameworks
2. Be familiar with various neural network frameworks
3. Learn Genetic programming
4. Be exposed to hybrid systems

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/Animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

MODULE – I	08 Hours
INTRODUCTION: Scope of soft computing, various components, description of Artificial neural networks, overview fuzzy logic, theory of genetic algorithms, theory of hybrid systems.	
MODULE – II	**08 Hours**
NEURAL NETWORKS: Fundamentals of neural network, basic models of ANN, learning and activation functions, basic fundamental McCulloch- Pitts neuron model **Learning Models:** Supervised learning networks, Adaline, Back propagation, Unsupervised learning network, Korhonen self-organizing feature maps networks	
MODULE – III	**09 Hours**
FUZZY SETS AND RELATIONS: Uncertainty and Imprecision - Chance vs ambiguity - Classical Sets and Fuzzy Sets - Classical Relations and Fuzzy Relations - Membership functions - Properties of Membership functions - Fuzzification and Defuzzification	
MODULE – IV	**06 Hours**
FUZZY LOGIC: Introduction to fuzzy logic, classical sets and Fuzzy sets, Classical relations and Fuzzy relations, Membership functions, Fuzzy arithmetic and Fuzzy measures, fuzzy decision making.	
MODULE – V	**08 Hours**
HYBRID SOFT COMPUTING TECHNIQUES: Introduction to hybrid systems – Integration of neural networks, fuzzy logic, and genetic algorithms – Neuro-fuzzy systems – Fuzzy-genetic systems – Neural-genetic systems – Case studies and applications of hybrid approaches in real-world problem solving	

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

Course Outcome		
Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Explain the principal components of soft computing that include ANN and genetic Algorithm	L2
2	Apply a suitable method of soft computing to solve a particular problem	L3
3	Understanding the significance of neural network in soft computing	L2
4	Analyse the Gentic algorithms and fuzzy logic for solving the optimization problems	L4
5	Implementing the soft computing techniques in real life applications	L3

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO1	3	1	1	-	1	-	-	-	-	-	-	2	1	1
CO2	3	2	2	1	2	-	-	-	1	-	-	3	2	2
CO3	3	2	1	1	2	-	-	-	-	-	-	3	2	1
CO4	3	3	2	3	2	-	-	-	1	-	-	3	3	2
CO5	3	3	3	3	3	1	-	-	1	1	1	3	3	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Textbooks

1. S.N Sivanandam and S N Deepa,"Principles of Soft Computing", Wiley India Pvt Ltd, 2011
2. J.S.R. Jang, CT Sun and E Mizutani, "Neuro Fuzzy and Soft Computing", PHI/Pearson Education 2004
3. 3.T.J. Ross, "Fuzzy Logic with Engineering Applications", McGraw Hill, 1995.
4. S. Rajasekaran and G.A.V. Pai, "Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis and Applications", PHI, 2007

Reference Books:

1. George J Klir, Ute St Clair, Bo Yuan,"Fuzzy Set Theory: Foundations and Application" Prentice HAll, 1997
2. David E Goldberg,"Genetic Algorithm in Search Optimization and Machine Learning "Pearson Education India 2013

Additional Resources

- https://archive.nptel.ac.in/courses/106/105/106105173/
- https://www.udemy.com/topic/fuzzy-logic/
- https://www.coursera.org/courses?query=neural%20networks

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

STREAM PROCESSING

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Course Code	: 24AD3610	**Credits**	: 03
Hours / Week	: 03 Hours	**Total Hours**	: 45 Hours
L–T–P–J : 3–0–0–0			

Course Learning Objectives: This Course will enable students to:

1. Understand stream processing fundamentals including data stream characteristics, event-driven architectures, the CAP theorem, Lambda and Kappa architectures, and the trade-offs between batch and stream processing.
2. Implement real-time data ingestion pipelines using Apache Kafka, configure producers and consumers, design partitioning strategies, and use Kafka Connect and Kafka Streams.
3. Build stream processing applications using Apache Flink (windowing, watermarks, stateful operations, exactly-once semantics) and Apache Spark Structured Streaming.
4. Design real-time analytics pipelines for IoT, fraud detection, clickstream, and monitoring use cases, and integrate real-time ML inference and anomaly detection on streaming data.
5. Deploy stream processing systems on Kubernetes, apply cloud-native streaming services, evaluate performance, and implement responsible data privacy practices for streaming systems.

Teaching-Learning Process (General Instructions)

These are sample pedagogical methods that teachers can adopt to accelerate attainment of various course outcomes. Lecture method includes not only traditional lectures but different types of teaching methods adopted to develop course outcomes. Interactive Teaching: Adopt Active Learning that includes brainstorming, group work, focused listening, formulating questions, note-taking, and collaborative pipeline design exercises using real streaming scenarios. Show Video/animation demonstrations of Kafka broker behaviour, Flink windowing operations, and real-time Grafana dashboards updating from live data streams. Encourage Collaborative (Group) Learning through end-to-end streaming pipeline design workshops and case study analysis. For Critical Thinking, ask Higher Order Thinking questions such as "How does Swiggy know your order status changes in real time?" to motivate event-driven architecture, Kafka, Flink, and stream-to-dashboard pipelines. Adopt Problem-Based Learning: start with real-world streaming failures before introducing distributed systems abstractions. Apply AI^2 Pedagogy: use AI tools to generate Kafka configurations, Flink DataStream code, watermark strategies, and deployment YAML, then guide students to critically evaluate AI outputs for correctness, configuration errors, and privacy implications. Discuss how every concept can be applied to production streaming systems in fintech, e-commerce, IoT, and healthcare.

MODULE – I	**09 Hours**
FOUNDATIONS OF STREAM PROCESSING: Batch vs. stream processing (latency, throughput, and trade-offs); data stream characteristics (unbounded, ordered, out-of-order events); event-driven architecture (events, streams, and message queues); CAP theorem for distributed streaming systems; record-at-a-time vs. micro-batch processing models; Lambda architecture (batch layer, speed layer, serving layer); Kappa architecture: simplifying with a stream-only approach; AI-assisted architecture diagram generation for Lambda and Kappa systems. (Text Book 1: Chapters 1, 2, 3, 11)	
MODULE – II	**09 Hours**
APACHE KAFKA FOR REAL-TIME DATA INGESTION: Kafka architecture (brokers, topics, partitions, offsets, replication factor); Kafka producers: message serialisation, partitioning strategies, and acknowledgements; Kafka consumers and consumer groups: offset management and rebalancing; Kafka topics: retention policies and log compaction; Kafka Connect: source and sink connectors for data integration; Kafka Streams: KStreams, KTables, and stateful transformations; Schema Registry with Avro: schema evolution and data contracts; Kafka security (SASL authentication, ACL authorisation, TLS encryption); AI-assisted Kafka producer/consumer code generation and partition design. (Text Book 2: Chapters 1, 2, 3, 4, 5, 6)	
MODULE – III	**09 Hours**
APACHE FLINK AND SPARK STRUCTURED STREAMING: Flink architecture (JobManager, TaskManager, dataflow graph); Flink DataStream API (sources, transformations, sinks); windowing strategies: tumbling, sliding, session, and global windows; event time vs. processing time and watermark strategies; stateful stream processing: keyed state and operator state; Flink checkpointing and exactly-once semantics; Spark Structured Streaming: DataFrame API, triggers, output modes, and watermarking; Flink vs. Spark Structured Streaming comparison and selection criteria; AI-assisted Flink DataStream code generation and windowing strategy recommendations. (Text Book 3: Chapters 1, 2, 3, 4, 5, 6; Text Book 4: Chapters 1, 2, 3)	
MODULE – IV	**09 Hours**
REAL-TIME ANALYTICS AND MACHINE LEARNING ON STREAMS: End-to-end real-time analytics pipeline design; IoT data streaming with MQTT protocol and sensor data pipelines; real-time fraud detection pipeline with streaming feature engineering (transaction velocity, merchant category deviation, geographic anomaly); real-time ML inference on streams using ONNX model serving with Flink; online feature computation for ML serving; streaming anomaly detection with statistical methods (Z-score, sliding window); stream-to-dashboard pipeline with Kafka, Flink, and Grafana; time-series databases: InfluxDB basics; AI-assisted real-time pipeline design and Flink ONNX inference integration. (Text Book 1: Chapters 10, 11; Text Book 3: Chapters 7, 8)	

MODULE – V	**09 Hours**
RELIABILITY, CLOUD DEPLOYMENT, AND RESPONSIBLE STREAMING: Exactly-once semantics: idempotent producers, transactions, and checkpoint barriers; backpressure handling and flow control in streaming systems; performance tuning: parallelism, partitioning, and state backends; cloud-native streaming services (AWS Kinesis, GCP Pub/Sub, Azure Event Hubs); deploying Flink and Kafka on Kubernetes (Flink Operator, Strimzi); NVIDIA RAPIDS for GPU-accelerated stream analytics (cuDF, cuML); AI-assisted stream processing with LLMs for log parsing and anomaly explanation; responsible streaming: data privacy, GDPR compliance, and consent in real-time systems. (Text Book 1: Chapters 8, 9; Text Book 3: Chapters 9, 10) **NVIDIA DLI Mapping:** Accelerating End-to-End Data Science Workflows; Best Practices in Feature Engineering for Tabular Data with GPU Acceleration.	

Course Outcome		
Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Understand the fundamentals of stream processing, event-driven architectures, data stream characteristics, the CAP theorem, and the differences between batch and stream processing paradigms.	L2
2	Apply Apache Kafka for real-time data ingestion, design topic partitioning strategies, configure producers and consumers, and implement fault-tolerant publish-subscribe messaging pipelines.	L3
3	Implement stream processing applications using Apache Flink and Apache Spark Structured Streaming with windowing, stateful operations, watermarks, and exactly-once processing semantics.	L3
4	Design real-time analytics pipelines for IoT, fraud detection, and monitoring use cases, and apply real-time ML inference, feature computation, and anomaly detection on streaming data.	L4
5	Evaluate stream processing system performance, apply fault tolerance and exactly-once semantics, deploy stream applications on cloud platforms, and assess responsible streaming AI practices.	L5

Table: Mapping Levels of COs to POs / PSOs														
COs	**Program Outcomes (POs)**											**PSOs**		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO1	3	2	1		2	1						2	2	1
CO2	3	3	2	1	2	1		1				3	3	2
CO3	3	3	3	2	2	1	1	2				3	3	3
CO4	3	3	2	2	2	1	1	2				3	3	3
CO5	2	3	2	1	3	2	2	1				3	2	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Textbooks

1. Kleppmann, M. (2017). Designing Data-Intensive Applications. O'Reilly Media. [All s]
2. Narkhede, N., Shapira, G., & Palino, T. (2021). Kafka: The Definitive Guide (2nd ed.). O'Reilly Media. [2]
3. Hueske, F., & Kalavri, V. (2019). Stream Processing with Apache Flink. O'Reilly Media. [3]
4. Chambers, B., & Zaharia, M. (2018). Spark: The Definitive Guide. O'Reilly Media. [3]

Additional Resources

1. Apache Flink Documentation: https://flink.apache.org/docs/
2. Apache Kafka Documentation: https://kafka.apache.org/documentation/
3. Confluent Developer: https://developer.confluent.io/
4. NVIDIA DLI: Accelerating End-to-End Data Science Workflows (8 Hours) – developer.nvidia.com/dli
5. NVIDIA RAPIDS Documentation: https://docs.rapids.ai/
6. Google Dataflow / Apache Beam: https://beam.apache.org/

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)
Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering
(Artificial Intelligence and Data Science)

OPEN ELECTIVE COURSES

Business Analytics in Fintech

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Course Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hours
L–T–P-J	:	3–0–0-0			

Course Learning Objectives: This Course will enable students to:

1. Understand the Fintech ecosystem, digital financial services landscape, regulatory environment, and the role of data analytics in driving Fintech business decisions.
2. Apply descriptive and diagnostic analytics including EDA, statistical testing, customer segmentation, cohort analysis, and dashboard design to financial datasets.
3. Implement predictive models for credit scoring, loan default prediction, fraud detection, and customer churn using classification and ensemble methods.
4. Apply time series forecasting, payment analytics, product recommendation, and NLP-based sentiment analysis to Fintech data and business problems.
5. Evaluate AI-driven Fintech innovations including RegTech, InsurTech, WealthTech analytics, responsible AI, and generative AI applications in financial services.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/Animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.

7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

MODULE – I	**09 Hours**
1: Fintech Ecosystem and Analytics Framework **Topics:** Fintech segments (payments, lending, WealthTech, InsurTech, RegTech); regulatory framework (RBI, SEBI, PCI-DSS, GDPR); open banking and PSD2; business analytics framework (descriptive, diagnostic, predictive, prescriptive); key Fintech KPIs (NPS, CAC, LTV, GMV). **Engineering Applications:** KPI dashboard design for a digital lending startup.	
MODULE – II	**09 Hours**
2: Descriptive and Diagnostic Analytics for Fintech **Topics:** EDA for financial datasets; A/B testing for Fintech; customer segmentation (RFM, K-Means); cohort analysis and CLV estimation; churn prediction (logistic regression); financial dashboard design with Tableau or Power BI; SQL aggregations and window functions for Fintech analytics. **Engineering Applications:** Customer churn analytics dashboard for a digital lending app.	
MODULE – III	**09 Hours**
3: Predictive Analytics and Credit Risk Modelling **Topics:** Credit scoring (CIBIL, FICO, alternative scoring); XGBoost and Random Forest for loan default prediction; class imbalance handling (SMOTE); model evaluation (AUC-ROC, KS statistic); fraud detection with supervised classification; SHAP for explainable credit decisions. **Engineering Applications:** Credit scorecard for peer-to-peer lending.	
MODULE – IV	**09 Hours**
4: Time Series, Payment Analytics, and NLP for Fintech **Topics:** Financial time series and ARIMA forecasting; UPI payment funnel analysis and drop-off detection; product recommendation (collaborative filtering basics); financial sentiment analysis with TF-IDF; FinBERT for financial sentiment classification. **Engineering Applications:** UPI transaction funnel analytics and revenue forecasting for a Fintech startup.	
MODULE – V	**09 Hours**
5: Advanced Fintech Analytics and Emerging AI Applications **Topics:** RegTech analytics (AML monitoring); InsurTech risk scoring; WealthTech and robo-advisors; responsible AI in Fintech (fairness, bias); GDPR and data privacy; generative AI and LLMs in Fintech (RAG-based Q&A, chatbots). **Engineering Applications:** LLM-powered financial advisory chatbot with RAG	

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Understand the Fintech ecosystem, digital financial services landscape, key players, regulatory frameworks (RBI, SEBI, PCI-DSS, GDPR), and data-driven business models in financial technology.	L2
2	Apply descriptive and diagnostic analytics techniques including EDA, statistical analysis, customer segmentation, and dashboard design to derive actionable insights from financial data.	L3
3	Implement predictive analytics models for credit scoring, loan default prediction, customer churn, and fraud detection using classification, regression, and ensemble techniques.	L3
4	Apply time series analysis, payment analytics, recommendation systems, and natural language processing for sentiment analysis and financial document intelligence in Fintech.	L3
5	Evaluate AI-driven Fintech innovations including algorithmic trading, RegTech, insurtech analytics, responsible AI, and ethical considerations in financial data analytics systems.	L5

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	2	2	1		2	1						2	2	1
CO2	3	3	2	1	2	1		1				3	3	2
CO3	3	3	2	2	2	1	1	2				3	3	3
CO4	3	3	2	2	2		1	2				3	2	3
CO5	2	3	2	1	3	2	2	1				3	2	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Textbooks

1. Provost, F., & Fawcett, T. (2013). Data Science for Business. O'Reilly Media.
2. McNeil, A. J., Frey, R., & Embrechts, P. (2015). Quantitative Risk Management. Princeton University Press. [3]
3. Chishti, S., & Barberis, J. (Eds.) (2016). The Fintech Book. Wiley. [1]
4. Huyen, C. (2022). Designing Machine Learning Systems. O'Reilly Media. [5]

Additional Resources

1. Yang, H. et al. (2020). FinBERT: A Pre-trained Financial Language Representation Model. arXiv:2006.08097.
2. RBI Guidelines on Digital Payments: https://www.rbi.org.in/
3. NPCI UPI Statistics and Reports: https://www.npci.org.in/
4. Kaggle Fintech Datasets: Give Me Some Credit, IEEE-CIS Fraud Detection, Home Credit Default Risk
5. Towards Data Science — Financial Analytics articles: https://towardsdatascience.com/

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)
Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering
(Artificial Intelligence and Data Science)

Foundations to Internet of Things

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Course Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45
L–T–P-J	:	3–0–0-0			

Course Learning Objectives:

1. This Course will enable students to:
2. Understand fundamental concepts and architecture of IoT systems.
3. Identify sensors, actuators, and embedded devices used in IoT.
4. Explain communication and data flow in IoT environments.
5. Explore IoT applications across different domains.
6. Develop basic IoT system design understanding.

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

MODULE – I	09 Hours
Introduction to IoT: **Topics**: Definition and characteristics of IoT, Evolution of IoT, IoT in daily life (smart homes, wearables), Basic IoT architecture (3-layer model), Benefits and challenges of IoT. **Engineering Applications:** Smart Home Systems, Wearable Devices **Case Study:** Smart Home Automation System	
MODULE – II	**09 Hours**
Topics: Sensors (temperature, motion, light), Actuators (LED, motor, buzzer), Microcontrollers (Arduino overview), Input–Process–Output model, Basic interfacing concepts. **Engineering Applications:** Environmental Monitoring, Smart Lighting Systems **Case Study:** Automatic Street Light System	
MODULE – III	**09 Hours**
Communication in IoT **Topics:** Basics of IoT communication, Wireless technologies (Wi-Fi, Bluetooth), Internet-based communication, MQTT protocol (conceptual), Data transmission flow in IoT systems. **Engineering Applications:** Wearable Devices, Smart Monitoring Systems **Case Study:** Smart Fitness Band	
MODULE – IV	**9 Hours**
IoT Data & Cloud Basics: Topics: IoT data generation and lifecycle, Introduction to cloud computing, IoT dashboards and visualization, Edge vs Cloud computing, Basics of IoT data analytics. **Engineering Applications:** Smart Monitoring Systems, Weather Forecasting **Case Study:** Smart Weather Monitoring System	
MODULE – V	**09 Hours**
IoT Applications & Security: **Topics:** Smart Home, Smart City, Smart Agriculture, Healthcare IoT, Benefits and limitations, Security issues (privacy, data risks), Future trends. **Engineering Applications:** Smart Cities, Healthcare Systems **Case Study:** Smart Irrigation System, Healthcare Wearable Monitoring	

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Understand the basic concepts, characteristics, and architecture of Internet of Things systems.	L1, L2
2	Identify IoT components such as sensors, actuators, embedded devices, and interfacing mechanisms	L1, L2
3	Explain communication mechanisms, wireless technologies, and data flow in IoT environments.	L2, L3
4	Analyze IoT applications and case studies across domains such as healthcare, agriculture, and smart cities	L4
5	Design simple IoT-based solutions considering	L3, L5

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)												PSOs		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**		**1**	**2**	**3**
CO1	**3**	**2**						1					2	**1**	
CO2	3	2	1		2								3	2	1
CO3	2	3	1		2			1					2	2	1
CO4	2	3	2	1	2	2	1		1				2	2	2
CO5	2	2	3	2	3	2	1	2	2	1	2		3	3	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Textbooks

1. Arshdeep Bahga & Vijay Madisetti – *Internet of Things: A Hands-on Approach*
2. Raj Kamal – *Internet of Things: Architecture and Design*.

Reference Textbooks

1. Arshdeep Bahga, Vijay Madisetti, *Internet of Things: A Hands-on Approach*, Universities Press.
   (Covers IoT fundamentals, architecture, sensors, communication, cloud integration, and applications)
2. Raj Kamal, *Internet of Things: Architecture and Design Principles*, McGraw Hill.
   (Covers IoT architecture, embedded systems, communication protocols, and security concepts)
3. Adrian McEwen, Hakim Cassimally, *Designing the Internet of Things*, Wiley.
   (Covers IoT design principles, smart devices, networking, and practical applications)
4. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry, *IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things*, Cisco Press.
   (Covers IoT communication technologies, MQTT, networking, and industrial IoT use cases)
5. Perry Lea, *Internet of Things for Architects*, Packt Publishing.
   (Covers IoT cloud platforms, data analytics, edge computing, and IoT system design)
6. Brian Russell, Drew Van Duren, *Practical Internet of Things Security*, Packt Publishing.
   (Covers IoT security, privacy, authentication, and risk management concepts)

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering

(Artificial Intelligence and Data Science)

Artificial Intelligence for Economics

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Course Code	:	**Credits**	: 03
Hours / Week	: 03 Hours	**Total Hours**	: 45
L–T–P–J : 3–0–0–0			

Course Learning Objectives: This Course will enable students to:

1. Understand the role of AI in economic systems and decision-making.
2. Learn basic machine learning concepts for economic applications.
3. Analyze economic data using AI techniques.
4. Explore AI-driven applications in finance, markets, and public policy.
5. Evaluate ethical and societal implications of AI in economics

Teaching-Learning Process (General Instructions)

Each consists of AI usage, AI evaluation, and human judgment. Adapting to personalized learning. Group projects and peer learning. AI-assisted evaluation of assignments and coding tasks. Research integration: literature survey on deep learning models; empirical study of different CNN architectures; study advanced language models such as PaLM and LLaMA; explore Python libraries for deep learning implementations; apply advanced DL topics to solve real-world complex problems.

MODULE – I	**09 Hours**

Introduction to AI in Economics

Topics: Introduction to Artificial Intelligence, Role of AI in Economics and Finance, AI vs Traditional Econometrics, Types of economic data (time-series, cross-sectional), Applications in banking, e-commerce, and public policy. **Engineering Applications:** Financial Systems, E-commerce Platforms

MODULE – II	**09 Hours**

Data & Predictive Models in Economics

Topics: Basics of Machine Learning (conceptual), Regression models, Classification models (Decision Trees), Time-series forecasting (GDP, inflation), Model evaluation basics..

Engineering Applications: Financial Forecasting, Economic Prediction Systems,

Case Study: Credit Risk Prediction in Banking

MODULE – III	**09 Hours**

Optimization & Decision Making **Topics:** Cost and profit optimization, Decision-making models, Pareto optimality (concept), AI-based decision systems, Applications in pricing and resource allocation. **Engineering Applications:** Business Optimization, Resource Allocation Systems	
MODULE – IV	**09 Hours**
Game Theory & Economic Intelligence Introduction to Game Theory, Strategic decision-making, Auctions and pricing strategies, Mechanism design (intro), AI in competitive markets**. Engineering Applications:** Market Design, Pricing Systems **Case Study:** Online Advertisement Auction System	
MODULE – V	**09 Hours**
Applications & Ethics **Topics:** Smart Home, Smart City, Smart Agriculture, Healthcare IoT, Benefits and limitations, Security issues (privacy, data risks), Future trends. **Engineering Applications:** AI in Finance (fraud detection, risk prediction), AI in public policy, Recommender systems, Introduction to reinforcement learning (concept), Ethical issues: bias, fairness, transparency. **Engineering Applications:** Fraud Detection Systems, Recommendation Engines **Case Study:** Stock Price Prediction using AI	

Course Outcome		
Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Understand basic Artificial Intelligence concepts and their applications in economics and finance	L1, L2
2	Apply simple machine learning and predictive models to economic and financial datasets.	L2, L3
3	Analyze economic decision-making, optimization, and forecasting using AI-based models.	L3, L4
4	Evaluate AI applications in finance, auctions, pricing systems, and public policy.	L4, L5
5	Interpret ethical, fairness, transparency, and societal implications of AI-driven economic systems.	L4, L5

Table: Mapping Levels of COs to POs / PSOs														
COs	**Program Outcomes (POs)**											**PSOs**		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO1	3	2			1			1				2	1	
CO2	3	3	1	1	2				1		1	3	2	1
CO3	2	3	2	2	2	1		1	1		1	3	2	1
CO4	2	3	3	2	3	2	1	1	1	1	2	3	3	2
CO5	2	3	2	2	2	2	3	2	2	1	3	2	2	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Textbooks

1. Varian, H. – *Artificial Intelligence, Economics, and Industrial Organization*
2. Agrawal, Gans & Goldfarb – *Prediction Machines*

Additional Resources

- NPTEL AI for Economics courses
- Research articles on AI in finance and policy