RCOEM

Shri Ramdeobaba College of Engineering and Management, Nagpur

SHRI RAMDEOBABA COLLEGE OF ENGINEERING AND MANAGEMENT, NAGPUR – 440013

PROGRAMME SCHEME & SYLLABI

2023-24

B.Tech in Computer Science and Engineering (Data Science)

ABOUT THE DEPARTMENT:

The Data Science was established with the primary objective of imparting knowledge in the field of computer science and engineering with special emphasis on data science. The department is well-equipped with state-of-the-art infrastructure which includes latest configuration desktops organized in three different laboratories. Latest software, wireless access point, LCD projectors and separate routers are used in the laboratories for teaching purpose. The department has well qualified and highly experienced faculty with expertise in diversified areas of Data Science and AI ML with industry experience. Besides excellence in teaching, there is a committed pursuance of high-quality research and innovation. Faculty members of the department have been involved in research and patent filling. The curriculum is designed to impart students with in-depth knowledge and understanding of the key technologies in data science. Open Electives and MOOCs provide an opportunity for interdisciplinary learning. Option of full six months internship is provided in reputed industries and organizations for VIII semester students. Department takes pride in excellent internships and placements of the final year students and has the distinction of consistently getting good results in all semesters. The student club of the department, DASCA, organizes various technical and co- curricular activities which ensures all round development of the students.

DEPARTMENT VISION:

To achieve excellent standards of quality-education in computer science and engineering with a specialized focus on data science, foster innovation, nurture collaborative culture and contribute in the field of data science to serve the industry and society.

DEPARTMENT MISSION:

To empower the next generation of computer scientists and engineers with the knowledge, skills, and tools essential to excel in the rapidly evolving field of data science.

To impart the ability to devise globally recognized solutions for the problems of society and industry,

To provide an environment for interdisciplinary collaboration, research, and lifelong learning.

PROGRAM EDUCATION OBJECTIVES (PEOs):

1. Graduates will have strong foundation of knowledge and skills in mathematics, statistics, programming and computer science to solve problems in data science.
2. Graduates will have the ability and attitude to adapt to emerging technological changes with lifelong learning skills.
3. Graduates will demonstrate collaborative learning and spirit of team work through multidisciplinary Data Science projects ensuring ethical and moral values.
4. Graduates will demonstrate professionalism, ethical attitude and leadership skills with lifelong learning in their career.

PROGRAMME OUTCOMES (POs):

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

PROGRAMME SPECIFIC OUTCOMES (PSOs):

1. Apply the concepts and practical knowledge of data science in analysis, design and development of computing systems and applications to multi-disciplinary problems.

2. Acquaint with the contemporary trends in industrial, research settings and thereby innovate novel solutions to existing problems.

B. Tech. Computer Science and Engineering (Data Science) [2023-24]

Teaching & Evaluation Scheme [B.Tech CSE(DS)]

Semester–I

Sr. No.	Category	Course Code	Course Name	Hours/week			Credits	Maximum marks			ESE Duration (Hrs)
				L	T	P		Continuous Evaluation	End Sem Exam	Total	
1.	BSC	CHT1001	Chemistry of Smart Materials	2	0	0	2	50	50	100	2
2.	BSC	CHP1001	Chemistry of Smart Materials Lab	0	0	2	1	50	-	50	-
3.	BSC	MAT1002	Calculus	3	0	0	3	50	50	100	3
4.	ESC	CDT1001	Digital Electronics	3	0	0	3	50	50	100	3
5.	ESC	CDP1001	Digital Electronics Lab	0	0	2	1	50	-	50	-
6.	ESC	CDT1002	Programming for problem solving	3	0	0	3	50	50	100	3
7.	ESC	CDP1002	Programming for problem solving Lab	0	0	2	1	50	-	50	-
8.	VSEC	CDT1003	Computer Workshop – I	1	0	0	1	50	-	50	-
9	VSEC	CDP1003	Computer Workshop – I Lab	0	0	2	1	50	-	50	-
10	HSSM-IKS	HUT1001	Foundational Literature of Indian Civilization	2	0	0	2	50	50	100	2
11	CCA	PET1001	Sports-Yoga-Recreation	1	0	0	1	50	-	50	-
12	CCA	PEP1001	Sports-Yoga-Recreation Lab	0	0	2	1	50	-	50	-
TOTAL				**15**	**0**	**10**	**20**			**850**	**-**

Semester–II

Sr. No.	Category	Course Code	Course Name	Hours/week			Credits	Maximum marks			ESE Duration (Hrs)
				L	T	P		Continuous Evaluation	End Sem Exam	Total	
1.	BSC	PHT2001	Introduction to Quantum Computing	2	1	0	3	50	50	100	3
2.	BSC	PHP2001	Introduction to Quantum Computing Lab	0	0	2	1	50	-	50	-
3.	BSC	MAT2002	Discrete Mathematics	3	0	0	3	50	50	100	3
4.	BSC	MAP2001	Computational Mathematics Lab	0	0	2	1	50	-	50	-
5.	BSC	CHT2007	Bioinformatics	2	0	0	2	50	50	100	2
6.	ESC	CDT2001	Object Oriented Programming	3	0	0	3	50	50	100	3
7.	ESC	CDP2001	Object Oriented Programming Lab	0	0	2	1	50	-	50	-
8.	PCC	CDT2002	Computer Architecture	2	0	0	2	50	50	100	2
9.	VSEC	CDT2003	Computer Workshop – II	1	0	0	1	50	-	50	-
10.	VSEC	CDP2003	Computer Workshop – II Lab	0	0	2	1	50	-	50	-
11.	AEC	HUT2002	English for Professional Communication	2	0	0	2	50	50	100	2
12.	AEC	HUP2002	English for Professional Communication Lab	0	0	2	1	50	-		
13.	CCA	HUP0001	Liberal/Performing Art	0	0	2	1	50	-	50	-
14.	VEC	HUT2004	Foundational Course in Universal Human Values	1	0	0	1	50	-	50	-
TOTAL				**16**	**1**	**12**	**23**			**1000**	**-**

Exit option : Award of UG Certificate in Major with 43 credits and an additional 8 credits.			
Exit Courses			
1	**Web Designer**	**Online/offline certification Course**	**8**
2	**IT Support Engineer**		**8**
3	**Certified Programmer (language learned in Sem-1 and/or Sem-2 [C,C++,Java, Python])**		**8**

SEMESTER III

Sr. No.	Category	Course code	Course Name	Hours/ week			Credits	Maximum marks			ESE Duration (Hrs)
				L	T	P		Continuous Evaluation	End Sem Exam	Total	
1	PCC	CDT3001	Data Structures and Algorithms	3	1	0	4	50	50	100	3
2	PCC	CDP3001	Data Structures and Algorithms Lab	0	0	2	1	50	-	50	-
3	PCC	CDT3002	Theory of Computations	3	0	0	3	50	50	100	3
5	PCC	CDP3003	Programming Language Lab	0	0	4	2	50	-	50	-
6	MDM	MAT3002	Probability and Statistics	3	0	0	3	50	50	100	3
8	VSEC	CDP3004	Software Laboratory - I	0	0	2	1	50	-	50	-
7	OE	CDT2980	Open Elective - I	2	0	0	2	50	50	100	2
8	AEC	HUT3001	Business Communication	2	0	0	2	50	50	100	2
9	HSSM	CDP3006	Idea Lab	0	0	4	2	50	-	50	-
10	VEC	CDT3007	Cyber Laws and Ethics in IT	2	0	0	2	50	50	100	2
TOTAL				17	1	12	22			800	

Course Code	Open Elective - I
CDT2980-1	Massive Open Online Course(s) of Swayam
CDT2980-2	Massive Open Online Course(s) of Coursera

Semester - IV

Sr. No.	Category	Course code	Course Name	Hours/ week			Credits	Maximum marks			ESE Duration (Hrs)
				L	T	P		Continuous Evaluation	End Sem Exam	Total	
1	PCC	CDT4001	Operating Systems	3	0	0	3	50	50	100	3
2	PCC	CDP4001	Operating Systems Lab	0	0	2	1	50	-	50	-
3	PCC	CDT4002	Artificial Intelligence	3	0	0	3	50	50	100	3
4	PCC	CDP4002	Artificial Intelligence Lab	0	0	2	1	50	-	50	-
5	PCC	CDT4003	Design and Analysis of Algorithms	3	0	0	3	50	50	100	3
6	MDM	MAT4001	Linear Algebra	3	0	0	3	50	50	100	3
7	OE	IDT2990-2	Open Elective - II	3	0	0	3	50	50	100	3
8	VSEC	CDP4005	Software Laboratory - II	0	0	2	1	50	-	50	-
9	HSSM	HUT4003	Managerial Economics	2	0	0	2	50	50	100	2
10	VEC	HUT4002	Environment Education	2	0	0	2	50	50	100	2
11	CEP	CDP4006	Community Engagement Project	0	0	4	2	50	-	50	-
TOTAL				**18**	**0**	**10**	**24**	**550**	**350**	**900**	

Course Code	Open Elective - II
IDCDT2990-2	Massive Open Online Course(s) of Coursera

Exit option : Award of UG Diploma in Major with 90 credits and an additional 8 credits			
Exit Courses			
1	Application Development (Android, Advanced Java)	Online/offline certification Course	8
2	Certified Software Engineer (DevOps)		8

Semester- V

Sr. No.	Category	Course Code	Course Name	Hours/ week			Credits	Maximum marks			ESE Duration (Hrs)
				L	T	P		Continuous Evaluation	End Sem Exam	Total	
1	PCC	CDT5001	Database Management Systems	3	0	0	3	50	50	100	3
2	PCC	CDP5001	Database Management Systems Lab	0	0	2	1	50	-	50	-
3	PCC	CDT5002	Machine Learning	3	0	0	3	50	50	100	3
4	PCC	CDP5002	Machine Learning Lab	0	0	2	1	50	-	50	-
5	PCC	CDT5003	Computer Networks	3	0	0	3	50	50	100	3
6	PEC	CDT5004	Program Elective - I	3	0	0	3	50	50	100	3
7	MDM	CDT5005	Financial Analytics	3	0	0	3	50	50	100	3
8	MDM	CDP5005	Financial Analytics Lab	0	0	2	1	50	-	50	-
9	OE	IDCDT3980	Open Elective - III	3	0	0	3	50	50	100	3
TOTAL				**17**	**0**	**6**	**21**	**450**	**300**	**750**	

Course Code	Open Elective - III
IDCDT3980-1	Open Elective through NPTEL
IDCDT3980-2	Massive Open Online Course(s) of Coursera
IDCDT3980-4	Open Elective through Industry/International MoUs
IDCDT3980-5	Fundamentals of AWS Practitioner

Semester- VI

Sr. No.	Category	Course code	Course Name	Hours/ week			Credits	Maximum marks			ESE Duration (Hrs)
				L	T	P		Continuous Evaluation	End Sem Exam	Total	
1	PCC	CDT6001	Data Warehousing and Business Intelligence	3	0	0	3	50	50	100	3
2	PCC	CDP6001	Data Warehousing and Business Intelligence Lab	0	0	2	1	50	-	50	-
3	PCC	CDT6002	Deep Learning - I	3	0	0	3	50	50	100	3
4	PCC	CDP6002	Deep Learning - I Lab	0	0	4	2	50	-	50	-
5	PCC	CDT6003	Data Analysis and Visualization	1	0	0	1	50	-	50	-
6	PCC	CDP6003	Data Analysis and Visualization Lab	0	0	2	1	50	-	50	-
7	PEC	CDT6004	Program Elective - II	3	0	0	3	50	50	100	3
8	PEC	CDP6004	Program Elective - II Lab	0	0	2	1	50	-	50	-
9	PEC	CDT6005	Program Elective - III	3	0	0	3	50	50	100	3
10	MDM	CDT6006	Applied Econometrics	2	0	0	2	50	50	100	2
11	VSEC	CDP6007	Mini Project	0	0	4	2	25	25	50	-
	TOTAL			**15**	**0**	**12**	**22**	**525**	**275**	**800**	

Exit option : Award of UG Degree in Major with 131 credits and an additional 8 credits			
Exit Courses			
1	**Certified Database Engineer (Oracle, DB2)**	**Online/offline certification Course**	**8**
2	**Certified Cloud Engineer (AWS, AZURE)**		**8**
3	**Certified Data Science Engineer**		**8**

Semester-VII

Sr. No.	Category	Course code	Course Name	Hours/ week			Credits	Maximum marks			ESE Duration (Hrs)
				L	T	P		Continuous Evaluation	End Sem Exam	Total	
1	PCC	CDT7001	Deep Learning - II	3	0	0	3	50	50	100	3
2	PCC	CDP7001	Deep Learning - II Lab	0	0	2	1	50	-	50	-
3	PCC	CDT7002	Cloud Computing	3	0	0	3	50	50	100	3
4	PCC	CDP7002	Cloud Computing Lab	0	0	2	1	50	-	50	-
5	PEC	CDT7003	Program Elective - IV	3	0	0	3	50	50	100	3
6	PEC	CDP7003	Program Elective - IV Lab	0	0	2	1	50	-	50	-
7	MDM	CDT7004	Customer Management	2	0	0	2	50	50	100	2
8	PRJ	CDP7005	Major Project - I	0	0	8	4	50	50	100	-
			TOTAL	**10**	**0**	**16**	**18**	**400**	**250**	**650**	
OR											
1.	INTR	CDP7006	Industry Internship	0	0	24	12	100	100	200	-
2.	PCC	CDT7006	NPTEL-1/MOOC-1	-	-	-	03	50	50	100	-
3.	PCC	CDT7007	NPTEL-2/MOOC-2	-	-	-	03	50	50	100	-
			TOTAL	**0**	**0**	**24**	**18**	**200**	**200**	**400**	

Semester- VIII

Sr. No.	Category	Course code	Course Name	Hours/ week			Credits	Maximum marks			ESE Duration (Hrs)
				L	T	P		Continuous Evaluation	End Sem Exam	Total	
1.	PEC	CDT8001	Program Elective-V	3	0	0	3	50	50	100	3
2.	PEC	CDT8002	Program Elective-VI	3	0	0	3	50	50	100	3
3.	Project	CDP8003	Major Project- II	0	0	12	6	50	50	100	-
			TOTAL	**6**	**0**	**12**	**12**	**150**	**150**	**300**	
OR											
1.	INTR	CDP8004	Industry Internship /TBI Internship / Research Internship	0	0	24	12	100	100	200	-
			TOTAL	**0**	**0**	**24**	**12**	100	100	**200**	

List of Program Electives

Elective-I	Elective-II	Elective-III	Elective-IV	Elective-V	Elective-VI
CDT5004-1 Compiler Design	CDT6004-1 Natural Language Processing	CDT6005-1 Data Science for Health Care	CDT7003-1 Image Analysis and Computer Vision	CDT8001-3 Mining Massive Datasets	CDT8002-1 Human Computer Interaction
CDT5004-2 Design Patterns	CDT6004-2 Blockchain Technology	CDT6005-2 Data Science for Genomics	CDT7003-2 Spatial Data Management	CDT8001-2 Generative AI	CDT8002-2 Optimization for Data Science
CDT5004-3 Software Engineering	CDT6004-3 Distributed and Parallel Computing	CDT6005-3 Data Science for Marketing	CDT7003-3 Information Retrieval	CDT8001-3 Information Security and Data Privacy	CDT8002-3 Reinforcement Learning
		CDT6005-4 Artificial Intelligence on Cloud			

SHRI RAMDEOBABA COLLEGE OF ENGINEERING AND MANAGEMENT, NAGPUR – 440013

B.Tech in Computer Science and Engineering (Data Science)

Scheme: Honor 2023-24 NEP

Scheme of Honors in Data Science

Sr. No.	Sem	Course Code	Name of Course	Hours per week		Credits	Maximum Marks			ESE Duration Hrs
				L	P		Continuous Evaluation	End Sem Exam	Total	
1	III	CDTH 3100	Data Science Essentials	3	0	3	100	-	100	-
2	IV	CDTH 4100	Software Architecture Analysis	3	0	3	100	-	100	-
3	V	CDTH 5100	Data Engineering	4	0	4	100	-	100	-
4	VI	CDTH 6100	Business and Web Analytics	4	0	4	100	-	100	-
5	VII	CDTH 7100	Project	0	8	4	50	50	100	-

SHRI RAMDEOBABA COLLEGE OF ENGINEERING AND MANAGEMENT, NAGPUR – 440013

B.Tech in Computer Science and Engineering (Data Science)

Scheme: Minor 2023-24 NEP

Scheme of B. Tech. Minor Specialization in Computer Science Engineering (Data Science)

Sr. No.	Sem	Course Code	Name of Course	Hours per week			Credits	Maximum Marks			ESE Duration Hrs
				L	T	P		Continuous Evaluation	End Sem Exam	Total	
1	III	CDTM 3100	Foundation of Data Science	3	0	0	3	50	50	100	3
2	IV	CDTM 4100	Statistics for Data Science	3	0	0	3	50	50	100	3
3	V	CDTM 5100	Exploratory Data Analysis	4	0	0	4	50	50	100	3
4	VI	CDTM 6100	AI tools for Data Science	4	0	0	4	50	50	100	3
5	VII	CDPM 7100	Major Project	0	0	8	4	50	50	100	-
			TOTAL	**18**	**0**	**0**	**18**	**250**	**250**	**500**	

Syllabus for Semester I, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT1001 Course: Digital Electronics

L: 3 Hrs, T: 0 Hr, P: 0Hr, Per Week Total Credits: 3

Course Objectives

The objective of this course is to familiarize the prospective engineers with:

1. Logic functions using Boolean algebraic theorems and techniques
2. Conventional combinational and sequential circuits including conversions of flip-flops.
3. The exploration of the semiconductor memories and programmable logic devices.
4. The basic concept of microprocessor with addressing mode and instruction set for programming.

SYLLABUS

UNIT-I

Basics of Digital Electronics

Motivation for digital systems: Logic and Boolean algebra, Number Systems. Logic Gates & Truth Tables, Demorgan's law, Minimization of combinational circuits using Karnaugh maps up to five variable. Map manipulation-essential prime implicants, non-essential prime implicants**.**

UNIT-II

Combinational Circuit Design

Design procedure: Multiplexers, Demultiplexer, Encoders, Decoders, Code Converters, Adders, Subtractor (Half, Full), BCD Adder/ Subtractor, ripple and carry look-ahead addition booth's Algorithm, bit-pair recoding, Integer Division- restoring and non-restoring division

UNIT-III

Sequential circuit Design-I

Storage elements, Flip-flops and latches: D, T, J/K, S/R flip-flops. Master Slave Conversion of one of type of F/F to another Sequential circuit. Analysis –Input equations, state table, and analysis with J-K Flip flops. Sequential circuit Design, Design procedure, designing with D & J-K Flip flop.

UNIT-IV

Sequential circuit Design-II

Counters, asynchronous and synchronous design using state and excitation tables. Registers & Shift registers., Mealey & Moore Machines

UNIT-V

Memory & Programmable logic Devices

Semiconductor RAM memories, Static and Dynamic Memories, ROM, higher order memory design, multi-module memories, Memory interleaving, , Secondary storage – Magnetic disk, Optical disk, PLA, PAL.

UNIT-VI

Fundamental of Microprocessor

Introduction to µp 8085, Addressing modes, Instruction set, Programming of µp 8085.

Course Outcomes

After successful completion of this course, the student will be able to,

1. Outline binary arithmetic operations and optimize Boolean functions using Karnaugh map (k-map) method.
2. Apply combinational circuits for realization of basic building blocks of conventional digital circuits.
3. Design sequential blocks like flip flops, counters, registers, simple finite state machine and similar circuits.
4. Describe the memory elements and combinational digital circuits implementation with programmable logic devices.
5. Use addressing modes and instruction set of target microprocessors for writing efficient assembly language programs.

Text Books

1. Morris Mano; Digital Logic Design; Fourth edition, McGraw Hill
2. R.P.Jain; Modern Digital Electronic; Fourth edition; Tata McGraw-Hill.
3. V.J.Vibhute; 8-Bit Microprocessor & Microcontrollers; fifth edition.

Reference books

1. A. Anand Kumar; Fundamental of Digital Electronics; Second Edition, PHI
2. A.P.Godse; Digital circuit & design; Technical Publications; 2009.
3. Ramesh Gaonkar; 8 bit Microprocessor; CBS Publishers; 2011.

Syllabus for Semester I, B. Tech. Computer Science & Engineering
Course Code: CDP1001 Course: Digital Electronics Lab

L: 0 Hrs, T: 0 Hr, P: 2Hr Per Week Total Credits: 1

Course Outcome

On Successful completion of course, students will be able to:

1. Use logic gates for designing digital circuits
2. Implement combinational circuits using VHDL
3. Implement sequential circuits using VHDL
4. Apply the knowledge gained for their project work based on the hardware digital circuits

Practical based on above theory syllabus

Syllabus for Semester I, B. Tech. Computer Science & Engineering

Course Code: CDT1002 **Course: Programming for Problem Solving**

L: 3 Hrs, T: 0 Hr, P: 0Hr, Per Week **Total Credits: 3**

Course Outcomes

On successful completion of course student will learn:

1.Develop C programs from the algorithm for simple arithmetic and logical problems.

2.Design programs using conditional branching, iteration and recursion.

3.Formulate algorithm/programs using arrays, pointers, structures and I/O operations.

4.Design programs using matrix manipulation along searching & sorting problems.

UNIT-I: Introduction to Programming

Introduction to components of a computer system (disks, memory, processor, where a program is stored and executed, operating system, compilers etc.) Idea of Algorithm: Steps to solve logical and numerical problems. Representation of Algorithm: Flowchart /Pseudo code with examples. Arithmetic expressions and precedence

UNIT-II: C Programming Language

Introduction to C language: Keywords, Constant, Variable, Data types, Operators, Types of Statements, Pre-processor Directives, Decision Control Statement-if, if-else, nested if-else statement, switch case, Loops and Writing and evaluation of conditionals and consequent branching.

UNIT-III: Arrays and Basic Algorithms

Arrays: 1-D, 2-D, Character arrays and Strings. Searching, Basic Sorting Algorithms (Bubble, Insertion and Selection), Finding roots of equations, notion of order of complexity through example programs (no formal definition required)

UNIT-IV: Functions and Recursion

User defined and Library Functions, Parameter passing in functions, call by value, passing arrays to functions: idea of call by reference. Recursion: As a different way of solving problems.

Example programs, such as Finding Factorial, Fibonacci series, Ackerman function etc. Quick sort or Merge sort.

UNIT-V: Pointers and Structures

Structures, defining structures, Array of Structures, Introduction to pointers, Defining pointers, Pointer arithmetic, pointer operators, Use of Pointers in self-referential structures, notion of linked list (no implementation)

UNIT-VI: File handling

Streams in C, Types of Files, File Input/ Output Operations: Modes of file opening, Reading and writing the file, Closing the files, using fflush ().

Text Books:

1. Programming in ANSI C: E. Balguruswami McGraw Hill
2. Mastering C: K. R. Venugopal and S. R. Prasad, Tata McGraw Hill

Reference Books

1. Programming with C: Byron Gottfried, Schaums Outline Series.
2. Let Us C: Yashwant Kanetkar, BPB Publication

Syllabus for Semester I, B. Tech. Computer Science & Engineering

Course Code: CDP1002 **Course: Programming for Problem Solving Lab**

L: 0 Hrs, T: 0 Hr, P: 2Hr, Per Week **Total Credits: 1**

Course Outcomes

On successful completion of course student will be able to:

1. Create C programs using loops and decision-making statements to solve and execute the given problem.
2. Develop programs and functions one dimensional and two-dimensional arrays.
3. Apply the concept of pointers, structures to develop programs.
4. Implement files in C to store the data for the given problem.

Practical's based on above theory syllabus

Syllabus for Semester I, B. Tech. Computer Science & Engineering

Course Code: CDT1003 **Course: Computer Workshop-1**

L: 1 Hrs, T: 0 Hr, P: 0 Hr, Per Week **Total Credits: 1**

Course Objectives

1. Understand the definition and principles of UI/UX in order to design with intention.

2. Achieve an understanding of the life-cycle of application design—the process, purpose, and tools.

3. Learn the basics of HCI (human-computer interaction) and the psychology behind user decision-making.

4. Explore UI/UX tools to interpret requirements of modern applications.

5. Elaborate design decisions through presentations of assignments.

Unit 1:

UI/UX Overview: Introduction to UI/UX, Principles of UI/UX, UI Components, Design Thinking, Interaction Design, Usability.

Unit 2:

UI Programming: Basic of HTML5, Elements of HTML5, Background of CSS, Bootstrap CSS, Fundamentals of JavaScript, HTML DOM Manipulations.

Unit 3:

UX Programming: Figma Basics, How to identify user needs, Wireframe and Prototype, Digital Storytelling.

Course Outcomes

On successful completion of the course, students will be able to:

1. Understand basics of UI/UX

2. Design and develop web pages using HTML, CSS and JavaScript

3. Infer the significance of Wireframing and build prototypes.

Text Books

1. UI/UX design for designer and developers: by Nathan Clark

2. Web Design: A Beginner's Guide Second Edition by Wendy Willard

3. User story mapping by Jeff Patton, O'Reilly Publication

Syllabus for Semester I, B. Tech. Computer Science & Engineering

Course Code : CDP1003 **Course : Computer Workshop-1 Lab**

L: 0 Hrs, T: 0 Hr, P: 2 Hr, Per Week **Total Credits: 1**

Course Objectives

1. Understand the definition and principles of UI/UX in order to design with intention.

2. Achieve an understanding of the life-cycle of application design—the process, purpose, and tools.

3. Learn the basics of HCI (human-computer interaction) and the psychology behind user decision-making.

4. Explore UI/UX tools to interpret requirements of modern applications.

5. Elaborate design decisions through presentations of assignments.

Syllabus:

Practical based on Theory Syllabus

Course Outcomes

On successful completion of the course, students will be able to:

1. Design and develop static web pages using HTML and CSS
2. Develop dynamic web pages using JavaScript
3. Create high-fidelity designs and prototypes in Figma

Syllabus for Semester II, B. Tech. Computer Science & Engineering

Course Code: CDT2001	**Course: Object Oriented Programming**
L: 3 Hrs, T: 0 Hr, P: 0Hr, Per Week	**Total Credits: 3**

Course Objectives

1. To make students understand Fundamental features of an object-oriented language like Java: object classes and interfaces, exceptions and libraries of object collections
2. Introduce students with fundamental concepts like exception handling, generics, collection classes and streams.

Syllabus

Unit I:

Features of Object-Oriented Programming languages, Abstraction, Encapsulation, Inheritance, polymorphism and late binding. Programming paradigms, Bytecode, JDK, JRE, JVM.

Concept of a class and object, ways of representing objects, access control of members of a class, instantiating a class, constructor.

Unit II:

Concept of overloading: Constructor Overloading, Function Overloading.

Arrays and Array of objects, Wrapper classes (Integer, Double etc.), String Class, creating packages, importing packages.

Lambda Expressions Introduction, Block, Passing Lambda expression as Argument

Unit III:

Concept of inheritance, methods of derivation, use of super keyword and final keyword in inheritance, run time polymorphism, abstract classes and methods, Interface, implementation of interface, static and non-static members.

Unit IV:

Exceptions, types of exception, use of try catch block, handling multiple exceptions, using finally, throw and throws clause, user defined exceptions, Introduction to streams, byte streams, character streams, file handling in Java, Serialization.

Unit V:

Generics, generic class with two type parameter, bounded generics.

Collection classes: Arraylist, LinkedList, TreeSet, HashMap, Iterator, ListIterator, Comparator, Comparable

Unit VI:

Introduction to Design Patterns, Need of Design Pattern, Classification of Design Patterns, Role of Design Pattern in Software design, Creational Patterns, Structural Design Patterns and Behavioral Patterns.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Understand the object-oriented programming features, classes, objects and methods.
2. Develop efficient programs by implementing the concept of Inheritance, polymorphism exception handling.
3. Use the concept of generics, collections, streams to develop solution to the given problem.
4. Analyze characteristics and need of design pattern in software design process.

Text Books

1. Herbert Schildt; JAVA The Complete Reference; Ninth Edition, Tata McGraw- Hill Publishing Company Limited.
2. Design Patterns By Erich Gamma, Pearson Education

Reference Books

1. Paul Deitel, Harvey Deitel; Java 9 for Programmers; Pearson
2. Herbert Schildt and Dale Skrien; Java Fundamentals A Comprehensive Introduction; Tata McGraw- Hill Education Private Ltd 2013.

Syllabus for Semester II, B. Tech. Computer Science & Engineering

Course Code : CDP2001 **Course : Object Oriented Programming Lab**

L: 0Hrs, T: 0 Hr, P: 2Hr, Per Week **Total Credits : 1**

Course Objectives

1. To develop ability of students to implement basic concepts and techniques of object oriented programming paradigm like encapsulation, inheritance, polymorphism, exception handling.
2. Develop solution to problems using collection classes, generics, streams, multithreading.

Course Outcomes

On completion of the course the student will be able to

1. Design solution to problems using concepts of object oriented programming like classes, objects, inheritance with proper exception handling.
2. Use collection classes, generic classes to design programs and perform database connectivity.
3. Implement programs based on streams and multithreading.

SYLLABUS

Experiments based on above Syllabus.

Syllabus for Semester II, B. Tech. Computer Science & Engineering

Course Code: CDT2002 **Course: Computer Architecture**

L: 2 Hrs, T: 0 Hr, P: 0Hr, Per Week **Total Credits: 2**

Course Objectives

The objective of this course is to familiarize the prospective engineers with:

1. Concepts of computer architecture by developing understanding of various functional units, components of computers and working of all the modules.
2. Design principles of modern computers including memory, bus system, input/output operation, interrupt handling mechanism and parallelization.

SYLLABUS

UNIT I : Basic Structure of Computers: Functional units of computer, basic operational concepts- Instruction, processor and memory, operating steps, address, Big- and Little-endian assignments, Instructions set architecture of a CPU- Instruction Formats, Instruction sequencing, addressing modes, and instruction set classification, subroutine & parameter passing, expanding opcode, RISC and CISC.

UNIT II : Basic Processing Unit and Data Representation: Basic Concepts- Instruction execution, Bus architecture- One bus and Multi-bus, Execution of a Complete Instruction, sequencing of control signals, Hardwired control, Micro-programmed Control. Floating point numbers-representation, guard bits and rounding.

UNIT III : Memory & Input/output: Cache memory, Cache size vs. block size, mapping functions, replacement algorithms, Cache read/write policy, Virtual Memory, I/O mapped I/O and memories mapped I/O, interrupt and interrupt handling mechanisms, vectored interrupts, synchronous vs. asynchronous data transfer, Bus Arbitration, Direct Memory Access

UNIT IV : Pipelining: Basic concepts of pipelining, throughput and speedup, Introduction of Parallel Computing: SISD, MISD, SIMD, MIMD

Course Outcomes:

On Successful completion of course, students will be able to:

1. Demonstrate the understanding about the functional units of a digital computer system.
2. Execute complete instruction on different types of bus architectures with control signal generation.
3. Analyse memory, multiprocessor and multicore architectures and their implications in parallel computing.

Text Books

1. V.C.Hamacher, Z.G.Vranesic and S.G.Zaky; Computer Organisation; 5th edition; Tata McGraw Hill, 2002.

2. W. Stallings; Computer Organization & Architecture; PHI publication; 2001.
3. J. P. Hayes; Computer Architecture & Organization; 3rd edition; McGraw-Hill; 1998.

Reference Books

1. M Mano; Computer System and Architecture; PHI publication; 1993.
2. A.S.Tanenbaum; Structured Computer Organization; Prentice Hall of India Ltd.

Syllabus for Semester II, B. TECH Computer Science & Engineering

Course Code : CDT2003 **Course : Computer Workshop-2**

L: 1Hrs, T: 0 Hr, P: 0 Hr, Per Week **Total Credits: 1**

Course Objective

The objective of this course is to familiarize the students with an important web framework for developing user interfaces. It aims for developing high end web applications by the use of ReactJS features.

Course Outcomes

After successful completion of this course, the student will be able to,

1. Implement the fundamentals of React with Java Script and JSX
2. Understand Templating concept along with different types of components in ReactJS
3. Understand different state and Life Cycle Methods
4. Implement Router with react Router.

Course Contents

UNIT-I

Introduction to React

React JS Introduction, Advantages of React JS, Introduction to JSX, Difference between JS and JSX.

UNIT-II

Components in React

React Components overview, Types of components, Controlled, Split Up, Composable, Reusable, Component Declarations and Styling Components

State and it significance, Read state and set state, Passing data to component using props, Validating props using prop Types, Supplying default values to props using default Props

UNIT-III

Routing with react router

Introduction to React Router, Routing in single page applications, Browser Router and Hash Router components Configuring route with Route component.

Text Books

- Pure React- a step by step guide - Dave Ceddia
- Road to learn react - Robin Wieruch
- React in Action 1st Edition - Mark Tielens Thomas

Syllabus for Semester II, B. TECH Computer Science & Engineering

Course Code : CDP2003 **Course : Computer Workshop-2**

L: 0 Hrs, T: 0 Hr, P: 2 Hr, Per Week **Total Credits: 1**

Course Objective

The objective of this course is to familiarize the students with an important web framework for developing user interfaces. It aims for developing high end web applications by the use of ReactJS features.

Syllabus:

Practical based on Theory Syllabus

Course Outcomes

After successful completion of this course, the student will be able to

1. Understanding the fundamentals of ReactJS including components, props, state, and lifecycle methods.
2. Design and implement complex applications by composing smaller, reusable components together.
3. Building Web Applications to create dynamic and interactive web applications using React and other related technologies like JSX and ES6.
4. Implement React Router to handle client-side routing and create single-page applications.

Syllabus for Semester III, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT3001 **Category:** Program Core Course (PCC)

Course: Data Structures and Algorithm

L: 3Hr**, T:** 1Hr**, P:** 0Hrs**, Per Week, Credits:** 4

Course Objectives

The objective of this course is to familiarize the prospective engineers with:

1. To impart to students the basic concepts of data structures and algorithms.

2. To familiarize students on different searching and sorting techniques.

To prepare students to use linear (stacks, queues, linked lists) and non-linear (trees, graphs) data structures.

3. To enable students to devise algorithms for solving real-world problems.

SYLLABUS

UNIT I Data Structures and Algorithms Basics

Introduction: basic terminologies, elementary data organizations, data structure operations; abstract data types (ADT) and their characteristics. Algorithms: definition, characteristics, analysis of an algorithm, asymptotic notations, time and space tradeoffs. Array ADT: definition, operations and representations – row-major and column- major.

UNIT II Stacks and Queues

Stack ADT: allowable operations, algorithms and their complexity analysis, applications of stacks –expression conversion and evaluation (algorithmic analysis), multiple stacks. Queue ADT: allowable operations, algorithms and their complexity analysis for simple queue and circular queue, introduction to double-ended queues and priority queues.

UNIT III Linked Lists

Singly Linked Lists: representation in memory, algorithms of several operations: traversing, searching, insertion, deletion, reversal, ordering, etc.Doubly and Circular Linked Lists: operations and algorithmic analysis. Linked representation of stacks and queues, header node linked lists.

UNIT IV Sorting and Searching

Sorting: different approaches to sorting, properties of different sorting algorithms (insertion, Shell, quick, merge, heap, counting), performance analysis and comparison.

Searching: necessity of a robust search mechanism, searching linear lists (linear search, binary search)and complexity analysis of search methods.

UNIT V Trees

Trees: basic tree terminologies, binary tree and operations, binary search tree [BST] and operations with time analysis of algorithms, threaded binary trees. Self-balancing Search Trees: tree rotations, AVL tree and operations, B+-tree: definitions, characteristics, and operations (introductory).

UNIT VI Graphs and Hashing

Graphs: basic terminologies, representation of graphs, traversals (DFS, BFS) with complexity analysis, path finding (Dijkstra's SSSP, Floyd's APSP), and spanning tree (Prim's method) algorithms.

Hashing: hash functions and hash tables, closed and open hashing, randomization methods (division method, mid-square method, folding), collision resolution techniques.

Course Outcomes

After successful completion of this course, the student will be able to,

1. Recognize different ADTs and their operations and specify their complexities.
2. Design and realize linear data structures (stacks, queues, linked lists) and analyze their computation complexity.
3. Devise different sorting (comparison based, divide-and-conquer, distributive, and tree-based) and searching (linear, binary) methods and analyze their time and space requirements.
4. Design traversal and path finding algorithms for Trees and Graphs.

Text Books

1. Ellis Horowitz, Sartaj Sahni& Susan Anderson-Freed, Fundamentals of Data Structures in C,Second Edition, Universities Press, 2008.
2. Robert L. Kruse and Bruce P. Leung, Data Structures and Program Design in C, Pearson Education;
3. Mark Allen Weiss; Data Structures and Algorithm Analysis in C; Second Edition; Pearson Education; 2002.

4. G.A.V. Pai; Data Structures and Algorithms: Concepts, Techniques and Application; FirstEdition; McGraw Hill; 2008.

Reference Books

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein; Introduction to Algorithms; Third Edition; PHI Learning; 2009.
2. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran; Fundamentals of Computer Algorithms; Second Edition; Universities Press; 2008.
3. A. K. Sharma; Data Structures using C, Second Edition, Pearson Education, 2013.

Syllabus for Semester III, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDP3001 **Category:** Program Core Course (PCC)

Course: Data Structures and Algorithm Lab

L: 3Hr**, T:** 0Hr**, P:** 0Hrs**, Per Week, Credits:** 3

Course Outcome

On Successful completion of course, students will be able to:

1. To enable students to employ different searching and sorting methods.
2. To prepare students to identify and apply linear (stacks, queues, linked lists) and non-Linear(trees, graphs) data structures in solving problems.
3. To encourage students to design and execute tree-based algorithms for solving real-world problems.

SYLLABUS

Experiments based on CDT3001 Syllabus in C / C++ / Java / Python

Course Outcomes

On completion of the course the student will be able to

1. Design and realize different linear data structures.
2. Identify and apply specific methods of searching and sorting to solve a problem.
3. Implement and analyze operations on binary search trees and AVL trees.
4. Implement graph traversal algorithms, find shortest paths and analyze them.

Reference Books

1. K R. Venugopal and Sudeep. R Prasad; Mastering C; Second Edition; McGraw Hill; 2015.
2. Ellis Horowitz, SartajSahni& Susan Anderson-Freed, Fundamentals of Data Structures in C,Second Edition, Universities Press, 2008.
3. Mark Allen Weiss; Data Structures and Algorithm Analysis in C; Second Edition; PearsonEducation; 2002.

Syllabus for Semester III, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT3002 **Category:** Program Core Course (PCC)

Course: Theory of Computations

L: 3Hrs**, T:** 0Hr**, P:** 0Hr**, Per Week, Credits:** 3

Course Objectives

The objective of this course is to familiarize the prospective engineers with:

1. To provide students an understanding of basic concepts in the theory of computation.
2. To teach formal languages and various models of computation.
3. To exhibit fundamental concepts related with computability theory.

SYLLABUS

UNIT-I Basics of Theory of Computation

Basics of Sets and Relation, Countability and Diagonalization, Principle of mathematical induction, Pigeon-hole principle. Fundamentals of formal languages and grammars, Chomsky hierarchy of languages.

UNIT-II Finite Automata

Deterministic finite automata (DFA), Nondeterministic finite automata (NFA) and equivalence with DFA, Minimization of finite automata, NFA with Epsilon Transitions, Finite Automata with output.

UNIT-III Regular Grammar

Regular expressions and Regular languages, Regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages

UNIT-IV Context-Free Grammar

Context-free grammars (CFG) and language (CFL), parse trees, ambiguity in CFG, Reduction of CFGs, Chomsky and Greibach normal forms.

UNIT-V Push Down Automata

Deterministic pushdown automata and non-deterministic push down automata, Acceptance by two methods: Empty stack and Final State, Equivalence of PDA with CFG, closure properties of CFLs.

UNIT-VI Turing Machines

The basic model for Turing machines (TM), Turing recognizable recursively enumerable) and Turing-decidable (recursive) languages, variants of Turing machines, unrestricted grammars and equivalence with Turing machines, TMs as enumerators.

Course Outcomes

On successful completion of the course, students will be able to demonstrate

1. Describe the formal relationships among machines, languages and grammars.
2. Design and Optimize finite automata for given regular languages.
3. Design context free grammar for a given language.
4. Design Pushdown Automata, Turing Machine for given languages.

Text Books

1. John Martin, Introduction to Languages and The Theory of Computation, Tata McGraw Hill

Reference Books

1. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education Asia.
2. Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation, Pearson Education Asia.
3. Dexter C. Kozen, Automata and Computability, Undergraduate Texts in Computer Science, Springer.
4. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing.

Syllabus for Semester III, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDP3003 **Category:** Program Core Course (PCC)

Course: Programming Language Lab

L: 0Hrs**, T:** 0Hr**, P:** 4Hr**, Per Week, Credits:** 2

Course Objectives:

1. The course focuses on developing the python programming skills to do a variety of programming tasks where the students are encouraged to develop application using python.
2. To cover the basic constructs of python programming, data structures and object-oriented concepts.
3. The course also targets the coverage of important modules and libraries available in python.

Course Outcome

On successful completion of the course, students will be able to:

1. Design Python Programs using Different Data and Control Structures.
2. Use Functions, Python Files, Modules and Packages to handle complex python programs.
3. Develop Mathematical Models and Scientific Applications in Python using various Libraries.
4. Write Small Python Applications using Web Scrapping and PyGame Modules.

Syllabus

- Introduction to Python: Basic Building Blocks of Python, Installation and Working with Python, Keywords, Variables and Operators.
- Data Types: int , float, complex, User Input, Arithmetic Expressions ,Using Strings and Operations on Strings, List, Slicing List and Strings, Split, Tuples, Dictionary, Sets.
- Flow Control: Conditional blocks: if, else, elseif, for Loops in Python: Loops with range, Strings, List and Dictionaries, While Loop, break, continue and pass statements.
- Python Functions: Library Functions, User Defined Functions, Function Argument Types, Recursion, Returning Multiple Values, Lambda, Map, Filter, Reduce.

- Python OOPS Basics: Classes, Object, Class Variable and Instance Variable.
- Files: Reading and Writing Files in Python, File Operations and Modes.Modules and Packages: To Create and Import Module.
- Open-Source Python library- Pandas, Sci Py, NumPy, Matplotlib and Seaborn.
- Web scrapping: Web scrapping with the help of standard libraries like Requests and Beautiful Soup.
- Python Pygame (Game Development Library)

Practical based on above syllabus

Text Books

1. Python Programming Using Problem Solving Approach: Reema Thareja, OxfordUniversity, Press; First edition.
2. Learning Python: Powerful object-oriented programming, Mark Lutz, O'REILLY publications 5th addition.
3. Introduction to Computing & Problem Solving with Python Jeeva Jose and P SojanLal Ascher.
4. Problem Solving with Algorithms and Data Structures using Python by Brad Miller and David Ranum, 2nd addition.

Reference Books

1. Allen Downey, Jeffrey Elkner, Chris Meyers, Learning with Python, Dreamtech Press
2. David M. Baezly “Python Cookbook” O'Reilly Media; Third edition, 201

Syllabus for Semester III, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDP3004 **Category: (VSEC)**

Course: Software Laboratory - I

L: 0Hr, T: 0Hr, P: 2Hrs, Per Week, Credits: 1

Course Objectives:

1. Familiarize students with the use and applications of various statistical tools and packages in data science.
2. Provide hands-on experience in statistical programming and data analysis.
3. Equip students with the skills to perform exploratory data analysis, statistical modeling, and inference.

Syllabus can include practical based on following:

1. **Data Exploration and Visualization**

Load a dataset containing information about housing prices in a city. Explore the dataset, calculate summary statistics, and visualize the distribution of housing prices using appropriate plots.

2. **Descriptive Statistics**

Given a dataset of student grades, calculate measures of central tendency (mean, median, mode) and measures of variability (variance, standard deviation). Interpret the results and discuss the implications of different measures.

3. **Probability Distributions**

Simulate a binomial distribution with given parameters (number of trials and probability of success). Plot the distribution and calculate the probability of different events occurring.

4. **Sampling Distributions and Confidence Intervals**

Generate a random sample from a population with known mean and standard deviation. Construct confidence intervals for the sample mean and interpret the results.

5. **Hypothesis Testing**

Analyze a dataset containing the weights of two different groups of individuals. Conduct a two-sample t-test to determine if there is a significant difference in mean weights between the two groups.

6. **Chi-square Test**

Given a dataset of categorical data, perform a chi-square test to determine if there is a significant association between two categorical variables.

7. **Analysis of Variance (ANOVA)**

Analyze a dataset containing crop yields from different fertilizer treatments applied to multiple plots of land. Conduct a one-way ANOVA to determine if there is a significant difference in mean crop yields among the different fertilizer treatments.

8. **Simple Linear Regression & Multiple Linear Regression**

Using a dataset of housing prices and square footage, fit a simple linear regression model to predict housing prices based on square footage. Evaluate the model's performance and interpret the regression coefficients.

Analyze a dataset containing information about cars, including factors such as engine size, fuel efficiency, and price. Fit a multiple linear regression model to predict the price of a car based on various features.

9. **Non-linear Regression**

Given a dataset of population growth over time, fit a non-linear regression model (e.g., exponential or logistic) to model the population growth pattern. Evaluate the model's fit and interpret the results.

10. **Logistic Regression**

Using a dataset containing customer data and purchase history, build a logistic regression model to predict whether a customer will make a purchase based on various factors (e.g., age, income, past purchases).

These problem statements cover a range of scenarios and applications, allowing students to practice data exploration, visualization, descriptive statistics, probability distributions, inferential statistics, regression analysis, and their implementation using statistical programming tools. Additionally, the case study problem statement encourages students to apply their skills to a real-world dataset and communicate their findings effectively.

Course Outcomes:

Upon completion of the course, students will be able to:

1. Understand and apply different statistical techniques on given data.
2. Perform and interpret various probability distributions.
3. Carry out hypothesis testing and calculate confidence intervals.
4. Create and evaluate regression models.
5. Implement statistical analysis using programming tools and packages.
6. Analyze and visualize data using statistical programming techniques.
7. Apply statistical methods and programming skills to real-world data science problems.

Reference Books:

1. An Introduction to Statistics with Python: With Applications in the Life Science by Thomas Haslwanter, Springer.
2. Practical Statistics for Data Scientists by Peter Bruce, Andrew Bruce, Peter Gedeck, Reilly Media.

Syllabus for Semester III, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDP3006 **Category:** (HSSM)

Course: Idea Lab

L: 0Hr**, T:** 0Hr**, P:** 4Hrs**, Per Week, Credits:** 2

Course Objectives

This course provides students with a unique opportunity to engage, explore, experience, express and excel in innovative thinking. The course will accelerate the development of indigenous solutions by inculcating creative skills, problem-solving skills and entrepreneurship skills in students. The ultimate goal is to help students build his/her creative acumen to address real life challenges.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Identify a real-world challenge or potential opportunities that demand consideration.
2. Design and propose original ideas or innovative solutions to identified real-world challenge.
3. Build a plan to develop proof of concept (POC)/ prototype/minimum viable product (MVP) for the idea.
4. Critically evaluate the feasibility and viability of proposed idea, considering its impact on various stakeholders.

Execution Plan for the Subject:

- Conduction of lectures/guest lectures to familiarize the students with concepts of principles of design thinking, innovative and business-oriented solution building.
- Students will identify the potential areas or real-world problems that require innovative solution.
- Students will contribute to the creation and presentation of their solution. The developed solution can be a prototype or an idea or a business plan or a software solution, etc.
- Students will analyze and provide constructive feedback on the ideas and solutions presented by self and peers.

Syllabus for Semester III, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT3007 **Category:** Value Education Course (VEC)

Course: Cyber Laws and Ethics in IT

L: 2Hrs**, T:** 0Hr**, P:** 0Hr**, Per Week, Credits:** 2

Course Objectives:

1. To explore ethical principles in the business world and IT organization
2. To understand various intellectual property issues such as copyrights, patents, Trade secrets, and trademarks.
3. To identify emerging plagiarism policies, anonymity issues, identity theft, and consumer profiling.
4. To interpret Indian Information Technology Act 2000 and explore cyber security issues in the society and business world.

Syllabus:

UNIT I

Ethics in business world & IT professional malpractices, Introduction to firewalls, IDS System, Distortion and fabrication of information

UNIT II

Ethics of IT Organization: Contingent Workers H- IB Workers, Whistle- blowing, Protection for Whistle- Blowers, Handling Whistle- blowing situation, Digital divide.

UNIT III

Intellectual Property: Copyrights, Patents, Trade Secret Laws, Key Intellectual property issues, Plagiarism, Privacy: The right of Privacy, Protection, Key Privacy and K-Anonymity issues, Identity Theft, Consumer Profiling,

UNIT IV

Cyber laws and rights in today's digital age, Emergence of Cyberspace, Cyber Jurisprudence, Cyber Crimes against Individuals, Institution and State, Hacking, Digital Forgery, Cyber Stalking/Harassment, Cyber terrorism, cyber tort, Cyber Defamation & hate speech,Competitive Intelligence, Cybersquatting, The Indian Information Technology Act 2000 IT Act.

Course Outcomes

On successful completion, of course student will able to learn:

1. To analyze the role of ethics in IT organization.
2. To identify various cyber laws with respect to legal dilemmas in the Information Technology field.
3. To interpret various intellectual property rights, Privacy, Protection issues in Information Technology field.
4. To describe the ways of precaution and prevention of Cyber Crime as well as Human Rights.

Text Books:

1. George Reynolds, “Ethics in information Technology”,5th edition, Cengage Learning
2. Hon C Graff, Cryptography and E-Commerce - A Wiley Tech Brief, Wiley Computer Publisher, 2001.

Reference Books:

1. Michael Cross, Norris L Johnson, Tony Piltzecker, Security, Shroff Publishers and Distributors Ltd.
2. Debora Johnson,” Computer Ethic s”,3/e Pearson Education.
3. Sara Baase, “A Gift of Fire: Social, Legal and Ethical Issues, for Computing and the Internet,” PHI Public at ions.
4. Chris Reed & John Angel, Computer Law, OUP, New York, (2007).
5. Dr Pramod Kr.Singh, “Laws on Cyber Crimes [Along with IT Act and Relevant Rules]” Book Enclave Jaipur India.

Syllabus for Semester IV, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT4001 **Category:** Program Core Course (PCC)

Course: Operating System **L:** 3Hrs**, T:** 0Hr**, P:** 0Hr**, Per Week, Credits:** 3

Course Objectives

The objective of this course is to familiarize the prospective engineers with:

1. To learn how Operating System is Important for Computer System.
2. To make aware of different types of Operating System and their services.
3. To learn different process scheduling algorithms and synchronization techniques to achieve better performance of a computer system.
4. To know memory organization and management concepts.
5. To learn secondary memory, directory and file management

SYLLABUS

Unit I: Introduction:

Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS - Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine, Case study on different Operating Systems.

Unit II: Processes and CPU Scheduling:

Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching

Threads: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads.

Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non-pre-emptive, FCFS, SRTF, Priority, RR

Unit III: Process Synchronization:

Critical Section, Mutual Exclusion, Peterson's solution, Hardware Solution, Software Solutions, Semaphores, Monitors, Message Passing, Classical IPC Problems: Producer-Consumer Problem, Reader-Writer Problem, Dinning Philosopher Problem etc.

Unit IV: Deadlocks:

Definition, Necessary and sufficient conditions for Deadlock, Starvation, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.

Unit V: Memory Management:

Basic concept, Logical and Physical address mapping, Memory allocation: Contiguous Memory allocation, Fixed and variable partition, Internal and External fragmentation and Compaction,

Paging: Page allocation, Hardware support for paging, Advantages & Disadvantages of paging. Segmentation, Virtual Memory: Basics of Virtual Memory

Demand Paging : Page Fault Mechanism, Page Replacement, Page Replacement algorithms: First in First Out (FIFO), Least Recently used (LRU), and Optimal.

Unit VI: File Management:

Concept of File, Access methods, File types, File operations, Directory structure, File System structure

Disk Management**:** Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK

Course Outcomes

On successful completion of the course, students will be able to:

1. Understands the different services provided by Operating System at different level.
2. Understanding the performance and design trade-offs in complex software systems
3. Understand the role of various components (process, memory, file systems, etc.) of operating system.
4. Understands the use of different process scheduling algorithm and synchronization techniques to avoid deadlock
5. Analyze and apply resource (CPU, Memory, Disk) management policies.

Text Books

1. Operating System Concepts, 8th Edition by A. Silberschatz, P.Galvin, G. Gagne,Wiley India Edition.
2. Modern Operating Systems, 2nd Edition by Andrew Tanenbaum, PHI.

Reference Books

1. Operating Systems: Internals and Design Principles, 5th Edition, William Stallings,Prentice Hall of India.
2. Operating Systems, W. Stalling, Macmillan.
3. Operating Systems, H. M. Dietel, Addison Wesley Longman.
4. Understanding the Linux Kernel, 3rd Edition, Daniel P. Bovet, Marco Cesati, O'Reilly.
5. Linux Kernel Development, Robert Love, Third Edition, Addison-Wesley, 2010.
6. The design of Unix Operating system, Maurice J. Bach, Pearson Education, India.

Syllabus for Semester IV, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDP4001 **Category:** Program Core Course (PCC)

Course: Operating System lab

L: 0Hrs**, T:** 0Hr**, P:** 2Hr**, Per Week, Credits:** 1

SYLLABUS

Practical based on CDT4001 Syllabus

1. Linux System Commands
2. Execution of I/O system calls
3. CPU scheduling algorithms
4. Inter process communication
5. Process synchronization
6. Deadlock Detection
7. Memory Management
8. Disk management

Course Outcomes

On Successful completion of course, students will be able to:

1. Implement system commands by making use of LINUX system calls.
2. Implement processes and process schedulers.
3. Design solutions to process synchronization and deadlock handling.
4. Implement Memory management and File management solutions.

Syllabus for Semester IV, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDP4002 **Category:** Program Core Course (PCC)

Course: Artificial Intelligence **L:** 3Hrs**, T:** 0Hr**, P:** 0Hr**, Per Week, Credits:** 3

Course Objectives

The objective of this course is to familiarize the prospective engineers with:

1. To understand challenges involved in designing intelligent systems.
2. To represent given problem using state space representation and solve it by using different search techniques.
3. To understand knowledge representation methods using logic programming.
4. To understand uncertainty theory in designing AI systems.

SYLLABUS

UNIT I:

Introduction: Basics of problem solving, problem representation (toy problems and real world problems); Structure of agent, rational agent, Specifying task environment, Properties of task environment; measuring problem-solving performance

UNIT II:

Uninformed search techniques: Depth, Breadth, Uniform Cost, Depth Limited, Iterative deepening DFS, Bidirectional Search

UNIT III:

Informed search techniques: Heuristic Based Search, Greedy Best First Search, A* Search; Local Search algorithms: Hill-climbing, Simulated Annealing, Genetic Algorithms.

UNIT IV:

Adversarial Search: Two player Games, The min-max algorithm, Alpha-Beta pruning. Constraint Satisfaction Problems: Constraint propagation, backtracking search

UNIT V:

Propositional Logic: Inference, Equivalence, Validity and satisfiability, Resolution, Forward and Backward Chaining, First Order Logic: Syntax and Semantics of FOL, Inference in FOL, Unification, Forward Chaining, Backward Chaining, and Resolution.

UNIT VI:

Uncertainty Knowledge and Reasoning: Probability and Baye's Theorem, Statistical reasoning: Bayesian networks, Naïve bayes algorithm, Fuzzy Logic, Introduction to expert system

Course Outcomes:

1. Represent given problem using state space representation
2. Apply uninformed and informed search techniques.
3. Solve the fully informed two player games using different AI techniques.
4. Solve the AI problems by using logic programming
5. Apply uncertainty theory based on techniques like probability theory and fuzzy logic.

Text Book:

1. Stuart Russel and Peter Norvig; Artificial Intelligence: A Modern Approach; Third Edition; Pearson Education, 2009.

Reference Books:

1. E.Rich, K. Knight, S. B. Nair; Artificial Intelligence; 3rd Edition; Tata McGraw Hill, 2014.
2. Denis Rothman; Artificial Intelligence By Example: Develop machine intelligence from scratch using real artificial intelligence use cases; Kindle Edition, Packt Publishing Ltd, 2018

Syllabus for Semester IV, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDP4002 **Category:** Program Core Course (PCC)

Course: Artificial Intelligence Lab **L:** 0Hrs**, T:** 0Hr**, P:** 2Hr**, Per Week, Credits:** 1

Course Outcomes:

On successful completion of the course, students will be able to:

1. Implement different AI toy problems by using search techniques.
2. Design two player games using min-max algorithm with Alpha-Beta pruning.
3. Simulate AI problems using logic programming.
4. Implement algorithms to handle uncertainty.

PRACTICALS BASED ON CDT4002 SYLLABUS

Reference Books

1. Stuart Russel and Peter Norvig; Artificial Intelligence: A Modern Approach; Third Edition; Pearson Education.
2. E.Rich, K. Knight, S. B. Nair; Artificial Intelligence; 3rd Edition; Tata McGraw Hill.
3. Denis Rothman; Artificial Intelligence By Example: Develop machine intelligence from scratch using real artificial intelligence use cases; Kindle Edition, Packt Publishing Ltd, 2018

Syllabus for Semester IV, B. Tech. Computer Science & Engineering (Data Science)
Course Code: CDP4003 **Category:** (PCC)
Course: Design and Analysis of Algorithms
L: 3Hr**, T:** 0Hr**, P:** 0Hrs**, Per Week, Credits: 3**

Course Objectives:

The objective of this course is to familiarize the prospective engineers with:

1. Techniques of effective problem solving in computing.
2. Analysis of different paradigms of problem solving to solve a given problem in efficient way.

Syllabus

UNIT-I

Algorithm Design Basics and Recurrence Relations

Principles of designing algorithms and complexity calculation, Asymptotic notations for analysis of algorithms, worst case and average case analysis, amortized analysis and it's applications. Recurrence relations and their solutions using substitution, recurrence tree and master theorem methods

UNIT-II

Divide and Conquer technique

Introduction, Basic strategy, Binary Search, finding Maximum and Minimum elements from array, Quick sort, Merge sort, Strassen's matrix multiplication, Maximum sub-array problem, Closest pair of points problem.

UNIT-III

Greedy technique

Introduction, Basic strategy, fractional knapsack problem, Minimum cost spanning trees, Huffman Coding, activity selection problem, Scheduling problem with and without deadlines, find maximum sum possible equal to sum of three stacks, K Centers Problem.

UNIT-IV

Dynamic Programming technique

Introduction, Basic strategy, difference between Dynamic and Greedy approach, Bellmen ford algorithm, all pairs shortest path algorithm, multistage graphs, optimal binary search trees, traveling salesman problem, String Editing, Longest Common Subsequence problem and its variations.

UNIT-V

Backtracking and Branch and bound techniques

Introduction, Basic strategy, N-Queen's problem, Vertex coloring problem, Hamiltonian circuit problem, sum of subset problem, Branch and bound general technique, applications, travelling sales person problem.

UNIT-VI

P and NP class

Basic concept, non-deterministic algorithms NP-hard and NP-complete problems, Cook's Theorem, decision and optimization problems, polynomial reduction, Introduction to Approximation algorithm , vertex cover problem, clique cover problem

Course Outcomes

After successful completion of this course, the student will be able to,

1. Use mathematical formulation, complexity analysis and methodologies to solve the recurrence relations for algorithms.
2. Apply Greedy and Divide and Conquer algorithms and their usage in real life examples.
3. Apply Dynamic programming and Backtracking and branch and bound Paradigms to solve the real life problems.
4. Analyze P and NP class problems and formulate solutions using standard approaches.

Text Books

1. Thomas H. Cormen et.al; "Introduction to Algorithms"; 3 Edition; Prentice Hall, 2009.
2. Horowitz, Sahani and Rajasekaram; "Computer Algorithms", Silicon Press, 2008.

3. Brassard and Bratley; “Fundamentals of Algorithms”, 1 Edition; Prentice Hall, 1995.
4. Richard Johnsonbaugh, “Algorithms”, Pearson Publication, 2003.

Reference books

1. Parag Himanshu Dave, Balchandra Dave, “Design and Analysis of Algorithms” Pearson Education, O’relly publication
2. Richard Johnsonbaugh, “Algorithms”, Pearson Publication, 2003.

Syllabus for Semester IV, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDP4005 **Category:** (VSEC)

Course: Software Laboratory – II **L:** 0Hr**, T:** 0Hr**, P:** 2Hrs**, Per Week, Credits:** 1

Course Objectives:

The objective of this course is to impart necessary and practical knowledge of recent Software strategies based on frameworks and development skills required to design real-life web based projects by using advanced object oriented programming by :

1. Designing Enterprise applications by encapsulating an application's business logic.
2. Designing and developing multi-tier web applications
3. Designing applications using existing frameworks like Spring and Hibernate.

Experiments based on:

JDBC Java Database Connectivity (JDBC): The Design of JDBC, Basics of Structured Query Language, JDBC Configuration, Executing a basic SQL Statement.

Servlet: Handling the Client Request, Generating HTML response, reading form Data from Servlets: Reading Three Parameters, managing a session.

Java Server Pages (JSP): Invoking Java Code with JSP Scripting Elements, JSP directives and actions, Integrating Servlets and JSP

Spring Framework: Spring Bean Life Cycle, Spring Bean Scope, Basic Bean Wiring

Hibernate: Hibernate Configuration, Hibernate Sessions, Collections Mappings, Hibernate Query Language

Course Outcomes

On successful completion of the course, the student will be able to:

1. Implement Software based database application.
2. Demonstrate server and client-side programming using servlets and Java server pages.
3. Perform Dependency Injection using Spring, and create mappings in Hibernate using HQL.

Text Books:

1. M. Deitel, P. J. Deitel, S. E. Santry; Advanced Java 2 Platform HOW TO PROGRAM; Prentice Hall.

2. Cay Horstman, Gary Cornell; Core JAVA Volume-II Advanced Features; 8th Edition Craig Walls; Spring In Action; 2nd Edition

3. Marty Hall, Larry Brown; Core Servlets and Java Server Pages Volume-1: Core Technologies; 2nd Edition

Reference Books:

1. Jim Keogh; "J2EE:The Complete Reference"; McGraw Hill; Fifth Edition.

2. Spring Framework Documentation https://spring.io/

3. Hibernate Framework architecture Documentation https://hibernate.org/

Syllabus for Semester IV, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDP4006 **Category:** (CEP)

Course: Community Engagement Project

L: 0Hrs**, T:** 0Hr**, P:** 4Hr**, Per Week, Credits:** 2

Course Objectives

The objective of Community Engagement Project is to instill a sense of social responsibility amongst the students, empowering them to apply their knowledge and skills to positively impact and contribute to the society.

Execution Plan for the Subject:

- The students will impart their knowledge and skills in the society by identifying the potential needs or identify a society need and address it by building a technical solution.

Course Outcomes

On successful completion of the course, students will be able to:

1. Propose a community engagement project tailored to address society needs by devising a strategy or solution to address it.
2. Apply technical knowledge or skills towards execution of the proposed solution.
3. Evaluate the effectiveness of the project in addressing community needs.
4. Demonstrate ethical principles, project management skills, team work and communication skills for project completion within the confines of a deadline.

Syllabus for Semester V, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT5001 **Course: Database Management Systems**

L:3 Hrs, T:0 Hrs, P:0 Hrs Per Week **Total Credits: 3**

Course Objectives

1. To understand the role of a database management system in an organization.
2. To construct simple and advanced database queries using a data language.
3. To understand and apply logical database design principles and database normalization.
4. To recognize the need for transaction management and query processing.

SYLLABUS

UNIT-I Introduction to Database System Concepts and Architecture

Databases and Database Users, Characteristics of the Database Approach, Advantages of Using the DBMS Approach, When Not to Use a DBMS, Data Models, Schemas, and Instances, Three-Schema Architecture and Data Independence, Database Languages and Interfaces, The Database System Environment.

UNIT-II The Relational Data Model and SQL

Relational Model Concepts, Relational Model Constraints and Relational Database Schemas, Update Operations, Transactions, and Dealing with Constraint Violations, SQL Data Definition, Data Types and Constraints, Data Management in SQL, Transforming ER Model into Relational Model.

UNIT-III Database Design and Normalization

Functional Dependencies, Inference Rules, Equivalence, and Minimal Cover, Properties of Relational Decomposition, Normal Forms Based on Primary Keys, General Definitions of Second and Third Normal Forms, Boyce-Codd Normal Form, Other Dependencies and Normal Forms.

UNIT IV Indexing and Hashing

Ordered Indices, B+-Tree Index Files and its Extensions, Static Hashing and Dynamic Hashing, Comparison of Ordered Indexing and Hashing, Bitmap Indices, Some General Issues Concerning Indexing.

UNIT V Query Processing and Optimization

Measures of Query Cost, Query Operation: Selection, Sorting and Join Operation, Transformation of Relational Expressions, Estimating Statistics of Expression Results, Choice of Evaluation Plans.

UNIT VI Transaction Processing, Concurrency Control and Recovery

Introduction to Transaction Processing, Characterizing Schedules Based on Recoverability, Characterizing Schedules Based on Serializability, Two-Phase Locking Techniques for Concurrency Control, Deadlock Handling and Multiple Granularity, Database Recovery Techniques.

Course Outcomes:

On completion of the course the student will be able to

1. Identify the basic concepts and various data model used in database design.
2. Recognize the use of normalization and functional dependency.
3. Understand the purpose of query processing and optimization.
4. Apply and relate the concept of transaction, concurrency control and recovery in database.

Text Books:

1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan; "Database System Concepts" Sixth Edition, Tata McGraw Hill, 2011.

2. Ramez Elmasri and Shamkant Navathe; "Fundamentals of Database Systems", Sixth Edition, Addison Wesley 2011.

Reference Books:

1. Raghu Ramakrishnan and Johannes Gehrke; "Database Management Systems"; Third Edition; Tata McGraw Hill Publication, 2003.
2. C. J. Date; "Database in Depth – Relational Theory for Practitioners"; O`Reilly Media, 2005.

Syllabus for Semester V, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDP5001 Course: Database Management Systems Lab

L:0 Hrs, T:0 Hrs, P:2 Hrs Per Week Total Credits: 1

Course Objectives

1. To enable students to use DDL, DML and DCL.
2. To prepare students to conceptualize and realize database objects (tables, indexes, views and sequences) and execute SQL queries.
3. To encourage students to design and execute PL/SQL blocks and triggers.

Syllabus:

DBMS lab based on Theory syllabus

Course Outcome

On completion of the course the student will be able to

1. Demonstrate database user administration and authorizations.
2. Execute simple, nested, multiple table, and advanced queries for data retrieval.
3. Construct PL-SQL block structure and Trigger for specific application.
4. Implement various integrity constraints, views, sequences, indices and synonym on database.

Reference Books

1. James Groff, Paul Weinberg and Andy Oppel, SQL - The Complete Reference, 3rd Edition, McGraw Hill, 2017.

Syllabus for Semester V, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT5002 Course: Machine Learning

L:3 Hrs, T:0 Hrs, P:0 Hrs Per Week Total Credits: 3

Course Objectives

1. To introduce the basic concepts and techniques of machine learning.
2. To understand major machine learning algorithms.
3. To identify machine learning techniques suitable for a given problem.

Syllabus

UNIT – I

Concept Learning: The concept learning task, General-to-specific ordering of hypotheses, Version spaces, Inductive bias, Decision Tree Learning, Rule Learning: Propositional and First-Order, Over-fitting, Cross Validation, Experimental Evaluation of Learning Algorithms.

UNIT – II

Instance-Based Learning: K-Nearest neighbor algorithm, Radial basis functions, Case- based learning. Computational Learning Theory: probably approximately correct (PAC) learning, Sample complexity, Computational complexity of training, Vapnik Chervonenkis dimension.

UNIT – III

Artificial Neural Networks: Linear threshold units, Perceptron, Multilayer networks and backpropagation, recurrent networks.

UNIT – IV

Probabilistic Machine Learning: Maximum Likelihood Estimation, MAP, Bayes Classifiers Naïve Bayes, Bayes optimal classifiers, Minimum description length principle.

Bayesian Networks, Inference in Bayesian Networks.

UNIT – V

Expectation Maximization algorithm, preventing over fitting, Gaussian Mixture Models, K- means and Hierarchical Clustering.

Hidden Markov Models, Reinforcement Learning, Support Vector Machines, Ensemble learning: boosting, bagging.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Solve the problems related to the fundamental concepts in machine learning.
2. Apply machine learning algorithms to solve classification, regression and clustering problems.
3. Analyze the strengths and weaknesses of various machine learning approaches.
4. Apply various machine learning models to efficiently solve real-world problems.

Text Books

1. Tom Mitchell; Machine Learning- an Artificial Intelligence Approach, Volume-II; Morgan Kaufmann, 1986.
2. Christopher Bishop, Pattern Recognition and machine learning; Springer Verlag, 2006.

Reference Books

1. Soumen Chakrabarti; Mining the Web: Discovering Knowledge from Hypertext Data, Morgan Kaufmann, 2003.
2. A. K. Jain and R. C. Dubes; Algorithms for Clustering Data; Prentice Hall PTR, 1988.
3. Ethem Alpaydin, Introduction to Machine Learning, PHI.

Syllabus for Semester V, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDP5002 **Course: Machine Learning Lab**

L:3 Hrs, T:0 Hrs, P:0 Hrs Per Week **Total Credits: 3**

Course Objectives

1. To implement basic machine learning algorithm for solving problem.
2. To understand the usage of datasets in implementing machine learning problems.
3. To learn various modern tools, packages and techniques for machine learning.

Syllabus Technology: Python.

- To implement Find-S algorithm to find maximally specific hypothesis on given dataset.
- To implement Candidate Elimination algorithm on Enjoy Sports dataset and find candidate hypothesis.
- To implement linear regression algorithm on given dataset.
- To implement decision tree classifier on given dataset and display generated tree.
- To Implement KNN algorithm for classification and regression on given datasets.
- To implement the perceptron algorithm for AND, OR and NOR Boolean functions.
- To implement the backpropagation algorithm for machine learning using suitable dataset.
- To implement the Naïve Bayes algorithm on given dataset.
- To implement the support vector machine algorithm on given dataset.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Implement fundamental concepts used in machine learning algorithms.
2. Implement python programs for various learning algorithms.
3. Apply appropriate machine learning algorithms to various data sets.
4. Apply machine learning algorithms to solve real world problems.

Text Books

1. Tom Mitchell; Machine Learning- an Artificial Intelligence App roach, Volume-II; Morgan Kaufmann, 1986.
2. Christopher Bishop, Pattern Recognition and machine learning; Springer Verlag, 2006.

Reference Books

1. Soumen Chakrabarti; Mining the Web: Discovering Knowledge from Hypertext Data, Morgan Kaufmann, 2003.
2. A. K. Jain and R. C. Dubes; Algorithms for Clustering Data; Prentice Hall PTR, 1988
3. Ethem Alpaydin, Introduction to Machine Learning, PHI.

Syllabus for Semester V, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT5003 **Course: Computer Networks**

L:3 Hrs, T:0 Hrs, P:0 Hrs Per Week **Total Credits: 3**

Course Objectives

1. To Introduce the fundamental concepts of each layer in the OSI and TCP/IP models.
2. To implement, and troubleshoot network topologies and examine network and transport Layer protocol working.
3. To learn about network security, firewalls, and intrusion detection.
4. To Investigate applications of AI in computer network.

Syllabus

UNIT – I

Data communication Components: Representation of data and its flow Networks, Various Connection Topology, Protocols and Standards, OSI model, Data Link Layer: Error Detection and Error Correction - Hamming Distance, CRC; Flow Control and Error control protocols - Stop and Wait, Go back – N ARQ, Selective Repeat ARQ. IEEE 802.3.

UNIT – II

Network Layer: Internet Protocol (IP) – Logical Addressing: IPV4, IP addressing and subnetting, IPV6, ARP, RARP, BOOTP and DHCP–Delivery, Routing protocols.

UNIT – III

Transport Layer: Elements of Transport protocols: Addressing, Connection establishment, Connection release, User Datagram Protocol (UDP), Transmission Control Protocol (TCP). TCP congestion control. Traffic shaping Leaky Bucket and token bucket. Algorithm.

UNIT IV

Application Layer: Domain Name Space (DNS), File Transfer Protocol (FTP), WWW, HTTP, Network security principles and threats, Cryptographic techniques (symmetric/asymmetric encryption), SSL/TLS protocols.

UNIT – V

Firewalls, VPNs, and IPS/IDS (Intrusion Detection and Prevention Systems), Bluetooth. Load Balancing Optimization using AI/ML, AI-based Threat Hunting in Network Security.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Implement the computer network protocols with topologies.
2. Apply error detection and correction mechanisms related to data Link Layer and implement Routing algorithm.
3. Analyze TCP protocol related to traffic shaping and routing algorithms.
4. Implement network security mechanisms to protect data and infrastructure.

Text Books

1 "Computer Networks" by Andrew S. Tanenbaum, David J. Wetherall (5th Edition), Pearson Education

2 "Data and Computer Communications" by William Stallings (10th Edition), Pearson Education

3 Data Communication and networking by Behrouz Forouzan (4th Edition) Mc Graw Hill Publication.

4 Cryptography And Network Security by Behrouz Forouzan (3rd Edition) Mc Graw Hill Publication.

Reference Books

1. "Computer Networking: A Top-Down Approach" by James F. Kurose, Keith W. Ross (7th Edition), Pearson Education.

2. "Network Security Essentials" by William Stallings (5th Edition), Pearson Education.

Syllabus for Semester V, B. Tech. Computer Science and Engineering (Data Science)

Course Code: CDT5004 Course: Compiler Design

L:3 Hrs, T:0 Hrs, P:0 Hrs Per Week Total Credits: 3

Course Objectives

The main objective of this course is to introduce the fundamental concepts of compiler design and language translation. It aims to develop an understanding of the structure, function, and complexity of modern compilers. The students will learn the various phases of compilation with practical implementation using compiler writing tools.

Syllabus UNIT I

Introduction to Compilers, Phases of Compiler, Relating Compilation Phases with Formal Systems, Lexical Analysis, tokens, pattern and lexemes, Design of Lexical analyser, Regular Expression, transition diagram, recognition of tokens, Lexical Errors.

UNIT II

Syntax Analysis- Specification of syntax of programming languages using CFG, Top-down parser, design of LL (1) parser, bottom-up parsing technique, LR parsing, Design of SLR, CLR, LALR parsers, Handling Ambiguous Grammars, Applications of the LR Parser.

UNIT III

Syntax directed translation- Study of syntax directed definitions & syntax directed translation schemes, Type and Type Checking, Implementation of SDTS, Intermediate notations, translation of Assignment Statement, controls structures, Array reference.

UNIT IV

Code optimization- machine independent Optimisation, Local optimization techniques, loop optimization- control flow analysis, data flow analysis, Loop invariant computation, Induction variable removal, other loop optimization techniques, Machine-dependent Optimization techniques. Code generation- Problems in code generation, Simple code generator, code generation using labelling algorithm, code generation using gencode algorithm

UNIT V.

Storage allocation & Error Handling- Run time storage administration, stack allocation, Activation of Procedures, Storage Allocation Strategies, Garbage Collection, symbol table management, Error handling, Error detection and recovery- lexical, syntactic and semantic, Error recovery in LL & LR Parser

Course Outcomes:

On successful completion of the course, students will be able to:

1. Exhibit role of various phases of compilation, with understanding of types of grammars and design complexity of compiler.
2. Design various types of parses and perform operations like string parsing and error handling.
3. Demonstrate syntax directed translation schemes, their implementation for different programming language constructs.
4. Implement different code optimization and code generation techniques using standard data structures.

Text Books

1. Aho, Sethi, and Ullman; Compilers Principles Techniques and Tools; Second Edition, Pearson education, 2008.
2. Alfred V. Ahoand Jeffery D. Ullman; Principles of Compiler Design; NarosaPub.House, 1977.
3. VinuV. Das; Compiler Design using Flex and Yacc; PHI Publication, 2008.
4. Manoj B Chandak, Khushboo P Khurana; Compiler Design; Universities Press, 2018.

Reference Books

1. Vinu V. Das; Compiler Design using Flex and Yacc; PHI Publication
2. V. Raghavan; Principles of Compiler Design, McGraw Hill Education (India)

Syllabus for Semester V, B. Tech. Computer Science and Engineering (Data Science)

Course Code: CDT5004 Course : Design Patterns

L:3 Hrs, T:0 Hrs, P:0 Hrs Per Week Total Credits: 3

Course Objectives

1. To learn the fundamentals of software design by referring a catalog of design patterns
2. Demonstrate how to use design patterns to address code development issues.
3. Identify the most suitable design pattern to address a given application design problem.
4. Apply design principles (e.g., open-closed, dependency inversion, substitution, etc).
5. Critique code by identifying and refactoring anti-patterns.

Syllabus UNIT-I:

Elements of Design Pattern, Describing Design Pattern, Design Pattern Classification, Role of design patterns in software design, Selection and usage of Design Patterns, Example implementation of design pattern using UML, Case Study: Designing a Document Editor.

UNIT-II:

Creational Patterns: Introduction, Role of Creational patterns, Creational Pattern types: Factory method, Abstract Factory, Builder, Prototype, Singleton, Comparative study of creational patterns, and examples based on real life applications.

UNIT-III:

Structural Design Patterns: Introduction, Role of Structural patterns, Encapsulating complex structures to simplify interactions between components, Decoupling Components, Structural Pattern types: Adapter, Bridge, Composite, Decorator, Façade, Proxy, Comparative study of structural patterns, and examples based on real life applications.

UNIT-IV:

Behavioral Patterns-I: Introduction, Role of Behavioral pattern, Encapsulation of Behavior, Behavioral Pattern types: Chain of Responsibility, Template Method, State, Strategy, and Iterator.

UNIT-V:

Behavioral Patterns-II: Effect of single object on set of objects, Analysis of mutual behavior of classes and object's state, Reference control between objects, Behavioral Pattern types: Observer, Mediator, Memento, Interpreter, Comparative study of Behavioral patterns, and examples based on real life applications.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Analyze the need and ability of design patterns in the software design process.
2. Implement various solutions for creation of objects, their structure and the interaction between objects.
3. Develop a loosely coupled application using design patterns.
4. Analyze the tradeoffs of applying a design pattern to a given problem.

Text Books:

1. Design Patterns: Elements of reusable object-oriented software by Gamma Erich, Helm Richard, Johnson Ralph, and Vlissides John, Pearson Education
2. Design Patterns Explained by Alan Shallowly and James Trott, Addison-Wesley

Reference Books:

1. Pattern's in JAVA Vol-I by Mark Grand, WileyDreamTech.
2. JAVA Enterprise Design Patterns Vol-III by Mark Grand, WileyDreamTech.
3. Head First Design Patterns by Eric Freeman, O'Reilly.

Syllabus for Semester V, B. Tech. Computer Science and Engineering (Data Science) Course Code: CDT5004 Course: Software Engineering

L:3 Hrs, T:0 Hrs, P:0 Hrs Per Week Total Credits: 3

Course Objectives

1. To familiarize students with the fundamentals of software engineering principles and practices.
2. To aid students in designing software systems using structured and object-oriented approaches.
3. To apprise students with different software testing and debugging strategies in building a quality software.
4. To introduce students to the practice of software project management.

Syllabus Unit-I

The Evolving Role of Software - Software Characteristics, Applications, Principles and Myths; Software Engineering as a Layered Technology; Software Process Framework.

Software Process Models - Waterfall Model, Evolutionary Models, Unified Process Model, Agile Process Models, Extreme Programming (XP), Scrum Model; Requirements Engineering.

Unit-II

Requirements Analysis, Analysis Modeling Approaches; Data Modeling, Object-Oriented Analysis, Scenario-Based Modeling, Flow-Oriented Modeling, Class-based Modeling, Behavioral Model, Design Concepts, The Design Model, Component Level Design, User Interface Design.

Unit-III

Basic concepts of Testing, Software Testing Life Cycle (STLC), Verification and Validation, Unit Testing, Integration Testing, Validation Testing, System Testing, Art of Debugging. White-Box Testing, Basis Path Testing, Control Structure Testing, Black-Box Testing, Equivalence Partitioning, Boundary Value Analysis, Web Testing, Test case design, Building, Execution, Automated Testing.

Unit-IV

Software Project management- Plans, Methods and Methodology; Project Success and Failure, Project Evaluation, Cost-benefit evaluation technique, Project Planning & Scheduling.Software Effort Estimation- Albrecht Function Point Analysis, COSMIC Function Point, Cost Estimation, COCOMO Model, Project Scheduling. Software Quality.A Framework for Product Metrics, Metrics for Analysis & Design Models, Metrics for Source Code, Metrics for Testing & Maintenance. Metrics for process & project - Software measurement.

Unit-V

Risk management - Risk strategies, Software risks, Risk identification, Risk refinement, RMMM Risk Response development & Risk Response Control, Risk Analysis. Change Management-Software Configuration Management, SCM Repository, SCM Process, Estimation, Reengineering- Software reengineering, Reverse engineering.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Elucidate software engineering practices and various process models.
2. Use software design approaches for designing real-time solutions.
3. Demonstrate White Box Testing and Black Box Testing for building bug-free quality software.
4. Integrate software project management practices in software product development.

Text Books

1. Roger S. Pressman and Bruce R. Maxim; Software Engineering – A Practitioner's Approach; Eighth Edition, McGraw Hill; 2015.
2. Ian Somerville; Software Engineering; Seventh Edition; Pearson Education. 2008.

Reference Books

1. Pankaj Jalote; An Integrated Approach to Software Engineering; Third Edition, Springer, 2005.
2. Rajib Mall; Software Project Management, 5th Edition, McGrawHill.
3. David Gustafsan; Software Engineering;Schaum's Series,Tata McGraw Hill,2002.

Syllabus for Semester VI, B. Tech. Computer Science and Engineering (Data Science)

Course Code: CDT6001 Course: Data Warehousing and Business Intelligence

L:3 Hrs, T:0 Hrs, P:0 Hrs Per Week Total Credits: 3

Course Objectives

- To understand data warehouse concepts and conduct dimensioning modeling in building data warehouses
- To study the basics of data integration and ETL technology
- To know business intelligence architecture and technologies.
- To study enterprise reporting
- To know future trends in BI

Syllabus:

UNIT-I:

Architecture of DW, OLTP vs OLAP, Dimensional modelling : dimensions and facts, Star and Snowflake schema, pros & cons of the Star / Snowflake schema, Types of dimension and facts.

UNIT-II:

Basics of data preprocessing, concepts of data integration and data transformation, data reduction, data summarization, introduction to data quality, data profiling concepts and applications. Extraction, Transformation, and Load.

UNIT-III:

Space management in data warehouses : B-tree indexes, bitmap indexes, clusters, hash indexes, functional indexes, table partitioning.

UNIT-IV:

Introduction to BI, Leveraging Data and knowledge for BI, BI Components, BI Dimensions, Information Hierarchy, Business Intelligence and Business Analytics. BI Architecture , BI Life

Cycle. Role of Data Warehousing in BI, data warehousing building blocks, Metadata in the data warehouse

UNIT-V:

Enterprise Reporting : Introduction to business metrics and KPIs, Basics of Enterprise Reporting: Introduction to enterprise reporting, concepts of dashboards, balanced scorecards, Business Activity Monitoring, Six Sigma. Memento Design pattern and its implementation

Course Outcome:

After the completion of the course, the students will be able to:

1. Apply the concepts of data warehousing and space management for BI problems.
2. Demonstrate knowledge of data preprocessing techniques.
3. Apply appropriate BI methods to find solutions to business problems

Text Books:

1. Business Intelligence: A Managerial Perspective on Analytics, 3rd Edition, Ramesh Sharda, Dursun Delen, Efaim Turban, Prentice Hall 2013
2. Fundamentals of Business Analytics, R N Prasad and S Acharya, Wiley India.
3. Paulraj Ponnian, - Data Warehousing Fundamentals||, John Willey

Syllabus for Semester VI, B. Tech. Computer Science and Engineering (Data Science)

Course Code: CDP6001 Course: Data Warehousing and Business Intelligence lab

L:0 Hrs, T:0 Hrs, P:2 Hrs Per Week Total Credits: 1

Course Outcomes:

On successful completion of the course, students will be able to:

1. Implement the ETL process.
2. Design a data warehouse
3. Execute OLAP queries on a data warehouse.
4. Create good reports.

Practical based on syllabus of CDT6001.

Syllabus for Semester VI, B. Tech. Computer Science and Engineering (Data Science)

Course Code: CDT6002 **Course**: **Deep Learning - I**

L:3 Hrs, T:0 Hrs, P:0 Hrs Per Week **Total Credits: 3**

Course Objectives

1. To introduce basic deep learning algorithms.
2. To understand real-world problems which can be solved by deep learning methods.
3. To identify deep learning techniques suitable for a real-world problem.

Syllabus:

UNIT I: Basics of Deep Learning

History of Deep Learning, McCulloch Pitts Neuron, Thresholding Logic, Perceptrons,, Perceptron Learning Algorithm and Convergence, Multilayer Perceptrons (MLPs), Representation Power of MLPs, Sigmoid Neurons.

UNIT II: Training of Feedforward Neural Networks

Feedforward Neural Networks, Representation Power of Feedforward Neural Networks, Training of Feedforward Neural Networks, Gradient Descent (GD), Momentum Based GD, Nesterov Accelerated GD, Stochastic GD, AdaGrad, RMSProp, Adam

UNIT III: Optimization Algorithm

Activation Function and Initialization Methods: Sigmoid, Tanh, ReLU, Xavier and He Initialization, Regularization: Bias and Variance, Overfitting, Hyperparameters Tuning, L1 and L2 Regularization, Data Augmentation and Early Stopping, Parameter Sharing and Tying.

UNIT IV: Convolutional Neural Network (CNN)

Convolutional Neural Networks, 1D and 2D Convolution, Visualizing Convolutional Neural Networks, Guided Backpropagation.

UNIT V: Recurrent Neural Network (RNN)

Recurrent Neural Networks, Backpropagation Through Time (BPTT), Vanishing and Exploding Gradients, Long Short-Term Memory (LSTM) Cells, Gated Recurrent Units (GRUs). Variants of CNN and RNN : Encoder-Decoder Models, Attention Mechanism, LeNet, AlexNet, ZF-Net, VGGNet, GoogLeNet, ResNet.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Apply fully connected deep neural networks to real-world problem-solving scenarios.
2. Evaluate the performance of various deep learning models in terms of optimization, bias-variance trade-off, overfitting, and underfitting.
3. Analyze the role of convolutional and recurrent neural networks in addressing different real-world problems.
4. Create advanced deep learning models by designing variants of CNNs and RNNs tailored to specific applications.

Text Books

1. Sandro Skansi, *Introduction to Deep Learning*, Springer
2. Charu C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*, Springer, 2019
3. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016
4. Dr. S. Lovelyn Rose, Dr. L. Ashok Kumar, Dr. D. Karthika Renuka, *Deep Learning using Python*, Wiley Publication

Reference Books:

1. Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer, 2006
2. Yegnanarayana, B., *Artificial Neural Networks*, PHI Learning Pvt. Ltd., 2009
3. A. Ravindran, K. M. Ragsdell, and G. V. Reklaitis, *Engineering Optimization: Methods and Applications*, John Wiley & Sons, Inc., 2016

Syllabus for Semester VI, B. Tech. Computer Science and Engineering (Data Science)

Course Code: CDP6002 **Course**: **Deep Learning – I Lab**

L:0 Hrs, T:0 Hrs, P:2 Hrs Per Week **Total Credits: 1**

Course Objectives:

1. To solve problems in linear algebra, probability, optimization using artificial neurons.
2. To understand the usage of publically available datasets.
3. To use various python packages and tools for deep learning.

Course Syllabus:

Experiments based on CDT6002

Course Outcomes:

On completion of the course the student will be able to

1. Identify the deep learning algorithms which are more appropriate for various types of learning tasks in various domains.
2. Implement deep learning models in Python using the PyTorch/Tensorflow library and train them with real-world datasets.
3. Analyze and evaluate deep learning model's.

Text Books:

1. Sandro Skansi, Introduction to Deep Learning, Springer
2. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.

Reference Books:

1. Francois Chollet, Deep Learning with Python, Manning Publications Co.
2. Golub, G., H., and Van Loan, C., F., Matrix Computations, JHU Press, 2013.

Syllabus for Semester VI, B. Tech. Computer Science and Engineering (Data Science) Course Code: CDT6003 Course: Data Analysis and Visualization

L:1 Hrs, T:0 Hrs, P:0 Hrs Per Week Total Credits: 01

Course Objectives

- To understand data analytics life cycle for solving challenging business problems
- To adopt appropriate statistical procedures for analysis based on goals and nature of data
- To employ best practices in data visualization to develop charts, maps tables and other visual representations of data

Syllabus

Introduction and Overview:

Importance of analytics and visualization, data preprocessing, Basic Analysis Techniques, Data Analytics Lifecycle and Different Phases

Association Rules and Regression:

Apriori Algorithm, Evaluation of Candidate Rules, Frequent Itemsets and Rule Generation, Validation and Testing, Diagnostics, Linear Regression, Logistic Regression, Choice of a Model.

Time Series Analysis: Box-Jenkins Methodology, ARIMA (Auto Regressive Integrated Moving Average) Model, Choice of a Model, Overview of ARMAX, Spectral Analysis and GARCH.

Data Visualization with Tools:

Principles of effective visualization, mapping data to uncertainty (error bars) using Tableau, Power BI, and Dash. Hands-on creation of interactive charts: bar/line/scatter plots, geographic maps, filters (dimension, date, measure), and custom calculations in Tableau; DAX measures and table visuals in Power BI; Plotly Dash apps for web-based dashboards. Commonly used file formats (CSV, Excel, JSON) and software integration for real-world datasets.

Creating Stories with Data

Dash for Python-coded interactive storytelling with callbacks. Case studies: Superstore analysis, fiscal calendars, profitability KPIs using these tools; Building interactive dashboards: Combine

visuals with event listeners, layouts, and cross-filters in Tableau (publish to Tableau Public); Power BI reports/dashboards with AI visuals and sharing.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Apply data preprocessing and basic data analysis techniques.
2. Conduct data analytics using scientific methods.
3. Analyze time series data.
4. Create presentations and visualizations.

Text Books

1. David Dietrich, Barry Heller and Beibel Yang, - Data Science and Big Data Analytics – Discovering, Analyzing, Visualizing, and Presenting Data, John Wiley and Sons [EMC Education Services], 2015.
2. Claus O. Wilke, - Fundamentals of Data Visualization – A Primer on Making Informative and Compelling Figures, O'Reilly, 2019.
3. Python: Data Analytics and Visualization, Packt Publishing, 2017.

Reference Books

1. Jiawei Han, Micheline Kamber and Jian Pei, - Data Mining Concepts and Techniques, 3rd edition; Morgan Kaufmann Publishers, 2011.
2. "Storytelling with Data: A Data Visualization Guide for Business Professionals" by Cole Nussbaumer Knaflic (2nd Edition, Wiley, 2020) – Focuses on practical visualization principles, audience-driven narratives, and real-world examples.
3. "The Big Book of Dashboards: Visualizing Your Data Using Real-World Business Scenarios" by Steve Wexler et al. (Wiley, 2017) – Offers case studies on dashboard design with tools like Tableau and Power BI.
4. "Data Visualization: A Practical Introduction" by Kieran Healy (Princeton University Press, 2018) – Covers R/Python tools for reproducible graphics, aligning with student coding needs.

Syllabus for Semester VI, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDP6003 Course: Data Analysis and Visualization Lab

L:0 Hrs, T:0 Hrs, P:1 Hrs Per Week Total Credits: 1

Course Objectives

1. To apply statistical methods for data analytics to provide business solutions.
2. To develop insights based on analytical results to facilitate better understanding of consumer attitude, perceptions and behavior.
3. To create data visualizations for effective communication to user.

Syllabus

Experiments based on Data Visualization and Analytics Theory Syllabus preferably using R, Python.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Apply different data preparation techniques.
2. Apply various data analysis techniques.
3. Apply analytics on time series data.
4. Design effective presentations and visualizations.

Reference Books

1. David Dietrich, Barry Heller and Beibel Yang, - Data Science and Big Data Analytics – Discovering, Analyzing, Visualizing, and Presenting Data, John Wiley and Sons [EMC Education Services], 2015.
2. Kyran Dale, - Data Visualization with Python and JavaScript – Scrape, Clean and transform Your Data, O'Reilly, 2016.

Syllabus for Semester V, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT6004-1 Course: Natural Language Processing

L:3 Hrs, T:0 Hrs, P:0 HrsPer Week Total Credits: 3

Course Objectives

1. To familiarize the concepts and techniques of natural language processing.
2. To learn computational techniques that enable machines to process, understand, and generate human language efficiently.
3. To apply the statistical learning methods and cutting-edge research models to solve natural language processing problems.
4. To integrate natural language processing into real-world systems to develop, evaluate, and enhance applications.

Syllabus:

UNIT I

Introduction to NLP, Definition and Scope, A Brief History, Importance, Challenges, Tasks, Significance, NLP Pipeline and Applications, Morphological Analysis and Generation using Finite State Automata, Finite State Transducer, Hidden Markov model, Viterbi Algorithm, Applications of Tagging.

UNIT II

Lexical Analysis, Part-of-Speech (POS) Tagging, Approaches for POS Tagging, Rule-Based, Stochastic, Hybrid Approach, Taggers Evaluations, Tokenization with NLTK.

Syntax and Parsing, Types of Constituents in Sentences, Context-Free Grammar (CFG), CFG Parsing, Top-Down Parser, Bottom-Up Parser, Shallow Parsing and Chunking, Thematic Roles, Conditional Random Fields, Maximum Likelihood Estimation, Lexical and Probabilistic Parsing, Probabilistic Context Free Grammars, The Probability of a String, Inside-Outside Algorithm, CKY Parsing.

UNIT III

Semantic Analysis, Lexical Vs Compositional Semantic Analysis, Word Senses and Relations, Types of Lexical Semantics, Word Sense Disambiguation, WordNet and Online Thesauri, Word Similarity and Thesaurus Methods, Text Representation, Word Embedding, TF-IDF, Bag of Words, Word2Vec, Skip-gram.Pragmatic Analysis and Discourse, Discourse Phenomena, Coherence and Coreference, Importance of Coreference Relations, Discourse Segmentation, Algorithms for Coreference Resolution.

UNIT IV

N-Gram Language Model, Language Modeling and Chain Rule, Markov Chain in N-Gram Model, Shannon's Method in N-Gram Model, Smoothing Techniques, Extrinsic Evaluation Scheme, Zero Counts Problems, Smoothing Techniques, Laplace (Add-One) Smoothing, Add-k Smoothing, Backoff and Interpolation Smoothing, Good Turing Smoothing, The Transformer, Large Language Models, Language Model Evaluation, Entropy, Perplexity, ROUGE, BLEU.

UNIT V

Major NLP Applications, Information Retrieval Systems, Social Network Analysis, Sentiment Analysis, Information Extraction, Named Entity Recognition, Text Classification, Text Summarization Systems, Machine Translation, Word Alignment, Content Recommendation System, Answering Questions, Applications in Finance, E-Commerce, Travel and Hospitality, Marketing, Insurance, Healthcare, Law, Supply Chain, Telecommunication, Education and Research.

Course Outcomes

On Successful completion of course, students will be able to:

1. Understand core NLP concepts and techniques.
2. Apply various POS tagging approaches and parsing techniques to analyze sentence structure and utilize probabilistic models for syntactic analysis.
3. Analyze various semantic and pragmatic analysis techniques and discourse phenomena to enhance text representation and understanding.

4. Implement N-Gram language models and Transformer-based models for effective language modeling and text generation.
5. Design and develop innovative NLP solutions to address real-world challenges across industries like finance, healthcare, e-commerce, education and research.

Textbooks

1. Daniel Jurafsky and James H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, Third Edition, 2025, https://web.stanford.edu/~jurafsky/slp3.
2. Raymond ST. Lee, Natural Language Processing: A Textbook with Python Implementation, Springer Nature Singapore Pte Ltd. 2024, ISBN: 978-9819919987.
3. Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, and Harshit Surana, Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems, O'Reilly Media, Inc., USA, First edition, 2020, ISBN: 978-1492054054.
4. Dipanjan Sarkar, Text Analytics with Python: A Practitioner's Guide to Natural Language Processing, Second Edition, Apress Media, LLC, California, 2019, ISBN: 978-1484243534.

Reference Books

1. Natural Language Processing with Python: From Basics to Advanced Projects, Second Edition, 2024, Cuantum Technologies LLC. Plano, ISBN: 979-8894968483.
2. Jyotika Singh, Natural Language Processing in the Real World: Text Processing, Analytics, and Classification, First edition, 2023, CRC Press is an imprint of Taylor & Francis Group, LLC, ISBN: 978-1003264774.
3. Gerhard Paaß and Sven Giesselbach, Foundation Models for Natural Language Processing: Pre-trained Language Models Integrating Media, Artificial Intelligence: Foundations, Theory, and Algorithms, Springer Nature Switzerland Pte Ltd. 2022, ISBN: 978-3031231896.
4. Lewis Tunstall, Leandro von Werra, and Thomas Wolf, Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media, Inc., USA, Revised First edition, May 2022, ISBN: 978-1098136796.

Syllabus for Semester VI, B. Tech. Computer Science and Engineering (Data Science)

Course Code: CDP6004-1 Course: Natural Language Processing Lab

L:0 Hrs, T:0 Hrs, P:2 Hrs Per Week Total Credits: 1

Course Objectives

1. To familiarize the concepts and techniques of natural language processing.
2. To learn computational techniques that enable machines to process, understand, and generate human language efficiently.
3. To apply the statistical learning methods and cutting-edge research models to solve natural language processing problems.
4. To integrate natural language processing into real-world systems to develop, evaluate, and enhance applications.

Syllabus:

1. Experiments based on the above syllabus of CDT6004 .
2. Few lab sessions shall be conducted using virtual lab platforms to enhance learning experiences and accessibility.
3. Utilise Hugging Face, Stanford, Kaggle, MIT OpenCourseWare, OpenAI Learning, Fast.ai, AllenNLP like repositories and platforms.

Textbooks

1. Daniel Jurafsky and James H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, Third Edition, 2025, https://web.stanford.edu/~jurafsky/slp3.
2. Raymond ST. Lee, Natural Language Processing: A Textbook with Python Implementation, Springer Nature Singapore Pte Ltd. 2024, ISBN: 978-9819919987.

3. Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, and Harshit Surana, Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems, O'Reilly Media, Inc., USA, First edition, 2020, ISBN: 978-1492054054.
4. Dipanjan Sarkar, Text Analytics with Python: A Practitioner's Guide to Natural Language Processing, Second Edition, Apress Media, LLC, California, 2019, ISBN: 978-1484243534.

Reference Books

1. Natural Language Processing with Python: From Basics to Advanced Projects, Second Edition, 2024, Cuantum Technologies LLC. Plano, ISBN: 979-8894968483.
2. Jyotika Singh, Natural Language Processing in the Real World: Text Processing, Analytics, and Classification, First edition, 2023, CRC Press is an imprint of Taylor & Francis Group, LLC, ISBN: 978-1003264774.
3. Gerhard Paaß and Sven Giesselbach, Foundation Models for Natural Language Processing: Pre-trained Language Models Integrating Media, Artificial Intelligence: Foundations, Theory, and Algorithms, Springer Nature Switzerland Pte Ltd. 2022, ISBN: 978-3031231896.

Syllabus for Semester VI, B. Tech. Computer Science and Engineering (Data Science)

Course Code: CDT6004-2 **Course**: **Blockchain Technology**

L:3 Hrs, T:0 Hrs, P:0 Hrs Per Week **Total Credits: 3**

Course Objectives

This course aims to provide a survey on blockchain and the topics around such as history of blockchain, cryptography it uses, Bitcoin and other currencies, consensus algorithms, smart contracts, Ethereum, scalability and various use cases.

Syllabus

UNIT I

Blockchain Introduction: Blockchain Technology Mechanisms & Networks, Blockchain Origins, Blockchain Objectives, Blockchain Users & Adoption, Blockchain Challenges, P2P Systems, Hash Pointers and Data Structures, Blockchain Transactions

Unit II

Consensus Mechanism:Permissioned Blockchain, Permisionless Blockchain , Different Consensus Mechanism- Proof of Work, Proof of Stake, Proof of Activity, Proof of Burn, Proof of Elapsed Time, Proof of Authority, Proof of Importance.

Unit III

Cryptography Fundamentals:Encryption, Digital Signatures, Public-Key Cryptography, Private Key Cryptography, Distributed Denial-of-Service (DDoS) Attack, 51% Attack, Double spending problem, Merkel Tree, Security Threats to Blockchain Technology

Unit IV

Crypto currency and Wallet:Types of Wallet, Desktop Wallet, App based Wallet, Browser based wallet, Metamask, Creating a account in Metamask, Use of faucet to fund wallet, transfer of cryptocurrency in metamask.

Unit V

Smart contract and Ethereum:Overview of Ethereum, Writing Smart Contract in Solidity, Remix IDE , Different networks of ethereum, understanding blocks in blockchain, compilation and deployment of smart contracts in Remix

Course Outcomes:

On successful completion of the course, students will be able to:

1. Realize importance of blockchain technology and consensus mechanism
2. Identifying the security risks and challenges associated with blockchain technology
3. Implement browser based wallets and smart contracts in Remix IDE
4. Recognize the importance of blockchain security in various enterprise applications.

Text Books

1. Mastering Blockchain: Third Edition by Imran Bashier, Packt Publishing, 2020, ISBN: 9781839213199.

Reference Books

1. Blockchain: Blueprint for a New Economy by Melanie Swan, Oreilly Publication
2. Mastering Ethereum, by Andreas M. Antonopoulos, Gavin Wood
3. Bitcoin and Cryptocurrency Technologies (Princeton textbook) by Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder

Syllabus for Semester VI, B. Tech. Computer Science and Engineering (Data Science)

Course Code: CDP6004-2 **Course**: **Blockchain Technology Lab**

L:0Hrs, T:0 Hrs, P:2 Hrs Per Week **Total Credits: 1**

Course Objectives

This course aims to provide hands-on experience with blockchain development tools and frameworks, and will be able to use them to build and test their own blockchain projects.

Syllabus:

Experiments based on the above syllabus CDT6004-2

Course Outcome:

On completion of the course the student will be able to

1. Realize different blockchain tools and framework
2. Implement smart contracts for the development of enterprise applications
3. Apply different wallets for the deployments of smart contracts

Syllabus for Semester VI, B. Tech. Computer Science and Engineering (Data Science)

Course Code: CDT6004-3 Course: Distributed and Parallel Computing

L:3 Hrs, T:0 Hrs, P:0 Hrs Per Week Total Credits: 3

Course Objectives

1. To introduce the fundamental concepts of parallel and distributed computing, including types of parallelism, performance metrics, and architectural models.
2. To equip students with practical programming skills using parallel computing APIs (OpenMP, MPI, CUDA, Spark) for solving data- and compute-intensive problems.
3. To develop the ability to analyze, design, and optimize algorithms for multi-core, many-core, and distributed memory architectures using appropriate models and tools.

Syllabus

UNIT I:

Overview of parallel processing landscape: why and how, types of parallelism, Flynn's taxonomy and brief overview of parallel architectures, practical demonstration of CCR as an example HPC center. Basic concepts in parallel processing: formal definition of parallelism, concepts of work, speedup, efficiency, overhead, strong and weak scaling (Amdahl's law, Gustafson's law), practical considerations using parallel reduction and parallel prefix.

UNIT II:

Multi-core programming: shared memory and shared address space, data and task parallelism, summary of available APIs (OpenMP, TBB). Practical examples using OpenMP and parallel merge sort, pointer jumping, parallel BFS and basic linear algebra.

UNIT III:

Distributed memory programming: Message Passing Interface, latency + bandwidth model, distributed hashing, sample sort, parallel BFS, and basic linear algebra (matrix-vector, maxtrix-matrix products) with relation to graph algorithms.

UNIT IV:

Higher-level programming models: stateless programming in Map/Reduce, Apache Spark and fault-tolerance via Resilient Distributed Datasets. Triangle counting, connected components, single source shortest path using Spark

UNIT V:

Many-core programming: SIMD parallelism and massively parallel GPGPU accelerators. Brief overview of available APIs (OpenACC, oneAPI, OpenCL, SYCL). Programing GPUs in NVIDIA CUDA: data movement and organization, 1D/2D stencils, parallel prefix, matrix-matrix product.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Explain the principles of parallel and distributed computing, including types of parallelism, Flynn's taxonomy, and performance laws (Amdahl's, Gustafson's).
2. Implement parallel algorithms using OpenMP, MPI, CUDA, and Spark to solve computational and data-intensive problems.
3. Analyze the scalability, efficiency, and overhead of parallel algorithms and systems.
4. Design and optimize parallel solutions for real-world data science tasks, ensuring correctness, performance, and code quality.

Text Books

1. Grama, A., Gupta, A., Karypis, G., & Kumar, V. (2003). Introduction to Parallel Computing (2nd ed.). Pearson Education. ISBN: 978-0201648652.

2. McCool, M., Reinders, J., & Robison, A. (2012). Structured Parallel Programming: Patterns for Efficient Computation. Morgan Kaufmann. ISBN: 978-0124159938.

3. Karau, H., Konwinski, A., Wendell, P., & Zaharia, M. (2020). Learning Spark: Lightning-Fast Big Data Analysis (2nd ed.). O'Reilly Media. ISBN: 978-1492050049.

Reference Books

1. Kirk, D. B., & Hwu, W. W. (2016). Programming Massively Parallel Processors: A Hands-on Approach (3rd ed.). Morgan Kaufmann. ISBN: 978-0128119860.

2. Quinn, M. J. (2003). Parallel Programming in C with MPI and OpenMP. McGraw-Hill Education. ISBN: 978-0072822564.d Edition, O'Reilly Media, 2021.

Syllabus for Semester VI, B. Tech. Computer Science and Engineering (Data Science)

Course Code: CDP6004-3 Course: Distributed and Parallel Computing Lab

L:0 Hrs, T:0 Hrs, P:2 Hrs Per Week Total Credits: 1

Course Objectives

1. To introduce the fundamental concepts of parallel and distributed computing, including types of parallelism, performance metrics, and architectural models.
2. To equip students with practical programming skills using parallel computing APIs (OpenMP, MPI, CUDA, Spark) for solving data- and compute-intensive problems.
3. To develop the ability to analyze, design, and optimize algorithms for multi-core, many-core, and distributed memory architectures using appropriate models and tools.

Syllabus

Experiments based on the above syllabus of CDT6004

Text Books

1. Grama, A., Gupta, A., Karypis, G., & Kumar, V. (2003). Introduction to Parallel Computing (2nd ed.). Pearson Education. ISBN: 978-0201648652.

2. McCool, M., Reinders, J., & Robison, A. (2012). Structured Parallel Programming: Patterns for Efficient Computation. Morgan Kaufmann. ISBN: 978-0124159938.

3. Karau, H., Konwinski, A., Wendell, P., & Zaharia, M. (2020). Learning Spark: Lightning-Fast Big Data Analysis (2nd ed.). O'Reilly Media. ISBN: 978-1492050049.

Reference Books

1. Kirk, D. B., & Hwu, W. W. (2016). Programming Massively Parallel Processors: A Hands-on Approach (3rd ed.). Morgan Kaufmann. ISBN: 978-0128119860.

2. Quinn, M. J. (2003). Parallel Programming in C with MPI and OpenMP. McGraw-Hill Education. ISBN: 978-0072822564.d Edition, O'Reilly Media, 2021.

Syllabus for Semester VI, B. Tech. Computer Science and Engineering (Data Science)

Course Code: CDT6005-1 **Course**: **Data Science for Health Care**

L:0 Hrs, T:0 Hrs, P:2 Hrs Per Week **Total Credits: 1**

Course Objectives

1. Introduce healthcare data sources (EHR, medical imaging, biomedical signals, and genomics) and their challenges.
2. Learn machine learning and deep learning techniques for healthcare applications.
3. Develop skills in medical image analysis, signal processing, genomic data analysis, and clinical NLP and address bias, fairness, and ethical considerations in AI-driven healthcare.

Syllabus

Unit I:

Fundamentals of AI in Healthcare & EHR Analytics :Overview of AI in Healthcare, Introduction to Machine Learning & Deep Learning in Healthcare, AI Applications: Disease Diagnosis, Prognosis, Decision Support Systems, Electronic Health Records (EHR) Analytics: Components, Benefits, Barriers to Adoption, Coding Systems & Phenotyping Algorithms.

Unit II:

Image Analysis in Healthcare:Biomedical Imaging Modalities (X-ray, CT, MRI, Ultrasound), Image Processing Techniques: Object Detection, Image Segmentation, Image Registration, Feature Extraction for Disease Classification.

Unit III:

Biomedical Signal Analysis**:**Types of Biomedical Signals (ECG, EEG, EMG), ECG Signal Analysis: Signal Processing: Denoising, Feature Extraction, Multivariate Biomedical Signal Analysis, Cross-Correlation Analysis

Unit IV:

Genomic Data Analysis:Genomic Data Generation & Quality Control, Analysis Techniques : Normalization, Differential Expression Detection, Clustering & Classification in Genomics, Genome Sequencing Analysis & Personalized Medicine, Public Tools for Genomic Data Analysis (e.g., NCBI, Ensembl)

Unit V:

NLP for Clinical Text & Ethical Considerations in AI:Clinical Text Mining & Report Analysis, Core NLP Components in Healthcare, Challenges in Processing Clinical Reports, Applications of NLP in Healthcare (ICD Coding, Chatbots, Summarization) , Bias & Ethics in Healthcare AI, Algorithmic Bias in Medical AI Models, Explainability & Interpretability in Healthcare AI

Course Outcomes:

On successful completion of the course, students will be able to:

1. Analyze and preprocess healthcare data from various sources for AI applications.
2. Develop machine learning models for disease prediction, risk assessment, and diagnostics.
3. Apply AI techniques to medical imaging, signal processing, genomics, and NLP.
4. Evaluate AI models in healthcare for interpretability, fairness, and ethical concerns.

Text Books and Reference Books

1. Chandan K. Reddy, Charu C. Aggarwal, “Healthcare Data Analytics”, CRC Press
2. Sergio Consoli, Diego Reforgiato Recupero, Milan Petković; Data Science for Healthcare: Methodologies and Applications; Springer, 2019.
3. Eric Topol MD, Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again

Syllabus for Semester VI, B. Tech. Computer Science and Engineering (Data Science)

Course Code: CDT6005-2 **Course**: **Data Science for Genomics**

L:3 Hrs, T:0 Hrs, P:0 Hrs Per Week **Total Credits: 3**

Course Objectives

1. Develop a foundational understanding of genomic data science, including bioinformatics tools, machine learning, and deep learning techniques for analyzing genomic datasets.
2. Explore the integration of computational methods with genomics to enable applications in functional genomics, personalized medicine, and genome-wide association studies (GWAS).

Syllabus

Unit I:

Introduction to Genomics & Bioinformatics:Introduction to DNA, RNA, and Proteins, Genome Organization and Functional Genomics, Next-Generation Sequencing (NGS) Technologies, Bioinformatics Databases: NCBI, Ensembl, UCSC Genome Browser, KEGG, Genome Annotation and Comparative Genomic

Unit II:

Genomic Data Processing & Analysis:Data Formats in Bioinformatics: FASTA, FASTQ, VCF, BAM, GFF, Sequence Alignment Algorithms: BLAST, Needleman-Wunsch, Smith-Waterman, Genomic Variant Calling: GATK, SAMtools, BCFtools, Quality Control & Preprocessing of NGS Data (FASTQC, Trimmomatic), Cloud Computing for Genomics (Google Genomics, AWS, Terra.bio)

Unit III:

Statistical & Machine Learning Approaches in Genomics:

Statistical Foundations: Bayesian Inference, Maximum Likelihood Estimation (MLE), Dimensionality Reduction: PCA, t-SNE, UMAP for single-cell RNA sequencing (scRNA-seq),

Supervised Learning: Gene Expression Classification (SVM, Random Forest), Unsupervised Learning: Clustering of genes (K-Means, Hierarchical Clustering), Feature Selection & Biomarker Discovery in Genomics

Unit IV:

Deep Learning for Genomic Data:

Introduction to Deep Learning: ANN, CNN, RNN, CNNs for DNA Sequence Analysis & Motif Detection, Transformer Models in Genomics (e.g., DNABERT), Autoencoders & Variational Autoencoders (VAE) for DNA Data Augmentation, Hands-on Implementation using TensorFlow/PyTorch

Unit V:

Genome-Wide Association Studies (GWAS) & Variant Analysis:

Understanding GWAS: Identifying SNPs associated with traits, Population Genetics & Linkage Disequilibrium (LD), Statistical Methods in GWAS: Logistic Regression, Chi-square test, Disease Risk Prediction & Polygenic Risk Scores (PRS), Case Study: GWAS in Cancer Research & Personalized Medicine.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Demonstrate the ability to process, analyze, and interpret genomic sequences using computational techniques.
2. Apply machine learning and deep learning models for tasks such as gene expression analysis, mutation detection, and disease risk prediction.
3. Develop and implement high-throughput genomic data analysis using tools like GATK, SAMtools, BCFtools.
4. Perform statistical analysis and visualization of genomic data, including dimensionality reduction, clustering, and GWAS studies.

Text Books

1. Bioinformatics and Functional Genomics (3rd Edition), Jonathan Pevsner, Cambridge University Press, 2015.
2. Deep Learning for the Life Sciences, Bharath Ramsundar, Peter Eastman, Patrick Walters, Vijay Pande, O'Reilly Media, 2019.

Reference Books

1. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids, Richard Durbin, Sean R. Eddy, Anders Krogh, Graeme Mitchison, Cambridge University Press, 1998.
2. Bioinformatics Algorithms: An Active Learning Approach (2nd Edition, Vol. I), Phillip Compeau, Pavel Pevzner, Active Learning Publishers, 2015.

Syllabus for Semester VI, B. Tech. Computer Science and Engineering (Data Science)

Course Code: CDT6005-3 **Course**: **Data Science for Marketing**

L:3 Hrs, T:0 Hrs, P:0 Hrs Per Week **Total Credits: 3**

Course Objectives

1. Introduce marketing data sources (CRM, web analytics, social media, customer transactions) and their challenges.
2. Learn machine learning and deep learning techniques for marketing applications.
3. Develop skills in customer segmentation, demand forecasting, sentiment analysis, and marketing campaign optimization while addressing bias, fairness, and ethical considerations in AI-driven marketing.

Syllabus

Unit I: Fundamentals of AI in Marketing & CRM Analytics

Overview of AI in Marketing, Introduction to Machine Learning & Deep Learning in Marketing, AI Applications: Customer Insights, Personalization, Recommendation Systems, Customer Relationship Management (CRM) Analytics: Components, Benefits, Challenges, Data Collection & Preprocessing in Marketing: Customer Behavior Data, Clickstream Data, Transactional Data

Unit II: Customer Segmentation & Sentiment Analysis

Customer Lifetime Value (CLV) and Churn Prediction, Market Segmentation Techniques: RFM Analysis, K-Means Clustering, Text Analytics for Sentiment Analysis, Social Media & Review Data Processing

Unit III: Demand Forecasting & Recommendation Systems

Regression Models for Sales Forecasting, Time Series Forecasting Methods (ARIMA, Prophet), AI-Powered Recommendation Systems: Collaborative vs. Content-Based Filtering, Personalization Strategies in Digital Marketing, Case Study: AI-Driven Pricing Optimization

Unit IV: Marketing Campaign Analytics & A/B Testing

Key Marketing KPIs (CAC, ROAS, Conversion Rate, CTR), A/B Testing & Multivariate Testing for Campaign Optimization, Attribution Models (First-Touch, Last-Touch, Multi-

Touch), Web and social media Analytics Tools (Google Analytics, Facebook Insights), Case Study: Measuring the ROI of AI-Driven Marketing

Unit V: NLP in Marketing & Ethical Considerations in AI

NLP for Customer Reviews & Chatbots**,** Text Summarization & Topic Modeling in Marketing**,** AI Bias in Marketing: Fairness in Ad Targeting & Personalization**,** Ethical Issues in AI-Driven Consumer Behavior Analytics**,** Explainability & Interpretability of AI Models in Marketing

Course Outcomes:

On successful completion of the course, students will be able to:

1. Analyze and preprocess marketing data from various sources for AI applications.
2. Develop machine learning models for customer segmentation, demand forecasting, and recommendation systems.
3. Apply AI techniques to campaign optimization, sentiment analysis, and marketing strategy development.
4. Evaluate AI models in marketing for interpretability, fairness, and ethical concerns.

Text Books and Reference Books

1. Foster Provost & Tom Fawcett, "Data Science for Business", O'Reilly Media
2. Eric Siegel, "Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die", Wiley
3. Stephen Sorger, "Marketing Analytics: Strategic Models and Metrics", Pearson

Syllabus for Semester VI, B. Tech. Computer Science and Engineering (Data Science)

Course Code: CDT6005-4 **Course: Artificial Intelligence on Cloud**

L:3 Hrs, T:0 Hrs, P:0 Hrs Per Week **Total Credits: 03**

Course Objectives

By the end of this course students should be able to:

- Understand what AI is, common AI workloads, and where AI applies in real-world scenarios
- Become familiar with basic cloud fundamentals and to appreciate how it supports AI workloads
- Understand core machine learning concepts, and how ML (and deep learning/generative AI) relate to cloud services
- Recognize and describe how cloud supports AI workloads: computer vision, natural language processing (NLP), generative AI, and related services
- Understand principles of Responsible AI — fairness, safety, privacy, transparency, accountability — and ethical considerations when building AI solutions

Syllabus:

Unit 1: AI Workloads, Responsible AI & Basics

What is Artificial Intelligence — definition, importance, use cases; Overview of common AI workloads: computer vision, NLP, document processing, generative AI. Introduction to Responsible AI — guiding principles: fairness, reliability/safety, privacy & security, inclusiveness, transparency, accountability. Basics of cloud computing & why AI on cloud; introduction — Concept of Subscription, Resource Group, Resource Deployment Models, Different methods of creating resources.

Unit 2: Machine Learning Fundamentals on the cloud

Core ML techniques: regression, classification, clustering; Basics of deep learning & awareness of modern architectures (e.g. when deep learning might be used). Understanding

datasets: features vs labels; training / validation (and test) datasets; what it means to train and evaluate a model. Overview of capabilities: automated ML (AutoML), data and compute services for data science / ML, model management & deployment features.

Unit 3: Computer Vision Workloads on the cloud

Overview of vision-based AI workloads: image classification, object detection, optical character recognition (OCR), facial detection / analysis. Mapping vision workloads to cloud services: e.g. AI Vision service for image analysis; AI Face (or Face detection/analysis service) for face detection/analysis. Use-case discussion: when to use vision APIs vs custom ML/Deep learning, strengths/limitations, sample scenarios (image tagging, OCR for scanned documents, face recognition, object detection).

Core NLP workloads: key-phrase extraction, entity recognition, sentiment analysis, language modeling. Speech and language related workloads: speech-to-text (speech recognition), text-to-speech (speech synthesis), translation, conversational AI / chatbots / question-answering or document processing / document intelligence. Cloud services for NLP / language / speech, Cognitive Services for language and speech tasks. Scenario-based mapping: selecting appropriate service depending on workload (e.g. sentiment analysis, translation, speech recognition, Q&A, document processing).

Unit 5: Generative AI Workloads & Responsible AI Practice

Understanding generative AI: what it is, what generative AI models enable (e.g. text generation, image generation, content synthesis, summarization, chatbots), typical workloads. Generative AI services and capabilities: what they offer, when to use them.

Responsible AI in generative AI context: applying fairness, reliability, privacy & security, transparency, accountability when you use powerful generative models — discussion of ethics, biases, safeguards. Decision-making scenario: given a problem (e.g. generate text, summarize documents, translate, answer questions), how to decide which cloud AI service

/ model / architecture to use, and what ethical/legal/AI-governance considerations to keep in mind.

Course Outcomes

Upon completing the course successfully, students will be able to:

1. Define and describe common AI workloads — e.g., computer vision, NLP, document processing, generative AI — and match them to problem scenarios.
2. Explain basic ML concepts (regression, classification, clustering, dataset features/labels, train/validate splits) and differences from deep learning.
3. Describe how cloud supports AI/ML: know relevant cloud services (for ML, vision, language, generative AI) and when to choose each.
4. Identify appropriate cloud AI services to implement solutions for vision, language, document processing or generative-AI tasks.
5. Understand and apply Responsible AI principles — fairness, reliability/safety, privacy & security, inclusiveness, transparency, accountability — when designing AI solutions.

Text Books

1. The Microsoft Learn path "AI-900T00: Introduction to AI in Azure" — official, authoritative and aligned to the exam. Microsoft Learn+1
2. Official documentation for Azure AI services: Azure Machine Learning, Azure AI Vision, Azure AI Language, Azure AI Speech, Azure OpenAI Service / generative AI services, etc. (on Microsoft Docs). Microsoft Learn+2testpreptraining.com+2
3. Official "Skills measured" document for AI-900 — outlines what topics/exam domains are covered. Microsoft Learn+1

Syllabus for Semester VI, B. Tech. Computer Science and Engineering (Data Science)

Course Code: CDP6007 **Course**: **MINI PROJECT**

L:4 Hrs, T:0 Hrs, P:0 Hrs Per Week **Total Credits: 2**

Course Objectives

The objective of the Mini Project is to let the students map and utilize the technical knowledge acquired in the previous semesters to solve a real-world problem through team effort.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Identify and finalize the problem statement by investigating various domains and society needs.
2. Perform requirement analysis and design methodology for solving the identified problem.
3. Apply programming techniques and modern tools for the development of the solution.
4. Apply ethical principles, project management skills and demonstrate the ability to work in teams for project development within the confines of a deadline.
5. Communicate technical information employing written reports and presentations.

Syllabus for Semester VII, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT7001 Course: Deep Learning – II

L: 3 Hrs, T: 0 Hrs, P: 0 Hrs Per Week **Total Credits: 03**

Course Objectives

- To provide a deep understanding of how complex joint probability distributions are represented using Bayesian Networks (Directed) and Markov Networks (Undirected).
- To understand rules of independence, such as d-separation and local independencies, and how to perform causal and evidential reasoning within these structures.
- To introduce stochastic neural architectures like Restricted Boltzmann Machines (RBMs) and explain how they function as both graphical models and unsupervised learning tools.
- To develop proficiency in training advanced generative architectures, including Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs), through sampling techniques like Gibbs and Contrastive Divergence.

Syllabus

Unit-I:

Directed Graphical Models Recap of Probability Theory, Joint Distributions, representations of joint distribution, graphical representation of joint distribution, reasoning in a Bayesian network, Causal Reasoning, Evidential Reasoning, Independencies encoded by a Bayesian network (Case 1: Node and its parents), Independencies encoded by a Bayesian network (Case 2: Node and its non- parents), Independencies encoded by a Bayesian network (Case 3: Node and its descendants)

Unit -II:

Markov Networks Markov Networks: Motivation, Factors in Markov Network, Local Independencies in a Markov Network, Using joint distributions for classification and sampling, concept of a latent variable.

Unit -III:

Restricted Boltzmann Machines Introduction to Restricted Boltzmann Machines, RBMs as Stochastic Neural Networks, Unsupervised Learning with RBMs, Computing the gradient of thelog likelihood, Motivation for Sampling.

Unit – IV:

Markov Chains Introduction to Markov Chains, Need of Markov Chains, setting up a Markov Chain for RBMs, Training RBMs using Gibbs Sampling, Training RBMs using Contrastive Divergence.

Unit -V:

Variational Autoencoders Revisiting Autoencoders, Variational Autoencoders: The Neural Network Perspective, Variational autoencoders: (The graphical model perspective), Neural Autoregressive

Density Estimator, Masked Autoencoder Density Estimator. Introduction to Generative Adversarial Networks, Applications of GAN.

Course Outcomes

On successful completion of the course, students will be able to:

1. Construct and decompose Bayesian and Markov Networks, identifying the specific conditional independencies encoded between nodes, parents, and descendants.
2. Apply unsupervised learning techniques to Restricted Boltzmann Machines (RBMs), including computing log-likelihood gradients and using RBMs for feature extraction.
3. Design and set up Markov Chains to train models using Gibbs Sampling and Contrastive Divergence, enabling efficient parameter estimation in latent variable models.

Synthesize modern generative models like Variational Autoencoders (VAEs) and GANs, evaluating their performance from both the neural network and graphical model perspectives.

Text Books

1. Ian Good fellow and Yoshua Bengio and Aaron Courville. Deep Learning. An MIT Press book. 2016.
2. Charu C. Aggarwal. Neural Networks and Deep Learning: A Textbook. Springer. 2019.
3. Christopher Bishop, Pattern Recognition and machine learning; Springer Verlag, 2006.

Reference Books

1. Deep Learning from Scratch: Building with Python from First Principles by Seth Weidman published by O`Reilley.
2. Grokking Deep Learning by Andrew W. Trask published by Manning Publications.

Syllabus for Semester VII, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDP7001 Course: Deep Learning – II Lab

L: 0 Hrs, T: 0 Hrs, P: 2 Hrs Per Week Credits: 01

Course Objectives

- To implement Bayesian and Markov Networks for performing causal and evidential inference.
- To develop and train Restricted Boltzmann Machines (RBMs) using unsupervised learning techniques.
- To apply Gibbs Sampling and Contrastive Divergence for parameter estimation in latent models.
- To build and deploy VAEs and GANs for data generation and density estimation tasks.

Syllabus

Practical based on syllabus of CDT7001

Course Outcomes

On successful completion of the course, students will be able to:

1. Perform automated reasoning and independence checks in Directed and Undirected graphical models.
2. Implement RBMs to extract features and compute log-likelihood gradients from raw data.
3. Utilize Markov Chains and sampling algorithms to train complex stochastic neural networks.
4. Create functional Variational Autoencoders and GANs to generate synthetic data samples

Text Books

1. Nikhil Buduma, Fundamentals of Deep Learning,O'reilly
2. François Chollet, Deep Learning with Python Dr. S Lovelyn Rose,Dr. L Ashok Kumar,Dr.D Karthika Renuka ,Deep Learning using Python,Willey Publication

Reference Books

1. Charu C. Aggarwal. Neural Networks and Deep Learning: A Textbook. Springer. 2019.
2. Ian Goodfellow and Yoshua Bengio and Aaron Courville. Deep Learning. An MIT Press book. 2016.
3. Sandro Skansi, Introduction to Deep Learning, Springer

Syllabus for Semester VII, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT7002 **Course:** Cloud Computing

L: 3 Hrs, T: 0 Hrs, P: 0 Hrs Per Week **Total Credits: 03**

Course Objectives

1. Understand virtualization and containerization technologies used in cloud environments.
2. Explain and apply cloud architecture principles and service models.
3. Design and configure secure cloud networking and infrastructure.
4. Use AWS and Azure services for deploying applications and managing data.
5. Develop and evaluate scalable cloud-based application solutions.

Syllabus

Unit I: Virtualization & Containerization

Traditional vs virtualized infrastructure, Hypervisors (Type 1 & Type 2), Types of virtualization: Server, Storage, OS-level, Limitations of virtualization, Containerization concepts, Containers vs Virtual Machines, Docker basics: images, containers, lifecycle, Introduction to container orchestration, Kubernetes basics (Pods, Deployments, Services), Industry use cases.

Unit II: Cloud Fundamentals & Architecture

Introduction to cloud computing, Cloud service models: IaaS, PaaS, SaaS, Deployment models: Public, Private, Hybrid, Multi-cloud, Cloud architecture principles: Scalability, High availability, Fault tolerance, Elasticity, Cloud infrastructure: Regions, Availability Zones, Edge locations, High availability within datacenter, Cloud migration, Shared responsibility model, Introduction to public cloud platforms (AWS, Microsoft Azure, GCP), Introduction to core services in public cloud platforms, Different methods of deploying services in AWS and Azure, Concept of subscription and resource groups in Microsoft Azure.

Unit III: Cloud Networking, Security & Costing

Cloud networking: Virtual Private Cloud (VPC) / Virtual Network, Subnets and CIDR, Internet gateway, NAT gateway, Cloud security: Identity and Access Management (IAM), Security groups and Network ACLs, Data security basics, Costing: Pay-as-you-go pricing, Total Cost of Ownership (TCO), Cost optimization basics.

Unit IV: Core Cloud Services (AWS & Azure)

Compute services: AWS EC2, Azure Virtual Machines, VM configuration (Windows/Linux), Remote access: SSH, RDP, Storage services: AWS S3, EBS, EFS, Azure Blob Storage, Database services: AWS RDS, DynamoDB, Azure Cosmos DB.

Unit V: Cloud Application Deployment & Modern Practices

Application deployment in cloud: Hosting web applications on VM, Platform services (Elastic Beanstalk, Azure App Services), Serverless computing: AWS Lambda, Azure Functions, Load balancing and auto scaling, Infrastructure as Code (CloudFormation, ARM Templates), Cloud application architecture: Monolithic vs Microservices (conceptual), Basic API-based applications, Case studies: Cloud architecture examples (e-commerce, media applications).

Course Outcomes

On successful completion of the course, students will be able to:

1. Explain virtualization and container technologies in cloud computing.
2. Analyse and design cloud architectures based on performance and availability needs.
3. Implement cloud networking and security configurations.
4. Deploy and manage applications using cloud platforms (AWS/Azure).
5. Evaluate cloud solutions based on scalability, cost, and efficiency.

Text Books

1. Thomas Erl, Ricardo Puttini, Zaigham Mahmood, Cloud Computing: Concepts, Technology & Architecture, Pearson, 1st Edition, 2013.
2. Thomas Erl, Eric Barceló Monroy, Cloud Computing: Concepts, Technology, Security and Architecture, Pearson, 2nd Edition, 2023.
3. Rajkumar Buyya et al., Mastering Cloud Computing, McGraw Hill, 1st Edition, 2013.
4. A. Srinivasan, J. Suresh, Cloud Computing: A Practical Approach, Pearson India, 1st Edition, 2014.

Reference Books

1. Naresh Kumar Sehgal, P.C.P. Bhatt, Cloud Computing: Concepts and Practices, Springer, 1st Edition, 2018.
2. Kai Hwang et al., Distributed and Cloud Computing, Elsevier, 1st Edition, 2012.
3. Kris Jamsa, Cloud Computing, Jones & Bartlett Learning, 1st Edition, 2012.
4. Kelsey Hightower et al., Kubernetes Up and Running, O'Reilly, 2nd Edition, 2022.

5. Nigel Poulton, Docker Deep Dive, Latest Edition, 2023.

Syllabus for Semester VII, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDP7002 **Course: Cloud Computing Lab**

L: 3 Hrs, T: 0 Hrs, P: 2 Hrs Per Week **Total Credits: 01**

Course Objectives

- Create and manage virtual machines and container-based environments.
- Configure cloud networking components (VPC, subnets, gateways).
- Deploy storage and database services on cloud platforms.
- Host and manage applications on AWS and Azure.
- Implement serverless and scalable cloud solutions.

Syllabus

Practical's based on syllabus of CDT7002

Course Outcomes

On successful completion of the course, students will be able to:

1. Implement and demonstrate virtual machines and container-based environments to understand resource utilization and isolation.
2. Configure and manage cloud infrastructure components, including compute instances, storage services, and resource provisioning.
3. Design and implement virtual networks in cloud environments, including subnets, IP addressing, and connectivity mechanisms.
4. Deploy and manage applications using cloud platforms (AWS/Azure) by utilizing compute, storage, and database services.
5. Develop and deploy scalable and serverless cloud-based solutions, and integrate multiple services to build practical applications.

Text Books

1. Thomas Erl, Ricardo Puttini, Zaigham Mahmood, Cloud Computing: Concepts, Technology & Architecture, Pearson, 1st Edition, 2013.
2. Thomas Erl, Eric Barceló Monroy, Cloud Computing: Concepts, Technology, Security and Architecture, Pearson, 2nd Edition, 2023.
3. Rajkumar Buyya et al., Mastering Cloud Computing, McGraw Hill, 1st Edition, 2013.
4. A. Srinivasan, J. Suresh, Cloud Computing: A Practical Approach, Pearson India, 1st Edition, 2014.

Reference Books

1. Naresh Kumar Sehgal, P.C.P. Bhatt, Cloud Computing: Concepts and Practices, Springer, 1st Edition, 2018.

2. Kai Hwang et al., Distributed and Cloud Computing, Elsevier, 1st Edition, 2012.
3. Kris Jamsa, Cloud Computing, Jones & Bartlett Learning, 1st Edition, 2012.
4. Kelsey Hightower et al., Kubernetes Up and Running, O'Reilly, 2nd Edition, 2022.
5. Nigel Poulton, Docker Deep Dive, Latest Edition, 2023.

Syllabus for Semester VII, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT7003-1 Course: Image Analysis and Computer Vision

L: 3 Hrs, T: 0 Hrs, P: 0 Hrs Per Week Total Credits: 03

Course Objectives

- To enable students to design and develop computer vision applications, incorporating algorithms for object detection, tracking, and classification.
- To build understanding of image analysis techniques, including color models, segmentation, and image subtraction methods.
- To strengthen foundational knowledge in image processing, covering image fundamentals and essential terminologies.
- To equip learners with image enhancement and preprocessing techniques for improving image quality and preparing data for analysis.

Syllabus

Unit I

Fundamental Steps in Image Processing, Image representation, Components of digital image processing systems, Sampling and Quantization, Relationship between pixels – neighborhoods, adjacency connectivity, regions, boundaries, and distance measures.

Unit II

Image Enhancement in the Spatial Domain: Introduction to Spatial and Frequency methods, Basic Gray Level Transformations, Histogram Equalization, Histogram Processing, Local Enhancement, Image Subtraction, Image Averaging, Basics of Spatial Filtering, Smoothing Spatial Filters, Sharpening Spatial Filters.

Unit III

Color models: RGB, HSV, Grayscale, Image Segmentation: segmentation fundamentals, edge detection (Sobel, Canny), thresholding techniques (global, adaptive), region-based segmentation.

Unit IV

Image Compression and Coding: Introduction to image compression, coding redundancy and interpixel redundancy, fidelity criteria, image compression models, error-free compression techniques, variable length coding, bit-plane coding, lossless predictive coding, lossy compression methods, and image compression standards such as JPEG and MPEG.

Unit V

Motion Analysis and Dynamic Scene Understanding: Introduction to motion analysis, Optical Flow, Lucas–Kanade and Horn–Schunck approaches, feature point tracking, motion parameter estimation, motion-based object detection, background subtraction, texture Segmentation.

Course Outcomes

On successful completion of the course, students will be able to:

1. Demonstrate the knowledge of image fundamentals, basic terminologies in the field of image processing.
2. Apply different image enhancement methods for improving image quality.
3. Understand various colour models, image segmentation and subtraction techniques.
4. Apply algorithms for object detection, tracking, and classification in computer vision systems.
5. Apply algorithms for object detection, tracking, and classification in computer vision systems.

Text Books

1. Digital Image Processing : Rafael Gonzalez and Richard Woods, 3Ed., Pearson Pub.
2. Computer Vision: Algorithms and Applications by R. Szeliski, Springer, 2011.
3. Computer Vision: A Modern Approach, by David A. Forsyth and Jean Ponce, 1st edition, Prentice Hall, 2002.
4. Robot Vision, by Berthold Klaus Paul Horn, 1st edition, MIT Press, 1986.
5. Introductory Techniques for 3D Computer Vision, by E. Trucco and A. Verri, Publisher: Prentice Hall, 1998.

Reference Books

1. Fundamentals of Digital Image Processing : A.K.Jain, Prentice Hall.
2. Image Processing Principles & Applications: Tinku Acharya & Ajoy K. Ray, Willey
3. D. H. Ballard, C. M. Brown. Computer Vision. Prentice-Hall, Englewood Cliffs, 1982.
4. Image Processing, Analysis, and Machine Vision. Sonka, Hlavac, and Boyle. Thomson.
5. E. R. Davies, Computer & Machine Vision, Fourth Edition, Academic Press, 2012
6. Simon J. D. Prince, Computer Vision: Models, Learning, and Inference, Cambridge University Press, 2012

Syllabus for Semester VII, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDP7003-1 Course: Image Analysis and Computer Vision Lab

L: 0 Hrs, T: 0 Hrs, P: 2 Hrs Per Week Total Credits: 01

Course Outcomes

On successful completion of the course, students will be able to:

1. Understand and implement fundamental image processing operations such as image enhancement, filtering, and transformation.
2. Apply spatial and frequency domain techniques to process and analyze digital images.
3. Develop and evaluate algorithms for edge detection, image segmentation, and morphological operations.
4. Utilize image processing libraries to implement motion tracking techniques.
5. Analyze the performance of image processing techniques for real-world applications such as object recognition and image compression.

Syllabus Practical's based on syllabus of CDT7003-1

Text Books

1. Digital Image Processing: Rafael Gonzalez and Richard Woods, 3Ed., Pearson Pub.
2. Computer Vision: Algorithms and Applications by R. Szeliski, Springer, 2011.
3. Computer Vision: A Modern Approach, by David A. Forsyth and Jean Ponce, 1st edition, Prentice Hall, 2002.
4. Robot Vision, by Berthold Klaus Paul Horn, 1st edition, MIT Press, 1986.
5. Introductory Techniques for 3D Computer Vision, by E. Trucco and A. Verri, Publisher: Prentice Hall, 1998.

Reference Books

1. Fundamentals of Digital Image Processing: A.K.Jain, Prentice Hall.
2. Image Processing Principles & Applications: Tinku Acharya & Ajoy K. Ray, Willey
3. D. H. Ballard, C. M. Brown. Computer Vision. Prentice-Hall, Englewood Cliffs, 1982.
4. Image Processing, Analysis, and Machine Vision. Sonka, Hlavac, and Boyle. Thomson.
5. E. R. Davies, Computer & Machine Vision, Fourth Edition, Academic Press, 2012
6. Simon J. D. Prince, Computer Vision: Models, Learning, and Inference, Cambridge University Press, 2012

Syllabus for Semester VII, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT7003-2 Course: Spatial Data Management

L: 3 Hrs, T: 0 Hrs, P: 0 Hrs Per Week Total Credits: 03

Course Objectives

- To understand the fundamentals of spatial data and geospatial systems.

- To explore spatial data models, structures, and indexing techniques.
- To learn storage, querying, and management of spatial databases.
- To analyse spatial data using advanced analytical and visualization techniques.
- To apply spatial data concepts in real-world applications such as GIS, urban planning, and environmental monitoring.

Syllabus

Unit-I: Introduction to Spatial Data and GIS

Basics of spatial data: Definition, types (vector, raster), Spatial vs non-spatial data, Components of Geographic Information Systems (GIS), Coordinate systems and map projections, Data acquisition methods: Remote sensing, GPS, surveys, Applications of spatial data (urban planning, agriculture, disaster management)

Unit -II: Spatial Data Models and Structures

Vector data model: Points, lines, polygons, Raster data model: Grid representation, Topological relationships: Adjacency, connectivity, containment, Spatial data structures: Quad-trees, R-trees, Data representation and storage formats (Shapefile, GeoJSON), Metadata standards and data quality issues

Unit -III: Spatial Databases and Query Processing

Introduction to spatial databases, Spatial extensions of DBMS (PostGIS, Oracle Spatial)
Spatial data types and operations, Spatial query languages (SQL with spatial extensions)
Query processing techniques for spatial queries, Spatial indexing (R-tree, KD-tree)

Unit – IV: Spatial Data Analysis and Mining

Spatial analysis concepts, Buffering, overlay, and proximity analysis, Network analysis (shortest path, routing), Spatial statistics and interpolation methods, Spatial data mining: clustering, pattern detection, Handling big spatial data and real-time spatial analytics

Unit -V: Visualization and Applications

Spatial data visualization techniques, Thematic mapping and cartographic principles, Web GIS and cloud-based spatial systems, Tools: QGIS, ArcGIS, Google Earth Engine, Case studies: Smart cities, environmental monitoring, logistics, healthcare, Ethical and privacy issues in spatial data

Course Outcomes

On successful completion of the course, students will be able to:

1. To understand spatial data concepts and GIS fundamentals.
2. To explore spatial data models and indexing techniques.
3. To design and query spatial databases.
4. To analyse and visualize spatial data using modern tools.
5. To apply spatial data techniques in real-world domains.

Text Books

1. Shashi Shekhar & Sanjay Chawla, Spatial Databases: A Tour, Prentice Hall, 1st Edition
2. Paul Bolstad, GIS Fundamentals: A First Text on Geographic Information Systems, Eider Press
3. Paul A. Longley, Michael F. Goodchild, David J. Maguire, David W. Rhind, Geographic Information Systems and Science, John Wiley & Sons

Reference Books

1. Michael N. DeMers, Fundamentals of Geographic Information Systems, Wiley, 4th Edition
2. Ian Heywood, Sarah Cornelius, Steve Carver, An Introduction to Geographical Information Systems, Pearson Education, 4th Edition
3. P.A. Burrough & Rachael A. McDonnell, Principles of Geographical Information Systems, Oxford University Press, 2nd Edition

Syllabus for Semester VII, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDP7003-1 **Course: Spatial Data Management Lab**

L: 0 Hrs, T: 0 Hrs, P: 2 Hrs Per Week **Total Credits: 01**

Course Outcomes

On successful completion of the course, students will be able to:

- Provide practical knowledge of spatial data handling and management techniques.
- Understand spatial data models and database concepts for geographic information systems.
- Develop skills in spatial data analysis, querying, and visualization using GIS tools.
- Enable students to design and implement basic spatial data applications for real-world scenarios.

Syllabus

Practical's based on **CDT7003-2.**

Course Outcomes

After successful completion of this lab, students will be able to:

1. Understand and apply fundamental concepts of spatial data and spatial database systems.
2. Create, manage, and query spatial datasets using GIS tools and spatial DBMS.
3. Perform spatial analysis operations such as buffering, overlay, and spatial joins.
4. Analyze, visualize, and develop solutions for real-world problems using spatial data techniques.

Text Books

1. Shashi Shekhar & Sanjay Chawla, Spatial Databases: A Tour, Prentice Hall, 1st Edition
2. Paul Bolstad, GIS Fundamentals: A First Text on Geographic Information Systems, Eider Press
3. Paul A. Longley, Michael F. Goodchild, David J. Maguire, David W. Rhind, Geographic Information Systems and Science, John Wiley & Sons

Reference Books

1. Michael N. DeMers, Fundamentals of Geographic Information Systems, Wiley, 4th Edition
2. Ian Heywood, Sarah Cornelius, Steve Carver, An Introduction to Geographical Information Systems, Pearson Education, 4th Edition
3. P.A. Burrough & Rachael A. McDonnell, Principles of Geographical Information Systems, Oxford University Press, 2nd Edition

Syllabus for Semester VII, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT7003-3 **Course: Information Retrieval**

L: 3 Hrs, T: 0 Hrs, P: 0 Hrs Per Week **Total Credits: 03**

Course Objectives

- Understand the fundamental concepts of Information Retrieval
- Understand various data structures and ranking models used in IR systems
- Demonstrate the evaluating techniques for IR system
- Understand various probabilistic models
- Describe how web crawler works and understand the analysis of link structure.

Syllabus

Unit I:

Introduction: Introduction to Information Retrieval, Comparison of Database Management System and IR system, Components of IR system.

Boolean Model: Inverted Index and the terms vocabulary and postings lists, Data structures used in Information retrieval, Dictionaries and tolerant retrieval.

Unit II:

Introduction to index-construction and index-compression**.** Ranking Model: Scoring, term weighting and the vector space model, Cosine similarity measures, tf- idf model, Computing scores in a complete search system.

Unit III:

Evaluation in unranked and ranked information retrieval, relevance feedback and query expansion.

Unit IV:

Probabilistic Model: Review of basic probability theory, the probability ranking principle, the binary independence model, BM25 ranking function. Language models.

Unit V:

Web Search: Introduction to Web search basics, Web crawling and indexes, Link analysis, Page rank computation, Concept of Hub and Authority, near-duplicates, and spam detection.

Latent semantic Indexing, clustering and neural based retrieval(embeddings)

Course Outcomes

On successful completion of the course, students will be able to:

1. Realize the data structures like Inverted Indices used in Information retrieval systems.
2. Apply the concept of ranking models and evaluation techniques to IR problems.
3. Demonstrate the use of the probabilistic and machine learning techniques to IR systems.
4. Apply the concept of link analysis to web-based systems.

Text Books

1. An Introduction to Information Retrieval: Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, 1st Edition, Cambridge University Press

Reference Books

1. Information Retrieval: Implementing and evaluating search engines: Stefan Buttcher, Charles L. A. Clarke, Gordon V. Cormack, 1st Edition, MIT Press.
2. Modern Information Retrieval: Ricardo Yates, Berthier Ribeiro-Neto, Addison Wesley

Syllabus for Semester VII, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT7004 **Course: Customer Management**

L: 2 Hrs, T: 0 Hrs, P: 0 Hrs Per Week **Total Credits: 02**

Course Objectives

- To make the students understand the organizational need, benefits and process of creating long-term value for individual customers.
- To disseminate knowledge regarding the concept of Salesforce and Salesforce technologies.
- To enable the students understand the technological and human issues relating to implementation of Customer Relationship Management in the organizations.

Syllabus

Unit-I:

Introduction to CRM and Salesforce : Definition and importance of CRM, Key CRM concepts, Benefits of CRM for businesses, Introduction to Salesforce, Salesforce's role in business processes, Salesforce Cloud offerings , Overview of Salesforce architecture, Multi-tenant cloud architecture, Salesforce Data Model (Objects, Records, Fields), Understanding Tabs, Apps, and Objects

Unit -II:

Salesforce Administration Basics : Understanding Salesforce Setup menu, Creating and managing users, Profiles, Roles, and Permission Sets, Organizing security settings (Organization-Wide Defaults, Sharing Rules), Data Validation Rules, Creating and customizing Objects, Object Relationships, Creating and managing Fields, Workflow Rules, Process Builder, and Flow.

Unit -III:

Introduction to Apex Programming: Apex basics (Syntax, Variables, Methods), Apex classes and triggers, Working with SOQL and SOSL (Salesforce Object Query Language, Handling exceptions in Apex, Apex Triggers, Writing Apex triggers to handle database events, Trigger context variables and best practices, Governor limits and optimization techniques.

Unit – IV:

Advanced Salesforce Development – Lightning Web Components (LWC) : Overview of Lightning Web Components (LWC), LWC architecture and lifecycle, Creating and deploying LWC components, Handling events in LWC, Working with Apex from LWC, LWC Integration with Salesforce Data, Displaying Salesforce data in LWC, Handling record pages and lightning layouts, Best practices for LWC development.

Unit -V:

Salesforce Integration and Deployment, Salesforce Reports: Overview of integration in Salesforce, Integration tools: REST API, SOAP API, and Bulk API, Salesforce Connect, Introduction to Mulesoft for Salesforce integration, Introduction to Salesforce DX, Source-driven development and version control, Continuous Integration and Continuous Delivery (CI/CD) in Salesforce., Introduction to reports, types of reports, report builder, formatting reports, dashboard introduction, dashboard generation, charts in dashboards, limitations of Salesforce reports.

Course Outcomes

On successful completion of the course, students will be able to:

1. Understand the fundamentals of Salesforce and its role in CRM.
2. Gain practical skills in configuring and managing Salesforce environments.
3. Apply the basics of Salesforce development using Apex to customize the platform.
4. Develop modern web applications on the Salesforce platform using Lightning Web Components.
5. Integrate Salesforce with external systems and deploy applications in a production environment.

Text Books

1. Jason Ouellette; Development with the Force.com Platform, Second Edn, Addison Wesley, 2011.
2. Mohith Shrivastava; Salesforce Lightning Application Development, 2018. 2. Mohith Shrivastava; Salesforce Lightning Application Development, 2018
3. Judith W. Kincaid , Customer Relationship Management Getting it Right, Pearson Education
4. Customer Centricity –Focus on right customer for strategic advantage, by Peter Fader, Wharton Digital Press, 2012

Reference Books

1. Learning Salesforce Development with Apex – Paul Battisson
2. Salesforce for Beginners – Sharif Shaalan

Syllabus for Semester VII, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDP7005 **Course: Major Project - I**

L: 0 Hrs, T: 0 Hrs, P: 8 Hrs Per Week **Total Credits: 04**

Course Objectives

The objective of the Major Project is to let the students map and utilize the theoretical knowledge and technical knowledge acquired in the previous semesters to solve a real-world problem through team effort

.

Course Outcomes

On successful completion of the course, the student will be able to:

1. Identify and finalize the problem statement by investigating various domains and society needs.
2. Perform requirement analysis and design methodology for solving the identified problem.
3. Apply programming techniques and modern tools for the development of the solution.
4. Apply ethical principles, project management skills and demonstrate the ability to work in teams for project development within the confines of a deadline.
5. Communicate technical information employing written reports and presentations.

Syllabus for Semester VIII, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT8001-1 **Course: Mining Massive Datasets**

L: 3 Hrs, T: 0 Hrs, P: 0 Hrs Per Week **Total Credits: 03**

Course Objectives

- To develop understanding of similarity search techniques, distance measures, and hashing methods for large-scale data analysis.
- To introduce algorithms and models for mining and analyzing high-speed data streams.
- To equip students with scalable clustering techniques and their applications in big data environments.
- To provide knowledge of recommender systems, link analysis, and social network graph analysis for real-world problem solving.

Syllabus

Unit-I: Similarity Search

Introduction to similarity measures; Jaccard similarity (sets, documents); Shingling of Documents- k-Shingles, Choosing the Shingle Size, Hashing Shingles, Shingles Built from Words; Min Hashing technique; Locality Sensitive hashing and Analysis of the Banding Technique, Distance Measures- Euclidean Distances, Jaccard Distance, Cosine Distance, Edit Distance, Hamming Distance; Locality-Sensitive Families for Jaccard Distance; Applications of Locality-Sensitive Hashing- An Entity-Resolution Example, Fingerprint Matching**.**

Unit -II: Mining Data Streams

Stream data model and challenges; Sampling techniques- Reservoir sampling; Bloom filters;
Counting Distinct Elements in a Stream- Count-Distinct Problem, Flajolet-Martin Algorithm; Estimating Moments- Alon-Matias-Szegedy Algorithm for Second Moments; Sliding window models- Fixed window, Datar-Gionis-Indyk-Motwani Algorithm; Decaying Windows; Estimating frequencies in streams.

Unit -III: Clustering at Scale

Hierarchical Clustering in a Euclidean Space, Hierarchical Clustering in Non-Euclidean Spaces, CURE Algorithm, Representing Clusters in RGPF Algorithm, Stream-Clustering Algorithm.

Unit – IV: Recommender Systems and Link Analysis

Recommender Systems: long tail problem, content-based recommendations, Collaborative filtering Item based and user based, Matrix factorization.
Link Analysis: Web as a graph; PageRank algorithm, Modified Page Rank with personalization, Weighted page rank; HITS algorithm (Hub & Authority).

Unit -V: Social Network Graph Analysis

Degree centrality, Closeness centrality, Betweenness centrality, Clustering of Social-Network Graphs, Girvan-Newman Algorithm, Betweenness to Find Communities; Partitioning of Graphs; Random Walkers on a Social Graph; Random Walks with Restart

Course Outcomes

On successful completion of the course, students will be able to:

1. Apply similarity measures, shingling, MinHashing, and Locality Sensitive Hashing for efficient data processing.
2. Analyze and implement data stream mining algorithms such as sampling, Bloom filters, and frequency estimation techniques.
3. Design and evaluate clustering methods suitable for large-scale and streaming datasets.
4. Develop recommender systems and perform link analysis and social network analysis using graph-based techniques.

Text Books

1. Leskovec, Rajaraman, Ullman, Mining of Massive Datasets, Cambridge University Press.

Reference Books

1. Jiawei Han et al. (2011). Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann.
2. Christopher D. Manning et al. (2008). Introduction to Information Retrieval. Cambridge University Press.
3. Francesco Ricci et al. (2015). Recommender Systems Handbook (2nd ed.). Springer.

Syllabus for Semester VIII, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT8001-2 **Course: Generative AI**

L: 3 Hrs, T: 0 Hrs, P: 0 Hrs Per Week **Total Credits: 03**

Course Objectives

1. To understand the fundamental concepts, architectures, and probabilistic foundations of generative AI models.
2. To explore deep learning techniques such as autoencoders, GANs, and transformer-based models for generative tasks.
3. To develop skills in prompt engineering, large language models, and fine-tuning techniques for real-world applications.
4. To analyze evaluation metrics, optimization strategies, ethical concerns, and emerging trends in generative AI.

Syllabus

Unit I: Fundamentals of Generative AI

Introduction to Generative AI, History and evolution of AI, Generative vs Discriminative Models, Probabilistic modeling concepts, Types of Generative Models (explicit and implicit density models), Data distributions and latent space representation, Applications overview (text, image, audio, video generation), Advantages and Limitations, Ethical Issues and Responsible AI, Bias in generative systems, AI governance and regulations, Societal impact of generative AI, Overview of popular tools and frameworks

Unit -II: Deep Learning Foundations for Generative AI

Neural Network Fundamentals, Activation functions (ReLU, Sigmoid, Tanh), Loss functions and optimization techniques, Backpropagation and gradient descent, Autoencoder, Sparse Autoencoders, Denoising Autoencoders, Variational Autoencoders (VAE), Latent variable modeling, Generative Adversarial Networks, Types of GANs (DCGAN, Conditional GAN, CycleGAN), Generator–Discriminator training dynamics, Mode collapse problem, Transformer Architecture basics, Self-attention mechanism, Positional encoding, Encoder–Decoder architecture.

Unit -III: Prompt Engineering and NLP

Introduction to Prompt Engineering, Importance of prompt design, Types of Prompts (Zero-shot, Few-shot, Chain-of-thought, Role-based prompting), Prompt templates and structured prompting, Prompt optimization strategies, Prompt tuning techniques, NLP Basics (Tokenization, Stemming, Lemmatization), Word embeddings (Word2Vec, GloVe), Contextual embeddings, Working of Large Language Models, Text generation strategies (greedy search, beam search, sampling methods), Handling hallucinations in LLMs

Unit – IV: Generative AI with Large Language Models

Introduction to LLMs, Architecture of GPT and BERT models, Pre-training objectives (masked language modeling, next token prediction), Fine-tuning strategies, Transfer learning in LLMs, Reinforcement Learning from Human Feedback (RLHF), Parameter Efficient Fine-tuning (LoRA, adapters), Model quantization techniques, API Integration, Deployment pipelines, Scaling laws in LLMs, Distributed training basics, Memory and computational considerations

Unit -V: Evaluation, Optimization, and Research Trends in Generative AI

Evaluation Metrics for text (BLEU, ROUGE, METEOR, Perplexity), Evaluation Metrics for images (FID, IS Score), Human evaluation techniques, Model Optimization Techniques (Hyperparameter tuning, Grid search, Random search), Regularization techniques, Model compression (Pruning, Quantization, Knowledge distillation), Bias detection and fairness metrics, Explainable AI (XAI) techniques, Robustness testing, Adversarial attacks and defense mechanisms, Privacy concerns in generative AI, Emerging Trends (Multimodal models, Diffusion models, Foundation models, Autonomous AI agents), Future research directions

Course Outcomes

On successful completion of the course, students will be able to:

1. Explain the fundamental concepts of generative AI, including probabilistic modeling, generative vs discriminative approaches, and ethical considerations.
2. Apply deep learning techniques such as autoencoders, GANs, and transformer architectures to build basic generative models.
3. Design and optimize prompts and leverage large language models for text generation and problem-solving tasks.
4. Evaluate, optimize, and critically analyze generative AI models using performance metrics, fairness measures, and emerging research techniques.

Text Books

1. “Generative AI for everyone: Understanding the essentials and applications of this breakthrough technology". Altaf Rehmani.
2. "Introduction to Generative AI” , Numa Dhamani, Kindle Edition, 2024.
3. "Neural Networks and Deep Learning: A Textbook" by Charu C. Aggarwal.

Reference Books

1. “Generative Adversarial Networks Cookbook: Over 100 recipes to build generative models using Python, TensorFlow, and Keras" by Josh Kalin.

2. “Generative AI in Software Development: Beyond the Limitations of Traditional Coding” Jesse Sprinter, 2024

Syllabus for Semester VIII, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT8001-3 Course: Information Security and Data Privacy

L: 3 Hrs, T: 0 Hrs, P: 0 Hrs Per Week Total Credits: 03

Course Objectives

1. To build strong fundamental of Information Security and algorithm to realize goals of Security.
2. Understand authentication, intrusion detection and prevention and privacy issues.
3. Identify and mitigate software security vulnerabilities in existing systems.

Syllabus

Unit I: Fundamentals of Information Security

Security Goals, Types of Network Attacks, Threats, Vulnerabilities, Attacks, Data Integrity, Confidentiality, Anonymity, Message and Entity Authentication, Authorization, Non-repudiation, Network Security Model, Classical Encryption Techniques.

Unit -II: Symmetric Key Cryptography

Introduction to Stream Cipher and Block Cipher, Symmetric Key Block Cipher Algorithms (DES), Strength of DES, Block Cipher Modes of Operation.

Unit -III: Asymmetric Cryptography

Basic Mathematics for Asymmetric Key Algorithms, Asymmetric Key Algorithms, Related Attacks, Applications of Public Key Cryptography

Unit – IV: Authentication and Security Mechanisms

Hash Functions and their Security, Message Authentication Codes (MAC), Authentication Protocols, Passwords, Biometrics, Message Digest Algorithms, Digital Signatures, Kerberos.

Unit -V: Data Privacy and Protection Techniques

Introduction to Data Privacy, Definitions, Statistics, Privacy Policies, Privacy in Domains (Medical, Financial, etc.), Problems in Anonymous Data Publishing, Disclosure Risk, Suppression and Generalization, Anonymization Techniques, k-Anonymity, Overview of Differential Privacy and Protection Techniques.

Course Outcomes

On successful completion of the course, students will be able to:

1. Identify and investigate network security threat.
2. Apply encryption techniques to build security related applications.
3. Solve the problems related to key generation and key exchange algorithms.
4. Analyse authentication mechanisms and evaluate privacy related attacks for real life application
5. Apply privacy related algorithms for risk assessment.

Text Books

1. William Stallings; Cryptography & Networks Security Principles and Practice; 6th Edition Pearson Education, 2013.
2. Behrouz A. Forouzan, Cryptography and network security MC Graw Hill 3rd Edition
3. L. Sweeney, Computational Disclosure Control: A Primer on Data Privacy Protection, MIT Computer Science, 2002.
4. The Algorithmic Foundations of Differential Privacy, by Cynthia Dwork and Aaron Roth
5. B. Raghunathan, The Complete Book of Data Anonymization: From Planning to Implementation, Auerbach Pub, 2013.

Reference Books

1. Modern Cryptography: Theory and Practice, by Wenbo Mao, Prentice Hall PTR
2. Network Security Essentials: Applications and Standards, by William Stallings. Prentice Hall
3. Cryptography: Theory and Practice by Douglas R. Stinson, CRC press.

Syllabus for Semester VIII, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT8002-1 Course: Human Computer Interaction

L: 3 Hrs, T: 0 Hrs, P: 0 Hrs Per Week Total Credits: 03

----------Course Objectives

- To introduce students to understand systems that interact effectively with humans -
- unimodal as well as multi-modal.
- Enable students to implement, evaluate and provide long lasting solutions to support such HCI systems

Syllabus

Unit I: Introduction to Modern User Experience (UX) and Interfaces

Introduction to User Interface (UI) and User Experience (UX), importance of user-centered design, benefits of good design in digital products. Evolution from traditional interfaces to modern touch, voice, and immersive interfaces. Graphical User Interfaces (GUI), Natural User Interfaces (NUI), and Conversational Interfaces. Web and Mobile UI trends, principles of modern UI/UX design, accessibility and inclusive design fundamentals.

Unit II: Human-Centered Design and Interactive Technologies

Human cognition, perception, and psychology in interactive system design. Human interaction speeds and behavior in digital environments. Understanding user engagement, emotion, and experience design. Interaction devices and technologies: touch, gesture, voice, sensors, wearable devices, and IoT. Introduction to immersive technologies including Augmented Reality (AR), Virtual Reality (VR), and Mixed Reality (MR). Interaction paradigms: 2D, 3D, and spatial interfaces.

Unit III: Multimedia Interface Design and Prototyping

Design principles for multimedia interfaces including text, audio, video, animation, and interactive graphics. Screen design for mobile, web, and immersive environments. Information architecture, navigation design, and user flow. Prototyping tools and techniques (low-fidelity and high-fidelity). Iterative design process, usability heuristics, and design systems. Introduction to UX tools (Figma, Adobe XD) and rapid prototyping for AR/VR environments.

Unit IV: Immersive Systems and Universal Design

Design and implementation of AR/VR systems, including spatial UI, 3D interaction, and immersive storytelling. Multi-modal interaction (voice, gesture, gaze, haptics). Design of virtual environments and user immersion principles. Universal design for diverse users including accessibility in immersive systems. Ethical considerations in immersive technologies and user data privacy. Cross-platform design for mobile, web, and XR applications.

Unit V: Evaluation, Analytics, and Future Interfaces

Evaluation techniques for modern interfaces: usability testing, A/B testing, heuristic evaluation, and analytics-driven design. User research methods: surveys, interviews, eye-tracking, and behavioral analytics. User support systems: chatbots, intelligent assistants, and adaptive interfaces. Introduction to ubiquitous computing and smart environments. Future trends: AI-driven UX, immersive analytics, metaverse interfaces, and human-AI collaboration.

Course Outcomes

On successful completion of the course, students will be able to:

1. Understand basic concepts of user interface design for web, mobile, and AR/VR systems.
2. Identify different interaction devices and technologies used in human-computer interaction.
3. Explain the principles of universal and user-friendly design for diverse users.
4. Describe basic methods for evaluating HCI systems and providing user support.

Text Books

1. Wilbert O Galitz; The essential guide to user interface design; 2nd Edition; Wiley
2. DreamTech, 2002.
3. Ben Shneidermann; Designing the user interface; 3rd Edition; Pearson Education, 2009.
4. Alan Dix, et.al; Human - Computer Interaction; 3rd Edition; Pearson Education, 2003.

Reference Books

1. Prece, Rogers and Sharps; Interaction Design; 3rd Edition; Wiley Dreamtech, 2011.
2. Soren Lauesen; User Interface Design; Pearson Education, 2005

Syllabus for Semester VIII, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT8002-2 **Course: Optimization for Data Science**

L: 3 Hrs, T: 0 Hrs, P: 0 Hrs Per Week **Total Credits: 03**

Course Objectives

- Understand optimization techniques used in data science and machine learning.
- Apply mathematical optimization methods to real-world data problems.
- Analyze convergence and performance of optimization algorithms.
- Implement optimization techniques in Python for model training and tuning.

Syllabus

Unit I: Fundamentals of Optimization

Introduction to Optimization in Data Science, Types of Optimization Problems (Linear, Non-linear, Convex, Non-convex), Mathematical Preliminaries: Vectors, Matrices, Norms, Convex Sets and Convex Functions, Local vs Global Optima, Applications in Data Science

Unit II: Linear and Constrained Optimization

Linear Programming (LP) formulation, Graphical Method and Simplex Method, Duality Theory, Constrained Optimization Problems, Lagrange Multipliers, Karush-Kuhn-Tucker (KKT) Conditions

Unit III: Gradient-Based Optimization Methods

Gradient Descent (Batch, Stochastic, Mini-batch), Convergence Analysis, Learning Rate and Step Size Selection, Advanced Gradient Methods: Momentum ,Nesterov Accelerated Gradient, AdaGrad, RMSProp, Adam, Applications in Machine Learning (Regression, Neural Networks)

Unit IV: Advanced Optimization Techniques

Second Order Methods (Newton's Method, Quasi-Newton), Hessian Matrix and its role, Coordinate Descent, Proximal Gradient Methods, Regularization Techniques (L1, L2, Elastic Net), Optimization for Sparse Models.

Unit V: Optimization in Modern Data Science

Optimization in Deep Learning, Hyperparameter Optimization (Grid Search, Random Search, Bayesian Optimization), Distributed Optimization (Parallel and Large-scale systems), Convex

Optimization Libraries (CVXOPT, SciPy), Case Studies: Logistic Regression Optimization, SVM Optimization, Neural Network Training.

Course Outcomes

On successful completion of the course, students will be able to:

1. Understand and classify different optimization problems, including linear, nonlinear, convex, and non-convex formulations, along with their applications in data science.
2. Apply linear and constrained optimization techniques such as linear programming, Lagrange multipliers, and KKT conditions to solve real-world problems.
3. Analyze and implement gradient-based optimization algorithms and their variants for training machine learning models.
4. Evaluate advanced optimization methods, including second-order techniques and regularization approaches, for improving model performance.
5. Design and apply optimization strategies for modern data science applications, including deep learning and hyperparameter tuning..

Text Books

1. Boyd, Stephen, and Lieven Vandenberghe. Convex optimization. Cambridge university press, 2004.
2. Yurii, Nesterov, Introductory lectures on convex optimization: a basic course. Kluwer Academic Publishers, 2004.

Reference Books

1. Luenberger, D. G., and Y. Ye. Linear and nonlinear programming, Springer New York, 2008.
2. Nocedal, Jorge, and Stephen Wright. Numerical optimization. Springer Science & Business Media, 2006.

Syllabus for Semester VIII, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDT8002-3 **Course: Reinforcement Learning**

L: 3 Hrs, T: 0 Hrs, P: 0 Hrs Per Week **Total Credits: 03**

Course Objectives

- To understand the principles of reinforcement learning and decision-making under uncertainty.
- To study bandit algorithms and policy-based learning approaches for optimization.
- To develop knowledge of Markov Decision Processes and Bellman equations for sequential decision problems.
- To implement dynamic programming, temporal difference learning, and advanced RL techniques

Syllabus

Unit I: Introduction to Reinforcement Learning and Bandits

Introduction to Reinforcement Learning, Agent–Environment interaction, Elements of RL (state, action, reward, policy, value function), Exploration vs Exploitation dilemma, Multi-armed bandit problem, Regret minimization, Action-value methods, Incremental implementation, Upper Confidence Bound (UCB) algorithm, Optimism in the face of uncertainty, Probably Approximately Correct (PAC) learning framework, Sample complexity, Applications of bandit algorithms

Unit II: Advanced Bandit Algorithms and Policy Learning

Median Elimination algorithm, PAC guarantees in bandits, Exploration strategies comparison, Introduction to policy-based methods, Policy Gradient, Stochastic policies, Gradient ascent in policy space, REINFORCE algorithm, Variance reduction techniques, Comparison between value-based and policy-based methods, Limitations of bandit approaches

Unit III: Markov Decision Processes (MDPs)

Introduction to sequential decision making, Markov Decision Process, Markov property, State transition probabilities, Reward functions, Episodic vs continuing tasks, Discount factor, Policy evaluation, Policy improvement, Optimal policies, Value functions (state-value and action-value functions), Bellman expectation equations

Unit IV: Bellman Optimality and Dynamic Programming

Bellman optimality equations, Optimal value functions, Policy iteration, Value iteration, Convergence properties, Dynamic Programming methods, Model-based RL concepts, Limitations of DP, Introduction to Temporal Difference (TD) learning, TD(0), Monte Carlo vs TD methods, Bias-variance tradeoff in RL, On-policy vs off-policy learning (SARSA, Q-learning)

Unit V: Temporal Difference Learning and Eligibility Traces

Temporal Difference learning extensions, n-step TD methods, Forward and backward view of TD learning, Eligibility traces concept, TD(λ) algorithm, SARSA(λ), Watkins Q(λ), Trade-off between bias and variance using λ, Function approximation basics, Introduction to deep reinforcement learning, Stability and convergence issues, Practical challenges in RL implementation

Course Outcomes

On successful completion of the course, students will be able to:

1. Explain the principles of reinforcement learning, including agent–environment interaction, exploration strategies, and bandit algorithms.
2. Apply advanced bandit algorithms and policy-based methods such as policy gradients to solve decision-making problems.
3. Model sequential decision problems using Markov Decision Processes and compute value functions using Bellman equations.
4. Implement and evaluate reinforcement learning algorithms using dynamic programming, temporal difference learning, and eligibility traces.

Text Books

1. Richard S. Sutton and Andrew G. Barto, Reinforcement Learning: An Introduction, 2nd Edition, MIT Press, 2018.
2. Reinforcement Learning: An Introduction, Sutton and Barto, 2nd Edition.

Reference Books

1. Phil Winder, Reinforcement Learning: Industrial Applications of Intelligent Agents, O'Reilly Media, 2020.
2. Alexander Zai, Brandon Brown, Deep Reinforcement Learning in Action, 1st Edition, Manning Publications, 2020.

Syllabus for Semester VIII, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDP8003 **Course: Major Project - II**

L: 0 Hrs, T: 0 Hrs, P: 12 Hrs Per Week **Total Credits: 06**

Course Objectives

The objective of the Major Project is to let the students map and utilize the theoretical knowledge and technical knowledge acquired in the previous semesters to solve a real-world problem through team effort.

Course Outcomes

On successful completion of the course, the student will be able to:

1. Identify and finalize the problem statement by investigating various domains and society needs.
2. Perform requirement analysis and design methodology for solving the identified problem.
3. Apply programming techniques and modern tools for the development of the solution.
4. Apply ethical principles, project management skills and demonstrate the ability to work in teams for project development within the confines of a deadline.
5. Communicate technical information employing written reports and presentations.

SYLLABUS

B.Tech in Computer Science and Engineering (Data Science)

Scheme of Honors in Data Science

Sr. No.	Sem	Course Code	Name of Course	Hours per week		Credits	Maximum Marks			ESE Duration Hrs
				L	P		Continuous Evaluation	End Sem Exam	Total	
1	III	CDTH 3100	Data Science Essentials	3	0	3	100	-	100	-
2	IV	CDTH 4100	Software Architecture Analysis	3	0	3	100	-	100	-
3	V	CDTH 5100	Data Engineering	4	0	4	100	-	100	-
4	VI	CDTH 6100	Business and Web Analytics	4	0	4	100	-	100	-
5	VII	CDTH 7100	Project	0	8	4	50	50	100	-

Syllabus for Semester III, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDTH3100 **Course: Data Science Essentials**

L : 3 Hrs, T : 0 Hr, P : 0 Hr, Per Week **Total Credits: 3**

Course Objectives:

1. Develop proficiency in using Excel for data analysis and visualization, including advanced techniques like pivot tables, charts, and statistical functions.
2. Understand the fundamental concepts and techniques of data science, including data types, sources, cleaning, pre-processing, and exploratory analysis.
3. Apply data science skills to real-world problems and datasets through case studies and projects.
4. Learn how to utilize Plotly Dash for data modeling, transformation, visualization, and creating interactive dashboards and reports.

Syllabus:

Unit 1: Introduction to Data Science

Definition and overview of Data Science; Data Science workflow and processes; Applications of Data Science across various domains; Understanding different data types and sources; Ethics, privacy, and responsible data handling; Case studies and real-world examples of data science applications; Understanding the data science project lifecycle; Project ideation, planning, and management

Unit 2 : Data Analysis with Excel

Excel basics: workbooks, worksheets, formulas, and data formatting; Data cleaning and transformation techniques in Excel; Pivot tables for data summarization and analysis; Creating effective charts and graphs for data visualization; Statistical functions in Excel (e.g., AVERAGE, STDEV, CORREL); Introduction to Excel VBA and macros for automation

Unit 3: Advanced Data analysis with Excel

What-if analysis: Goal Seek & Scenarios, Introduction to Solver / optimization; Advanced formulas: nested and array functions; Advanced charts: scatter, histograms, map charts, Sparklines & trendlines, Dynamic dashboards with slicers & timelines, Power View / Pivot Charts in dashboards, Interactive controls : dropdowns, sliders, Geospatial visuals: map charts Dashboard planning & layout principles, Report templating & presentation-ready formatting, Exporting dashboards & sharing best practices

Unit 4: Data Pre-processing and Exploratory Analysis

Important Python libraries; Data cleaning and formatting techniques; Handling missing data and outliers; Feature engineering and selection; Descriptive statistics and summary measures; Correlation analysis and identifying relationships; Exploratory data analysis (EDA) techniques and best practices

Unit 5: Dash

Dash fundamentals; Callbacks and interactivity; Graphing and Visualization with Plotly; Web based dashboards; Real world applications and best practices.

Course Outcomes:

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

1) Explain core concepts and applications of data science, and describe the data science project lifecycle.
2) Apply data analysis, cleaning, and visualization techniques using Excel and Python, including advanced data processing, dashboard creation, and automation.
3) Design and construct interactive, web-based data dashboards using Dash

Recommended Resources:

- Excel Data Analysis for Dummies by Paul McFedries
- Power BI Desktop for Dummies by Brian Larson
- Python for Data Analysis by Wes McKinney
- Python Data Science Handbook by Jake VanderPlas
- The Book of Dash: Build Interactive Data Analysis and Visualization Apps with Python and Plotly by Adam Schroeder, Christian Mayer, and Ann Marie Ward
- Interactive Dashboards and Data Apps with Plotly and Dash by Packt Publishing

Syllabus for Semester IV, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDTH4100 Course: Software Architecture Analysis L : 3 Hrs, T : 0 Hr, P : 0 Hr, Per Week Total Credits: 3

Course Objectives

1. To implement ReactJs programming in end-to-end application development.
2. To comprehend the fundamentals of web development using MERN Stack.
3. To establish students as an expert web developer using this full stack developer course.

SYLLABUS

Unit I: Key issues in design and architecture fundamentals (general concepts, context, process, principles), Architectural structure and viewpoints: architectural styles, patterns, and frameworks, architectural trade-offs among various attributes,

Unit II: Hardware and systems engineering issues in software architecture, requirements traceability in architecture, service-oriented architectures, architectures for network, mobile, and embedded systems, Architecture in the Cloud.

Unit III: User interface: General human-computer interaction design principles, use of modes and navigation, coding techniques and visual design, response time and feedback, design modalities, localization and internationalization, human-computer interaction design methods, interface modalities, metaphors and conceptual models, HCI.

Unit IV: Detailed design for software: database design, design notations, Developing secure software, secure design principles and patterns

Unit V: Creating systems which can evolve, Documenting designs, Software design tools, Reinforce design issues and strategies

Unit VI: Design concepts, process and principles, concurrency, control and handling of events, error and exception handling, interaction and presentation, data structure centered design, aspect-oriented design

Text Books:

Software Architecture in Practice, 3/E, by Len Bass, Paul Clements, and Rick Kazman

Course Outcomes: On successful completion of the course Students will be able to:

1. Understand the need for software architecture and relationship to low-level Design
2. Develop architectural approaches from requirements and manage traceability between architecture and requirements
3. Analyse trade-offs among multiple architectural alternatives
4. Utilize quality attributes when designing software architectures
5. Recognize architectural patterns and apply them appropriately

Syllabus for Semester V, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDTH 5100 **Course: Data Engineering**

L : 4 Hrs, T : 0 Hr, P : 0 Hr, Per Week **Total Credits: 4**

Course Objectives

1. Understand the foundational concepts and significance of data engineering in data-driven enterprises.
2. Explore various data storage and processing technologies and their applications.
3. Gain hands-on skills in data ingestion, transformation, and storage using modern tools and Python.
4. Comprehend the design and orchestration of data pipelines for real-world data workflows.
5. Identify and apply best practices, governance principles, and tools in cloud-based data engineering environments.

Syllabus

Unit I

Introduction to Data Engineering: What is Data Engineering, Roles and responsibilities of a Data Engineer, Data Engineering vs. Data Science, Overview of the data lifecycle

Unit 2

Data Storage Technologies: Relational databases (SQL, PostgreSQL), NoSQL databases (MongoDB), Introduction to data lakes and data warehouses

Unit 3:

Data Processing: Introduction to batch and stream processing, Python for data processing, Basic data transformation and cleaning

Unit 4

Data Ingestion: Data sources (files, databases, APIs), Data ingestion tools (Python libraries, Apache Kafka), Introduction to data pipelines

Unit 5

Data Pipelines and Orchestration: Components of a data pipeline, Introduction to Apache Airflow, Scheduling and monitoring data pipelines

Unit 6

Data Engineering Tools and Practices: Introduction to cloud data platforms (AWS, GCP), Data governance and security basics, Best practices in data engineering

Course Outcomes:

On successful completion of the course, students will be able to:

1. Explain the fundamentals of data engineering and its importance.
2. Describe different data storage and processing technologies.
3. Perform basic data ingestion, transformation, and storage operations using Python.
4. Understand the concept of data pipelines and their orchestration.

5. Identify various tools and best practices used in data engineering.

Text Books

1. "Fundamentals of Data Engineering" by Joe Reis and Matt Housley, Publisher: O'Reilly Media
2. "Data Engineering with Python" by Paul Crickard, Publisher: Packt Publishing
3. "Data Engineering with Python" by Jesse Anderson, Publisher: Manning Publications

Reference Books

1. "Designing Data-Intensive Applications" by Martin Kleppmann, Publisher: O'Reilly Media
2. "Streaming Systems: The What, Where, When, and How of Large-Scale Data Processing" by Tyler Akidau, Slava Chernyak, and Reuven Lax, Publisher: O'Reilly Media
3. "The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling" by Ralph Kimball and Margy Ross, Publisher: Wiley
4. "Cloud Data Management" by Divyakant Agrawal, Amr El Abbadi, and Sudipto Das Publisher: Morgan & Claypool Publishers

Syllabus for Semester VI, B. Tech. Computer Science & Engineering (Data Science)

Course Code: CDTH 6100 **Course: Business & Web Analytics**

L : 4 Hrs, T : 0 Hr, P : 0 Hr, Per Week **Total Credits: 4**

Course Objectives

1. To familiarize students with the concepts of business and web analytics.
2. To enable students to deploy business and web intelligence to improve the outcomes of marketing or business plan.
3. Students will gain an understanding of the strategic and operational aspects of business and web analytics tools and technologies.

SYLLABUS

UNIT – I: Predictive Analytics : Trendlines and Regression analysis, Forecasting techniques

UNIT – II: Simulation and Risk Analysis : Monte Carlo simulation, Random Sampling from probability distributions, dynamic system simulation, spreadsheet modeling and analysis.

UNIT – III: Prescriptive Analytics : Linear Optimization, Interger and nonlinear optimization, optimization analytics.

UNIT – IV: Web Analytic fundamentals: Capturing data: Web logs or JavaScript's tags, Separate data serving and data capture, Type and size of data, Innovation, Integration, Selecting optimal web analytic tool, Understanding click stream data quality, Identifying unique page definition, Using cookies, Link coding issues.

UNIT – V: Web Metrics: Common metrics: Hits, Page views, Visits, Unique visitors, Unique page views, Bounce, Bounce rate, Page/visit, Average time on site, New visits; Optimization (e-commerce, non e-commerce sites): Improving bounce rates, Optimizing adwords campaigns; Real time report, Audience report, Traffic source report, Custom campaigns, Content report, Google analytics, Introduction to KPI, characteristics, Need for KPI, Perspective of KPI, Uses of KPI.
Relevant Technologies: Internet & TCP/IP, Client / Server Computing, HTTP (Hypertext Transfer Protocol), Server Log Files & Cookies, Web Bugs.

UNIT – VI: Web Analytics 2.0:Web analytics 1.0, Limitations of web analytics 1.0, Introduction to analytic 2.0, Competitive intelligence analysis : CI data sources, Toolbar data, Panel data ,ISP data, Search engine data, Hybrid data, Website traffic analysis: Comparing long term traffic trends, Analyzing competitive site overlap and opportunities.

Google Analytics: Brief introduction and working, Adwords, Benchmarking, Categories of traffic: Organic traffic, Paid traffic; Google website optimizer, Implementation technology, Limitations, Performance concerns, Privacy issues.

Course Outcomes:

On completion of the course the student will be able to

1. Apply predictive analytic techniques on real time data.
2. Perform simulation and risk analysis.
3. Explain and discuss web metrics.
4. Examine how different industries across the globe are using web analytics analytics.

Text Books:

1. James R. Evans, Business Analytics Methods, Models and Decisions, Pearson.
2. Clifton B., Advanced Web Metrics with Google Analytics, Wiley Publishing, Inc.2nd ed.
3. Kaushik A., Web Analytics 2.0, The Art of Online Accountability and Science of Customer Centricity, Wiley Publishing, Inc. 1st ed.
4. Sterne J., Web Metrics: Proven methods for measuring web site success, John Wiley and Sons

References:

1. Mathew Ganis, Avinash Koihrkar-Social Media Analytics-IBM Press
2. Jim Sterne, Social Media Metrics, Wiley

SYLLABUS

B.Tech in Computer Science and Engineering (Data Science)

Scheme of B. Tech. Minor Specialization in Computer Science Engineering (Data Science)

Sr. No.	Sem	Course Code	Name of Course	Hours per week			Credits	Maximum Marks			ESE Duration Hrs
				L	T	P		Continuous Evaluation	End Sem Exam	Total	
1	III	CDTM 3100	Foundation of Data Science	3	0	0	3	50	50	100	3
2	IV	CDTM 4100	Statistics for Data Science	3	0	0	3	50	50	100	3
3	V	CDTM 5100	Exploratory Data Analysis	4	0	0	4	50	50	100	3
4	VI	CDTM 6100	AI tools for Data Science	4	0	0	4	50	50	100	3
5	VII	CDPM 7100	Major Project	0	0	8	4	50	50	100	-
			TOTAL	**18**	**0**	**0**	**18**	**250**	**250**	**500**	

Syllabus for Semester III, Minor in Data Science

Course Code: CDTM3100-1 **Course:** Foundation of Data Science

L: 3Hrs**, T:** 0Hr**, P:** 0Hr**, Per Week** **Credits:** 3

Course Objectives

The objective of this course is to familiarize the prospective engineers with:

1. The fundamentals of data science and the basics of Python programming constructs.
2. Representation of complex data using python data structures - lists, tuples, dictionaries.
3. Manipulation of Numpy arrays and Pandas series and data frames
4. Data Wrangling techniques for Data Analysis

SYLLABUS

UNIT-I

Introduction

Introduction to Data Science, Benefits and uses, Facets of data, Evolution of Data Science – Data Science Roles – Data science process – Applications of Data Science in various fields – Data Security Issues.

UNIT-II

Python Basics

Basics of Python including data types, operators, variables, expressions, control structures using sample dataset, objects and functions, Class – Constructors – Object Creation – Inheritance – Overloading. Text Files and Binary Files – Reading and Writing, Decorators

UNIT-III

Data Structures

Python sequence data structures including – creation and manipulation String, Array, List, Tuple, Set, and Dictionary. Introduction to various python libraries for data science – Numpy, Pandas, seaborn, matplotlib, SciPy.

Unit – IV: Numpy and Pandas Packages

NumPy - NumPy ndarray:- Vectorization Operation - Array Indexing and Slicing - Transposing Array and Swapping Axes - Saving and Loading Array - Universal Functions - Mathematical and Statistical Functions in Numpy.

Series and DataFrame in pandas – Creating, manipulating and reading DataFrames, Panda's Index Objects - Reindexing Series and DataFrames - Dropping entries from Series and Data Frames -

Indexing, Selection and Filtering in Series and Data Frames - Arithmetic Operations between Data Frames and Series - Function Application and Mapping.

UNIT-V

Data Wrangling – Clean, Transfer, Merge and Reshape

Combining and Merging Data Sets – Reshaping and Pivoting – Data Transformation – String manipulations – Regular Expressions.

Unit – VI: Data Aggregation and Group Operations

GroupBy Mechanics – Data Aggregation – GroupWise Operations and Transformations – Pivot Tables and Cross Tabulations – Date and Time data types.

Course Outcomes

After successful completion of this course, the student will be able to,

1. Identify the need for data science and solve basic problems using Python conditionals, loops and functions.
2. Determine the methods to create and manipulate Python programs by utilizing the data structures like Array, lists, tuples, set and dictionaries.
3. Apply the efficient storage and data operations using NumPy arrays. and perform powerful data manipulations using Pandas
4. Interpret the data wrangling process including cleaning, transferring, merging and and reshape
5. Use the aggregations and group operations for data analysis in python.

Text Books

1. Davy Cielen, Arno D. B. Meysman, Mohamed Ali, “Introducing Data Science”, manning publications 2016.
2. Wes Mckinney. Python for Data Analysis. O'Reilly Media, 2013.
3. Grus, Joel. Data Science from Scratch: First Principles with Python. O'Reilly Media, 2015.

Reference books

1. Allen B. Downey, ``Think Python: How to Think Like a Computer Scientist’’, 2nd edition,Updated for Python 3, Shroff/O’Reilly Publishers, 2016
2. David J. Pine, Introduction to Python for Science and Engineering, CRC Press, 2019.
3. Jake vander Plas, Python Data Science Handbook – Essential Tools for Working with Data, O’Really Media, 2017

Syllabus for Semester IV, Minor in Data Science

Course Code: CDTM4100 **Course:** Statistics for Data Science

L: 3Hrs**, T:** 0Hr**, P:** 0Hr**, Per Week** **Credits:** 3

Course Objectives

The primary objective of this course is

- To provide students with a comprehensive understanding of statistical concepts, methods, and techniques essential for data analysis and decision-making processes.
- Students will learn how to collect, organize, analyze, and interpret data using various statistical tools and methodologies.

SYLLABUS

Unit 1:

Introduction, Variables and Types of data, Data collection and sampling techniques, uses and misuses of statistics, Organizing data, histograms, frequency polygons, ogives, other types of graphs.

Unit 2:

Measures of central tendency, measures of variation, measures of position, five-number summary, boxplots, etc.

Unit 3:

Probability distributions, normal distributions, binomial distribution, other types of distributions, central limit theorem.

Unit 4:

Confidence interval and sample size, Hypothesis testing.

Unit 5:

Testing the Difference Between Two Means, Two Proportions, and Two Variances, Correlation and Covariance.

Unit 6:

Simple linear regression, multiple linear regression, logistic regression.

Course Outcomes

On successful completion of the course, the students will be able to:

1. Apply sampling techniques to generate appropriate samples.
2. Apply descriptive statistical techniques for data analysis.
3. Interpret data to perform hypothesis testing.
4. Apply regression models.

Text Books:

1. Elementary Statistics A Step by Step Approach by Allan G. Bluman , McGraw Hill Publications, Seventh Edition.

2. Practical Statistics for Data Scientists by Peter Bruce and Andrew Bruce, O`Reilly Publication

Syllabus for Semester V, Minor in Data Science

Course Code: CDTM5100 **Course: Exploratory Data Analysis**

L : 4 Hrs, T : 0 Hr, P : 0 Hr, Per Week **Total Credits: 4**

Course Objectives

1. To introduces the methods for data preparation and data understanding.
2. To explore essential exploratory techniques for understanding multivariate data by summarizing it through statistical and graphical methods.
3. To summarize use of predictive analytics, data science and data visualization.

Syllabus

Unit I: Introduction to Exploratory Data Analysis
Introduction to Exploratory Data Analysis (EDA) –Steps in EDA, Data Types: Numerical Data – Discrete data, continuous data – Categorical data – Measurement Scales: Nominal, Ordinal, Interval, Ratio – Comparing EDA with classical and Bayesian Analysis – Software tools for EDA.

Unit II: Data Transformation

Transformation Techniques: Performing data deduplication - replacing values – Discretization and binning. Introduction to Missing data, handling missing data: Traditional methods - Maximum Likelihood Estimation.

Unit III: Descriptive statistics: mean, median, mode, range, variance, standard deviation Frequency distributions

Unit IV: Correlation Analysis and Time Series Analysis

Types of analysis: Univariate analysis - bivariate analysis - multivariate analysis. Time Series Analysis (TSA): Fundamentals of TSA - characteristics of TSA – Time based indexing - visualizing time series – grouping time series data - resampling time series data.

Unit V: Data Summarization and Visualization

Statistical summary measures, data elaboration, 1-D Statistical data analysis, 2-D Statistical data Analysis, contingency tables, n-D Statistical data analysis. Visualization: Scatter plots – Dot charts- Bar plots.

Unit VI: Feature Engineering & Dimensionality Reduction

Feature creation (binning, interaction terms, polynomial features, Principal Component Analysis (PCA) for dimensionality reduction, t-SNE for high-dimensional data visualization.

Course Outcomes:

On successful completion of the course, students will be able to:

CO1. Apply data cleaning and transformation techniques (handling missing values, outliers, discretization) to prepare datasets for analysis.

CO2. Compute descriptive statistics (mean, variance, distributions) and interpret univariate/bivariate relationships.

CO3. Analyze correlations and time-series patterns using appropriate statistical and visualization methods.

CO4. Design effective visualizations (scatter plots, bar charts) to summarize and communicate data insights.

CO5. Implement dimensionality reduction techniques (PCA, t-SNE) to optimize datasets for machine learning.

Text Books

1. Suresh Kumar Mukhiya, Usman Ahmed, "Hands-On Exploratory Data Analysis with Python" 1st Edition, 2020, Packt Publishing.
2. Martinez, W , Martinez A & J.L. Solka : Exploratory Data Analysis with MATLAB, CRC Press, A Chapman & Hall Book, 3rd Edition, 2017

Reference Books

1. Michael Jambu, "Exploratory and multivariate data analysis", 1991, 1st Edition, Academic Press Inc.
2. Charu C. Aggarwal, "Data Mining The Text book", 2015, Springer.
3. Craig K. Enders, "Applied Missing Data Analysis", 2010, 1st Edition, The Guilford Press.

Syllabus for Semester VI, Minor in Data Science

Course Code: CDTM6100 **Course: AI Tools for Data Science**

L : 4 Hrs, T : 0 Hr, P : 0 Hr, Per Week **Total Credits: 4**

Course Objectives

1. To introduce students with languages, tools, and data used by data scientists.
2. To introduce students about open source tools and their features
3. To enable students to Create and manage source code for data science in GitHub

Syllabus

Unit I: Data Science Fundamentals, Languages of Data Science, Introduction to Python, Introduction to R Language, Categories of Data Science Tools, Open source Tools for Data Science (GitHub, Jupyter Notebooks, and RStudio IDE.), Commercial Tools for Data Science, Cloud Based Tools for Data Science, Libraries for Data Science, Application Programming Interfaces (API), Data Sets - Powering Data Science, Sharing Enterprise Data - Data Asset eXchange, Machine Learning Models, The Model Asset Exchange

Unit II: Getting started with raw data, basic data structures in Python – lists, dictionary, tuples, set, the worlds of arrays with Numpy, creating an array, mathematical operation, indexing and slicing, the data structure of pandas series data frame, Data structures in R : vectors, lists, data frames, matrices, arrays, factors.

Unit III: Overview of Git, branching and merging, Git Workflow, Contributing to repositories via pull request.

Unit IV: Introduction to IBM Watson, Creating an Account on IBM Watson Studio, Jupyter Notebook in Watson Studio, Linking GitHub to Watson Studio, Other IBM Tools for Data Science

Unit V: IBM Watson Knowledge Catalog, Data Refinery, SPSS Modeler Flows in Watson Studio, IBM SPSS Modeler, SPSS Statistics, Model Deployment with Watson

Unit VI: Machine Learning, Auto AI in Watson Studio, IBM Watson OpenScale, R Markdown, Scientific Thinking, Experimental Design, Introduction to Big Data

Course Outcomes:

On successful completion of the course, students will be able to:
CO1. Use version control, markdown, git, GitHub. CO2. Write basic Python programs.
CO3. Write basic R programs CO4. Use IBM Watson Studio.

Text Books and Reference Books

1. **Python Programming Using** Problem Solving Approach: Reema Thareja, Oxford University Press; First edition
2. The Book of R - The Book of a First Course in R Programming and Statistics, Tilman
   M. Davies, No Starch Press.