



University of Madras

Chepauk, Chennai 600 020

[Est. 1857, State University, NAAC 'A' Grade, CGPA 3.32, NIRF 2019 Rank: 20]

Website: www.unom.ac.in, Tel. 044-2539 9000

Undergraduate Programme in Computer Science with Artificial Intelligence

Syllabus for B.Sc Computer Science with Artificial Intelligence (With effect from the Academic Year 2022 -23)

Learning Outcome Based Curriculum Framework

Note: The Committee is designed Learning Outcome Based Curriculum Framework of Undergraduate Computer Science Programmes prescribed by UGC

I Preamble

Bachelor of Computer Science with Artificial Intelligence is a 3 – Year Undergraduate Programme spread over six semesters. The course is designed to achieve a high degree of technical skills in Problem solving and Modern application development. The course develops requisite professional skills and problem solving along with developing the analytical abilities for pursuing a successful career in software industry and forms the required basics for further higher studies in Computer Science specifically in the area of Artificial Intelligence.

ELIGIBILITY FOR ADMISSION:

Candidates for admission to the first year of the Degree of Bachelor of Science in Computer Science with Artificial Intelligence Courses should have studied Mathematics at Higher Secondary level.

II Course Objectives

- Acquisition of Knowledge and understanding of systems, various programming languages and tools required for effective computation based problem solving and analysis.
- Utilize emerging technological tools learn, adapt and successfully engage effective procedural coding to meet the needs of technical and societal challenges
- Attain sufficient knowledge related to computer domains, render technical, soft and hard skills to apply them effectively in team work
- Empower the students with competencies in creative thinking and problem solving, interpersonal communication and managerial skills.

III Graduate Attributes

- Computational Knowledge
- Problem analysis & Solving
- Design & Development of Solutions
- Modern tool usage
- Communication skills
- Innovation & Entrepreneurship
- Societal & environmental concern
- Pragmatic Application of tools

IV Course Outcomes

After Completion of the course, the students are expected to

- Understand the basic principles and concepts of Computer Science and Artificial Intelligence with an integrate the knowledge gained in the domain with practical needs of the society and be an ethically and socially responsible Professional
- Explore latest emerging technologies in diverse areas of Computer Science and inculcate skills for successful career, entrepreneurship and higher studies
- Apply the concepts of Computer and practices via emerging technologies and Software development tools to solve pragmatic social concerns.

Course Structure

B.Sc Computer Science with Artificial Intelligence

SYLLABUS

2022 -23

ELIGIBILITY FOR ADMISSION:

Candidates for admission to the first year of the Degree of Bachelor of Science in Computer Science with Artificial Intelligence Courses should have studied Mathematics at Higher Secondary level.

S.NO.	PART	SUBJECT NAME	CREDITS	Lecture Hours	MAXIMUM MARKS			
					EXT. MKS	INT. MKS	TOTAL	
SEMESTER I								
1	I	Tamil/ Other languages – I	3	6	75	25	100	
2	II	BP2-ENG01- Communicative English I	3	3	75	25	100	
3	III	Core I : Introduction to Artificial Intelligence	4	6	75	25	100	
4	III	Practical I : PROLOG Lab	3	5	60	40	100	
5	III	BMA-CSA01-Allied I: Mathematics-I@	5	6	75	25	100	
6	IV	Basic Tamil/Advanced Tamil/NME I*	2	-	75	25	100	
7	IV	BP4-EPSC01- English for Physical Science I	3	4	50	50	100	
Total Credits			23					
SEMESTER II								
8	I	Tamil/ Other languages – II	3	6	75	25	100	
9	II	BP2-ENG02- Communicative English II	3	3	75	25	100	
10	III	Core II : Problem Solving Using Python@	4	6	75	25	100	
11	III	Practical II : Python Lab@	3	5	60	40	100	
12	III	BMA-CSA02-Allied II: Mathematics II@	5	6	75	25	100	
13	IV	Basic Tamil/Advanced Tamil/NME-II*	2	-	75	25	100	
14	IV	BP4-EPSC02- English for Physical Science II	3	4	50	50	100	
Total Credits			23					
SEMESTER III								
15	I	Tamil/ Other languages – III	3	6	75	25	100	
16	II	BP2-ENG03-Language Through Literature- I	3	6	75	25	100	
17	III	Core III : Java and Data Structures@	4	5	75	25	100	
18	III	Practical III : Data Structures using Java Lab@	3	3	60	40	100	
19	III	BST-CSA01-Allied III-Statistics I@	5	8	75	25	100	
20	IV	Soft Skill	3	2	50	50	100	
21	IV	Environmental Studies	Examination will be held in Semester IV					
Total Credits			21					
SEMESTER IV								
22	I	Tamil/ Other languages – IV	3	6	75	25	100	
23	II	BP2-ENG04- Language Through Literature - II	3	6	75	25	100	
24	III	Core IV : Relational Database Management System@	4	5	75	25	100	
25	III	Practical IV : PL/SQL Lab@	3	3	60	40	100	
26	III	BST-CSA02- Allied IV- Statistics II@	5	8	75	25	100	
27	IV	Soft Skill	3	2	50	50	100	
28	IV	Environmental Studies	2	-	75	25	100	
Total Credits			23					

SEMESTER V			CREDITS	Lecture Hour	EXT. MKS	INT. MKS	TOTAL
29	III	Core V: Operating Systems@	4	5	75	25	100
30	III	Core VI: Computer Networks@	5	5	75	25	100
31	III	Core VII: Computer Vision	4	5	75	25	100
32	III	Practical V: Computer Vision lab (Open CV)	3	4	60	40	100
33	III	Elective I-Choose any one from the list@	5	5	60	40	100
34	III	Practical VI :Elective 1 Lab@	3	4	75	25	100
35	IV	Value Education	2	2	75	25	100
Total Credits			26				
SEMESTER VI			CREDITS	Lecture Hour	EXT. MKS	INT. MKS	TOTAL
36	III	Core VIII: Cloud Computing@	4	6	75	25	100
37	III	Core IX:Internet Of Things@	5	6	75	25	100
38	III	Core X: Machine Learning@	4	6	75	25	100
39	III	Practical VII: Machine Learning Lab@	3	3	60	40	100
40	III	Elective II-Choose any one from the list@	5	5	75	25	100
41	III	Core Practical-VIII – Mini Project@	5	4**	60	40	100
42	V	Extension Activities	1				
Total Credits			27				
Total credits (Core, Elective, SBS)			143				
Non Major Elective I - I Semester							
*NME: Choose Any one From the Other Department							
Non Major Elective II – II Semester							
*NME: Choose Any one From the Other Department							
Elective I							
Natural Language processing + Lab Mobile Application Development for Machine Learning + Lab Introduction to Reinforcement Learning + Lab							
Elective II							
Information Security Network Security Block chain Technology							

@ - Common subject of other course/s.

** Lab Hours (Lab Hours for Development of Mini Project [External project optional])

INTRODUCTION TO ARTIFICIAL INTELLIGENCE

OBJECTIVES:

- Gain a historical perspective of AI and its foundations.
- Become familiar with basic principles of AI toward problem solving, inference, perception, knowledge representation, and learning.
- Experience AI development tools such as an Prolog
- Explore the current scope, potential, limitations, and implications of intelligent systems.

OUTCOMES:

- Demonstrate fundamental understanding of the history of artificial intelligence (AI) and its foundations.
- Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning.
- Demonstrate awareness and a fundamental understanding of various applications of AI techniques
- Demonstrate proficiency developing applications in Prolog.
- Demonstrate an ability to share in discussions of AI, its current scope and limitations, and societal implications

UNIT - I: Introduction, growth and Applications of AI

What Is Artificial Intelligence?: Brain Science and Problem Solving, The Turing Test and Chatterbots-The History of AI: The First Beginnings, Logic Solves (Almost) All Problems, The New Connectionism, Reasoning Under Uncertainty, Distributed; Autonomous and Learning Agents, AI Grows Up-The AI Revolution: AI and Society, Does AI Destroy Jobs? - Agents Knowledge - Based Systems

UNIT - II: Propositional logic

Reasoning in Daily Life-Inference Patterns, Validity, and Invalidity-Classification, Consequence, and Update-The Language of Propositional Logic: From natural language to logical notation, Inclusive and exclusive disjunction, implications, Double implications, Ambiguity - Semantic Situations, Truth Tables, Binary Arithmetic, Tautology, Contradiction, conjunctive normal form, equivalence of propositions

UNIT - III: First-Order Predicate Logic

Inference rules: Modus Ponens, Modus Tollens, Hypothetical Syllogism, Disjunctive Syllogism, Addition, Simplification, Resolution - Quantifiers in First-order logic-Properties of Quantifiers-Free and Bound Variables-Inference in First-Order Logic:FOL inference rules for quantifier: Universal Generalization, Universal Instantiation, Existential Instantiation, Existential introduction - Unification: Conditions for Unification, Unification Algorithm- Resolution: Steps for Resolution- Forward Chaining and backward chaining

UNIT - IV: Prolog

Why prolog for AI- Rules- Syntax- Constants- variables- characters- operators- Equality & unification - Arithmetics - Satisfying goals

UNIT - V: Data structures in Prolog

Trees- lists- Recursive search- Mapping- Recursive Comparison- Joining Structures - Accumulators (VS) Difference structures- Backtrack: Multiple solutions- The 'Cut':- Uses of Cut- problems with Cut

TEXT BOOK:

1. Introduction to Artificial Intelligence, Wolfgang Erte, Springer, Cham
2. Programming in Prolog, W.F. Clocksin, C.S. Mellish.-5th ed, Springer

REFERENCES:

1. Artificial Intelligence For Dummies, John Paul Mueller, Luca Massaron; Dummies
2. Artificial Intelligence: A Modern Approach. Stuart Russell, Peter Norvig; Prentice Hall
3. PROLOG: Programming for Artificial Intelligence, BRATKO, Pearson
4. Prolog by Example: by Helder Coelho, Jose C. Cotta, Springer

WEB REFERENCES:

1. <https://logicinaction.org/docs/ch2.pdf>

PRACTICAL - I :

I YEAR / I SEM

PROLOG LAB**OBJECTIVES:**

- Understand Formal logic and associated forms of programming
- Understand Reasoning modeling
- Interpret the logical consequences and validity of formulae using the rules of propositional and predicate logic
- Assess the completeness of Resolution Procedure, Soundness and completeness of Linear Resolution, Unification and Selective Linear Definite Resolution.

OUTCOMES:

- Demonstrate Logic Programming Paradigm, Prolog execution models, Prolog's basic and advanced prolog concepts such as LIST, CUT, and Fail using illustrative programming examples.
- Convert world knowledge into FOPL formula and construct well-crafted prolog programmes of moderate size
- Apply truth functional propositional Logic(PL) and first order predicate logic (FOPL) to world knowledge
- Describe the basic predicates to manipulate list data structure and sorting algorithms using PROLOG programming

LIST OF EXERCISES:

1. Check if the predicate functor(Term, F, A) succeeds if Term has functor F and arity A by defining a functor
2. Find the number of elements of a list (with size >1) and remove the middle element (at position size//2)
3. Create a list containing all integers within a given range and Reverse the list
4. Write a function append to concatenate elements of two lists into a third list
5. Write a predicate table/3 which prints the truth table of a given logical expression in two variables.
6. Write a Prolog program using (;) operator to decide whether or not any number was between two other numbers (i.e., to check if number N is between two numbers N1 and N2 if **either** N is greater than N1 and N is less than N2 **or** N is less than N1 or N is greater than N2.)
7. Write a Prolog function to find the factorial of a number
8. Query if an element is a member of a list, (using member predicate). To the in-built select function pass a member of the list and return the remainder of the list
9. Demonstrate the use of built-in predicate function findall in a program
10. Evaluate mathematical expressions involving power(**), integer division(/), mod, sqrt and the other basic math operations(+,-,*,%)

11. Use conditions to check the greatest of given two numbers in the stdin/input (not using the max operator)
12. Check the negation of the goal using \+ operator
13. Define a new infix operator is_bigger to compare the size of two animals mentioned in the facts
14. Check whether a given term represents a binary tree
15. Construct a completely balanced binary tree in Prolog

REFERENCES:

<https://sites.google.com/site/prologsite/prolog-problems/>
<https://ocw.upj.ac.id/files/Textbook-TIF212-Prolog-Tutorial-3.pdf>
<http://www.cse.unsw.edu.au/~billw/dictionaries/prolog/>
<https://www.dai.ed.ac.uk/groups/ssp/bookpages/quickprolog/>
<https://www.educba.com/prolog-programming/>

CORE - II :

I YEAR / II SEM

PROBLEM SOLVING USING PYTHON

OBJECTIVES:

- Describe the core syntax and semantics of Python programming language.
- Discover the need for working with the strings and functions.
- Illustrate the process of structuring the data using lists, dictionaries, tuples and sets.
- Understand the usage of packages and Dictionaries.

OUTCOMES:

- To Understand the principles of Python and acquire skills in programming in python
- To develop the emerging applications of relevant field using Python
- To Interpret the fundamental Python syntax and semantics and be fluent in the use of Python control flow statements.
- To Able to develop simple turtle graphics programs in Python

UNIT – I

Introduction: The essence of computational problem solving – Limits of computational problem solving- Computer algorithms-Computer Hardware-Computer Software-The process of computational problem solving-Python programming language - Literals - Variables and Identifiers - Operators - Expressions and Data types.

UNIT - II

Control Structures: Boolean Expressions - Selection Control - If Statement- Indentation in Python- Multi-Way Selection -- Iterative Control- While Statement- Infinite loops- Definite vs. Indefinite Loops- Boolean Flags and Indefinite Loops. Lists: List Structures - Lists in Python - Iterating over lists in Python.

UNIT - III

Functions: Program Routines- Defining Functions- More on Functions: Calling Value-Returning Functions- Calling Non-Value-Returning Functions- Parameter Passing - Keyword Arguments in Python - Default Arguments in Python-Variable Scope.

UNIT - V

Objects and their use: Software Objects - Turtle Graphics – Turtle attributes-Modular Design: Modules - Top- Down Design - Python Modules - Text Files: Opening, reading and writing text files - String Processing - Exception Handling.

UNIT - V

Dictionaries and Sets: Dictionary type in Python - Set Data type. Object Oriented Programming using Python: Encapsulation - Inheritance – Polymorphism. Recursion: Recursive Functions.

TEXT BOOK:

1. Charles Dierbach, "Introduction to Computer Science using Python - A computational Problem solving Focus", Wiley India Edition, 2015.

REFERENCE BOOKS:

1. Mark Lutz, "*Learning Python Powerful Object Oriented Programming*", O'reilly Media 2018, 5th Edition.
2. Timothy A. Budd, "*Exploring Python*", Tata MCGraw Hill Education Private Limited 2011, 1st Edition.
3. Allen Downey, Jeffrey Elkner, Chris Meyers, "*How to think like a computer scientist: learning with Python*", 2012.
4. Sheetal Taneja & Naveen kumar, "*Python Programming a Modular approach – A Modular approach with Graphics, Database, Mobile and Web applications*", Pearson, 2017.
5. Ch Satyanarayana M Radhika Mani, B N Jagadesh, "*Python programming*", Universities Press 2018.

WEB REFERENCES

- <http://interactivepython.org/courselib/static/pythonds>
- <http://www.ibiblio.org/g2swap/byteofpython/read/>
- <http://www.diveintopython3.net/>
- <http://greenteapress.com/wp/think-python-2e/>
- NPTEL & MOOC courses titled Python programming
- http://spoken-tutorial.org/tutorial-search/?search_foss=Python&search_language=English
<http://docs.python.org/3/tutorial/index.html>

PRACTICAL - II :

I YEAR / II SEM

PYTHON LAB

OBJECTIVES:

- To implement the python programming features in practical applications.
- To write, test, and debug simple Python programs.
- To implement Python programs with conditionals and loops.
- Use functions for structuring Python programs.
- Represent compound data using Python lists, tuples, dictionaries , turtles, Files and modules.

OUTCOMES:

- Understand the numeric or real life application problems and solve them.
- Apply a solution clearly and accurately in a program using Python.
- Apply the best features available in Python to solve the situational problems.

LIST OF EXERCISES:

1. Program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon user's choice.
2. Program to calculate total marks, percentage and grade of a student. Marks obtained in each of the five subjects are to be input by user. Assign grades according to the following criteria:
Grade A: Percentage ≥ 80 Grade B: Percentage ≥ 70 and < 80
Grade C: Percentage ≥ 60 and < 70 Grade D: Percentage ≥ 40 and < 60 Grade E: Percentage < 40
3. Program, to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user.
4. Program to display the first n terms of Fibonacci series.
5. Program to find factorial of the given number using recursive function.
6. Write a Python program to count the number of even and odd numbers from array of N numbers.
7. Python function that accepts a string and calculate the number of upper case letters and lower case letters.

8. Python program to reverse a given string and check whether the give string is palindrome or not.
9. Write a program to find sum of all items in a dictionary.
10. Write a Python program to construct the following pattern, using a nested loop 1
22
333
4444
55555
666666
7777777
88888888
999999999
11. Read a file content and copy only the contents at odd lines into a new file.
12. Create a Turtle graphics window with a specific size.
13. Write a Python program for Towers of Hanoi using recursion
14. Create a menu driven Python program with a dictionary for words and their meanings.
15. Devise a Python program to implement the Hangman Game.

CORE - III :

II YEAR / III SEM

JAVA AND DATA STRUCTURE

OBJECTIVES:

- To enable the students to learn the basic concepts of Java programming
- To use class and objects to create applications
- To have an overview of interfaces, packages, multithreading and exceptions.
- To familiarize students with basic data structures and their use in algorithms.

OUTCOMES:

- Students will be able to develop Java Standalone applications and Applets.
- Choose the appropriate data structure for modeling a given problem.

UNIT - I

History and Evolution of Java - Features of Java - Object Oriented Concepts – Bytecode - Lexical Issues - Data Types – Variables- Type Conversion and Casting- Operators - Arithmetic Operators - Bitwise - Relational Operators - Assignment Operator - The conditional Operator - Operator Precedence- Control Statements – Arrays.

UNIT - II

Classes - Objects - Constructors - Overloading method - Static and fixed methods - Inner Classes - String Class- Overriding methods - Using super-Abstract class - this keyword – finalize() method – Garbage Collection.

UNIT - III

Packages - Access Protection - Importing Packages - Interfaces - Exception Handling - Throw and Throws- The Java Thread Model- Creating a Thread and Multiple Threads - Thread Priorities Synchronization-Inter thread Communication - Deadlock
- Suspending, Resuming and stopping threads – Multithreading-I/O Streams - File Streams - Applets .

UNIT - IV

Abstract Data Types(ADTs)-List ADT-Array based implementation-linked list implementation-singly linked list-doubly linked list-circular linked list-Stack ADT operations-Applications-Evaluating arithmetic expressions-Conversion of infix to postfix expression-Queue ADT-operations-Applications of Queues.

UNIT - V

Trees-Binary Trees- representation - Operations on Binary Trees- Traversal of a Binary Tree -Binary Search Trees, Graphs- Representation of Graphs - Traversal in Graph -Dijkstra’s Algorithm, Depth-First vs Breadth-First Search.

TEXT BOOKS:

1. E.Balagurusamy,” *Programming with Java: A Primer*”, Tata McGraw Hill 2014, 5th Edition.
2. Mark Allen Weiss, “*Data Structures and Algorithms Analysis in C++*”, Person Education 2014, 4th Edition.

REFERENCES:

1. Herbert Schildt, “*JAVA 2: The Complete Reference*”, McGraw Hill 2018, 11th Edition.
2. Aho, Hopcroft and Ullman, “*Data Structures and Algorithms* “, Pearson Education 2003.
3. S. Sahni, “*Data Structures, Algorithms and Applications in JAVA*”, Universities Press 2005, 2nd Edition

WEB REFERENCES:

- NPTEL & MOOC courses titled Java and Data Structures
- <https://nptel.ac.in/courses/106106127/>
- <https://nptel.ac.in/courses/106105191/>

PRACTICAL - III :

II YEAR / III SEM

DATA STRUCTURES USING JAVA LAB**OBJECTIVES:**

- To implement linear and non-linear data structures
- To understand the different operations of search trees
- To implement graph traversal algorithm

OUTCOMES:

- Write functions to implement linear and non-linear data structure operations.
- Suggest appropriate linear and non-linear data structure operations for solving a given problem.
-

LIST OF EXERCISES:

1. Write a Java program to implement the Stack ADT using a singly linked list.
2. Write a Java program to implement the Queue ADT using a singly linked list.
3. Write a Java program for the implementation of circular Queue.
4. Write a Java program that reads an infix expression, converts into postfix form
5. Write a Java program to evaluate the postfix expression (use stack ADT).
6. Write a Java program to Insert an element into a binary search tree.
7. Write a Java program to delete an element from a binary search tree.
8. Write a Java program to search for a key element in a binary search tree.
9. Write a Java program for the implementation of BFS for a given graph.
10. Write a Java program for the implementation of DFS for a given graph

RELATIONAL DATABASE MANAGEMENT SYSTEM**OBJECTIVES:**

- Gain a good understanding of the architecture and functioning of Database Management Systems
- Understand the use of Structured Query Language (SQL) and its syntax.
- Apply Normalization techniques to normalize a database.
- Understand the need of transaction processing and learn techniques for controlling the consequences of concurrent data access.

OUTCOMES:

- Describe basic concepts of database system
- Design a Data model and Schemas in RDBMS
- Competent in use of SQL
- Analyze functional dependencies for designing robust Database

UNIT - I

Introduction to DBMS– Data and Information - Database – Database Management System – Objectives - Advantages – Components - Architecture. ER Model: Building blocks of ER Diagram – Relationship Degree – Classification – ER diagram to Tables – ISA relationship – Constraints – Aggregation and Composition – Advantages

UNIT - II

Relational Model: CODD's Rule- Relational Data Model - Key - Integrity – Relational Algebra Operations – Advantages and limitations – Relational Calculus – Domain Relational Calculus - QBE.

UNIT - III

Structure of Relational Database. Introduction to Relational Database Design - Objectives – Tools – Redundancy and Data Anomaly – Functional Dependency - Normalization – 1NF – 2NF – 3NF – BCNF. Transaction Processing – Database Security.

UNIT - IV

SQL: Commands – Data types – DDL - Selection, Projection, Join and Set Operations – Aggregate Functions – DML – Modification - Truncation - Constraints – Subquery.

UNIT - V

PL/SQL: Structure - Elements – Operators Precedence – Control Structure – Iterative Control - Cursors - Procedure

- Function - Packages – Exceptional Handling - Triggers.

TEXT BOOK:

1. S. Sumathi, S. Esakkirajan, “*Fundamentals of Relational Database Management System*”, Springer International Edition 2007.

REFERENCE BOOKS:

1. Abraham Silberchatz, Henry F. Korth, S. Sudarshan, “*Database System Concepts*”, McGrawHill 2019, 7th Edition.
2. Alexis Leon & Mathews Leon, “*Fundamentals of DBMS*”, Vijay Nicole Publications 2014, 2nd Edition.

WEB REFERENCES:

- NPTEL & MOOC courses titled Relational Database Management Systems
- <https://nptel.ac.in/courses/106106093/>
- <https://nptel.ac.in/courses/106106095/>

PL/SQL LAB**OBJECTIVES:**

- Learn the various DDL and DML commands
- Understand queries in SQL to retrieve information from database
- Understand PL/SQL statements: Exception Handling, Cursors, and Triggers.
- Develop database applications using front-end and back-end tools.

OUTCOMES:

- Implement the DDL , DML Commands and Constraints
- Create, Update and query on the database.
- Design and Implement simple project with Front End and Back End.

LIST OF EXERCISES

- 1) DDL commands with constraints.
- 2) DML Commands with constraints.
- 3) SQL Queries: Queries, sub queries, Aggregate function
- 4) PL/SQL : Exceptional Handling
- 5) PL/SQL : Cursor
- 6) PL/SQL : Trigger
- 7) PL/SQL : Packages
- 8) Design and Develop Application for Library Management
- 9) Design and Develop Application for Student Mark Sheet Processing
- 10) Design and Develop Application for PayRoll Processing

OPERATING SYSTEM**OBJECTIVES:**

- To understand the fundamental concepts and role of Operating System.
- To learn the Process Management and Scheduling Algorithms
- To understand the Memory Management policies
- To gain insight on I/O and File management techniques

OUTCOMES:

- Understand the structure and functions of Operating System
- Compare the performance of Scheduling Algorithms
- Analyze resource management techniques

UNIT - I

Introduction: Views - Types of System - OS Structure – Operations - Services – Interface- System Calls- System Structure - System Design and Implementation. Process Management: Process - Process Scheduling - Inter-process Communication. CPU Scheduling: CPU Schedulers - Scheduling Criteria - Scheduling Algorithms.

UNIT - II

Process Synchronization: Critical- Section Problem - Synchronization Hardware Semaphores - Classical Problems of Synchronization - Monitors. Deadlocks: Characterization - Methods for Handling Deadlocks - Deadlock Prevention - Avoidance - Detection - Recovery.

UNIT - III

Memory Management: Hardware - Address Binding – Address Space - Dynamic Loading and Linking – Swapping – Contiguous Allocation - Segmentation - Paging – Structure of the Page Table.

UNIT - IV

Virtual Memory Management: Demand Paging - Page Replacement Algorithms - Thrashing. File System: File Concept -. Access Methods - Directory and Disk Structure - Protection - File System Structures - Allocation Methods - Free Space Management.

UNIT - V

I/O Systems: Overview - I/O Hardware - Application I/O Interface - Kernel I/O Subsystem - Transforming I/O Requests to Hardware Operations - Performance. System Protection: Goals - Domain - Access matrix. System Security: The Security Problem - Threats – Encryption- User Authentication.

TEXT BOOK:

1. Abraham Silberschatz, Peter B Galvin, Greg Gagne, “*Operating System Concepts*”, Wiley India Pvt. Ltd 2018, 9th Edition,.

REFERENCES:

1. William Stallings, “*Operating Systems Internals and Design Principles*”, Pearson, 2018, 9th Edition.
2. Andrew S. Tanenbaum, Herbert Bos, “*Modern Operating Systems*”, Pearson 2014, 4th Edition.

WEB REFERENCES:

- NPTEL & MOOC courses titled Operating Systems <https://nptel.ac.in/courses/106106144/>

CORE - VI :

III YEAR / V SEM

COMPUTER NETWORKS

OBJECTIVES:

- To understand the concept of Computer network
- To impart knowledge about networking and inter networking devices

OUTCOMES:

- Analyze different network models
- Describe, analyze and compare a number of data link, network and transport layer
- Analysing key networking protocols and their hierarchical relationship in the conceptual model like TCP/IP and OSI

UNIT - I

Introduction – Network Hardware - Software - Reference Models - OSI and TCP/IP Models - Example Networks: Internet, ATM, Ethernet and Wireless LANs - Physical Layer - Theoretical Basis for Data Communication - Guided Transmission Media.

UNIT - II

Wireless Transmission - Communication Satellites - Telephone System: Structure, Local Loop, Trunks and Multiplexing and Switching. Data Link Layer: Design Issues - Error Detection and Correction.

UNIT - III

Elementary Data Link Protocols - Sliding Window Protocols - Data Link Layer in the Internet - Medium Access Layer - Channel Allocation Problem - Multiple Access Protocols - Bluetooth.

UNIT - IV

Network Layer - Design Issues - Routing Algorithms - Congestion Control Algorithms - IP Protocol - IP Addresses - Internet Control Protocols.

UNIT - V

Transport Layer - Services - Connection Management - Addressing, Establishing and Releasing a Connection - Simple Transport Protocol - Internet Transport Protocols (ITP) - Network Security: Cryptography.

TEXT BOOK :

1. A. S. Tanenbaum, “*Computer Networks*”, Prentice-Hall of India 2008, 4th Edition.

REFERENCE BOOKS:

1. Stallings, “*Data and Computer Communications*”, Pearson Education 2012, 7th Edition.
2. B. A. Forouzan, “*Data Communications and Networking*”, Tata McGraw Hill 2007, 4th Edition.
3. F. Halsall, “*Data Communications, Computer Networks and Open Systems*”, Pearson Education 2008.
4. D. Bertsekas and R. Gallager, “*Data Networks*”, PHI 2008, 2nd Edition.
5. Lamarca, “*Communication Networks*”, Tata McGraw Hill 2002.

WEB REFERENCES:

- NPTEL & MOOC courses titled Computer Networks
- <https://nptel.ac.in/courses/106106091/>

CORE - VII :

III YEAR / V SEM

COMPUTER VISION

OBJECTIVES:

- To introduce students the fundamentals of image formation;
- To introduce students the major ideas, methods, and techniques of computer vision and pattern recognition;
- To develop an appreciation for various issues in the design of computer vision and object recognition systems; and
- To provide the student with programming experience from implementing computer vision and object recognition applications.

OUTCOMES:

- Identify basic concepts, terminology, theories, models and methods in the field of computer vision
- Describe known principles of human visual system
- Describe basic methods of computer vision related to multi-scale representation, edge detection and detection of other primitives, stereo, motion and object recognition
- Suggest a design of a computer vision system for a specific problem

Unit1: Introduction

Image Processing, Computer Vision and Computer Graphics, what is Computer Vision - Low-level, Mid-level, High-level, Overview of Diverse Computer Vision Applications: Document Image Analysis, Biometrics, Object Recognition, Tracking, Medical Image Analysis, Content-Based Image Retrieval, Video Data Processing, Multimedia, Virtual Reality and Augmented Reality

Unit II: Digital Image Formation and low-level processing

Fundamentals of Image Formation, Transformation: Orthogonal, Euclidean, Affine, Projective; Fourier Transform, Convolution and Filtering, Image Enhancement, Restoration, Histogram Processing.

Unit III: Feature Extraction & Image Segmentation

Edges - Canny, LOG, DOG; Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Gabor Filters and DWT; Image Segmentation: Contour based representation, Region based representation, Level set representations, Fourier and wavelet descriptors, Multiresolution analysis.

Unit IV: Pattern & Motion Analysis

Clustering: K-Means, K-Medoids, Classification: Discriminant Function, Supervised, Un-supervised, Semi-supervised; Classifiers: Bayes, KNN, Dimensionality Reduction: LDA, ICA, Background Subtraction and Modeling, Spatio-Temporal Analysis, Dynamic Stereo; Motion parameter estimation.

Unit V: Applications

Photo album – Face detection – Face recognition – Active appearance and 3D shape models of faces
Application: Surveillance – foreground-background separation – particle filters – tracking and occlusion – combining views from multiple cameras – human gait analysis
Application: In-vehicle vision system: locating roadway – road markings – identifying road signs – locating pedestrians

TEXT BOOK:

1. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited 2011.

REFERENCE BOOK:

1. Computer Vision: A Modern Approach, D. A. Forsyth, J. Ponce, Pearson Education, 2003.
2. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, Second Edition, Cambridge University Press, March 2004.
3. Christopher M. Bishop; Pattern Recognition and Machine Learning, Springer, 2006
4. E. R. Davies, Computer & Machine Vision, Fourth Edition, Academic Press, 2012
5. Simon J. D. Prince, Computer Vision: Models, Learning, and Inference, Cambridge University Press, 2012
6. Mark Nixon and Alberto S. Aquado, Feature Extraction & Image Processing for Computer Vision, Third Edition, Academic Press, 2012.

WEB REFERENCES:

1. <https://viso.ai/blog/>
2. <https://learnopencv.com/>
3. <https://www.analyticsvidhya.com/blog/>
4. <https://www.rsipvision.com/rsip-vision-learns/>

PRACTICAL - V :

III YEAR / V SEM

COMPUTER VISION - LAB

OBJECTIVE:

- Understand What Is A Digital Image and what is Manipulating Images
- Understand Manipulating Images One Pixel At a Time, Pixel Transformations, geometric Operations

OUTCOMES:

- Implement Spatial Operations in Image Processing
- Implement the Image Gradients and Edge Detection Techniques
- Implement Extraction of desired features
- Implement object detection

LIST OF EXERCISES:

1. Perform the image transformations that include the geometric and morphological transformations.
2. Perform the image enhancement by applying contrast limited adaptive histogram Equalization
3. Perform the Contours and Region based segmentation in images
4. Perform the Wavelet Transforms on image using PyWavelets.
5. Perform the K-Means clustering for Image segmentation using CV2 library.
6. Perform basic motion detection and tracking using python and OpenCV
7. Perform face detection using OpenCV library
8. Perform Foreground Extraction in an image
9. Perform Pedestrian Detection using OpenCV and Python.

ELECTIVE - I :

III YEAR / V SEM

CHOOSE FROM THE GIVEN ELECTIVE - I-A/ELECTIVE - I-B/ELECTIVE - I-C

ELECTIVE - I-A :

III YEAR / V SEM

NATURAL LANGUAGE PROCESSING

OBJECTIVES:

- To grasp the significance of natural language processing in solving real-world problems.
- To map the appropriate processing technique to a problem and implement the technique.
- To demonstrate required design skills for large collection sets.
- To appreciate the theoretical formulation of the natural language processing techniques.

OUTCOMES:

- Describe the fundamental concepts and techniques of natural language processing.
- Distinguish among the various techniques, taking into account the assumptions, strengths, and weaknesses of each.
- Use appropriate descriptions, visualizations, and statistics to communicate the problems and their solutions.
- Analyze large volume text data generated from a range of real-world applications.

Unit – I: Introduction

Overview: Origins and challenges of NLP- Theory of Language -Features of Indian Languages – Issues in Font –Models and Algorithms- NLP Applications.

UNIT II - MORPHOLOGY AND PARTS-OF-SPEECH

Phonology – Computational Phonology - Words and Morphemes – Segmentation – Categorization and Lemmatisation – Word Form Recognition – Valency - Agreement - Regular Expressions – Finite State Automata – Morphology- Morphological issues of Indian Languages – Transliteration.

UNIT III - PROBABILISTIC MODELS

Probabilistic Models of Pronunciation and Spelling – Weighted Automata – N- Grams – Corpus Analysis – Smoothing – Entropy - Parts-of-Speech – Taggers – Rule based – Hidden Markov Models – Speech Recognition.

UNIT IV - SYNTAX

Basic Concepts of Syntax – Parsing Techniques – General Grammar rules for Indian Languages – Context Free Grammar – Parsing with Context Free Grammars – Top Down Parser – Earley Algorithm – Features and Unification - Lexicalised and Probabilistic Parsing.

UNIT V - SEMANTICS AND PRAGMATICS (6 hours) Representing Meaning – Computational Representation – Meaning Structure of Language – Semantic Analysis – Lexical Semantics – WordNet – Pragmatics – Discourse – Reference Resolution – Text Coherence – Dialogue Conversational Agents.

TEXT BOOK:

1. Ronald Hausser, “Foundations of Computational Linguistics”, Springer-Verleg, 1999.

REFERENCES:

2. Daniel Jurafsky and James H. Martin “Speech and Language Processing”, Prentice Hall, 2009.
3. Christopher D.Manning and Hinrich Schutze, “Foundation of Statistical Natural Language Processing”, MIT Press, 1999.
4. James Allen, “Natural Language Understanding”, Benjamin/Cummings Publishing Co. 1995.
5. Applied Natural Language Processing with Python: Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing, By Taweh Beysolow II, September 2018

WEB REFERENCES:

1. <https://blog.algorithmia.com/introduction-natural-language-processingnlp/>
2. <https://www.udacity.com/course/natural-language-processingnanodegree--nd892>.
<https://www.coursera.org/learn/language-processing>
3. <https://towardsdatascience.com/a-practitioners-guide-to-naturallanguage-processing-part-i-processing-understanding-text-9f4abfd13e72>
4. <https://www.edx.org/course/natural-language-processing>

ELECTIVE - I-B :

III YEAR / V SEM

MOBILE APPLICATION DEVELOPMENT FOR MACHINE LEARNING

OBJECTIVES:

- To Understand Android with Machine Learning
- To envisage outcomes automatically without human interference
- To learn Machine learning serving mobile application
- To analysis user behavior patterns

OUTCOMES:

- Android App Development with Machine Learning
- Targeted user behavior patterns
- Provide suggestions as well as recommendations for search requests
- Suggest apps which will reduce human interference

Unit I:

Mobile App Development:

Mobile App Development – Android Platform – Android SDK – Eclipse Installation – Building simple Android Applications.

Unit II:

Android Application Design Essentials:

Anatomy of an Android Application – Android Terminologies: Application Context, Activities, Services, Intents – Receiving and broadcasting Intents – Intent Filters – Content Providers.

Unit III:

Android User Interface Design Essentials:

User Interface Design Elements – Designing User Interfaces with Layouts – Drawing and working with animation.

Unit IV:

TensorFlow Lite:

Converting a Machine Learning model to TensorFlow Lite – Saving the model – Loading the Model and allocating Tensors – Performing prediction – Optimizing the model.

Unit – V:

TensorFlow Android App Development:

Editing Layout File – Adding TensorFlow Lite dependencies – Adding TensorFlow Lite model – Writing Activity code for getting insights from data - Case Study: Processing Images on a Mobile.

TEXT BOOKS:

1. Ryan Cohen and Tao Wang, “GUI Design for Android Apps”, Apress, 2014.
2. John Horton, “Android Programming for Beginners”, Second Edition, Packt, 2018.

REFERENCES:

1. Ted Hagos, “Learn Android Studio with Kotlin: Efficient Android App Development”, Apress, 2018.
2. Laurence Moroney, “AI and ML for Coders”, O’Reilly, 2020.
3. Reto Meier, “Professional Android 2 Application Development”, Wiley India Pvt Ltd
4. Mark L Murphy, “Beginning Android”, Wiley India Pvt Ltd
5. Android Application Development All in one for Dummies by Barry Burd, Edition: I

WEB REFERENCES:

<http://venturebeat.com/2009/01/01/android-netbooks-on-their-way-likely-by-2010/>

ELECTIVE - I-C :

III YEAR / V SEM

INTRODUCTION TO REINFORCEMENT LEARNING

OBJECTIVES:

- To understand statistical learning techniques where an agent explicitly takes actions and interacts with the world.
- To understand the importance and challenges of learning agents that make decisions.
- To introduce you to the fundamentals of Reinforcement Learning.

OUTCOMES:

- Formalize problems as Markov Decision Processes
- Understand basic exploration methods and the exploration/exploitation tradeoff
- Understand value functions, as a general-purpose tool for optimal decision-making
- Know how to implement dynamic programming as an efficient solution approach to an industrial control problem.
- Will be able to start using RL for real problems,

Unit – I: Monte-Carlo Methods

Monte-Carlo methods: policy evaluation, rollouts, on policy and off-policy learning, importance sampling

Unit – II: Temporal Difference Learning

Temporal Difference learning: TD prediction, Optimality of TD(0), SARSA, Q-learning, Games and after states, Maximization Bias and Double Learning.

Unit – III: EligibilityTraces

Eligibility traces: n-step TD prediction, TD(λ), forward and backward views, Q(λ), SARSA(λ), replacing traces and accumulating traces.

Unit – IV: Function Approximation

Function Approximation: Value prediction, gradient descent methods, linear function approximation, Control algorithms, Fitted Iterative Methods, Deep Q-learning.

Unit – V: Computer Vision

Policy Gradient methods: non-associative learning - REINFORCE algorithm, exact gradient methods, estimating gradients, approximate policy gradient algorithms, actor-critic methods, Asynchronous Advantage Actor-Critic.

Hierarchical RL: MAXQ framework, Options framework, HAM framework, Option discovery algorithms.

Case studies: Elevator dispatching, Samuel's checker player, TD-gammon, Acrobot, Helicopter piloting, Alpha Go

BOOKS:

- R. S. Sutton and A. G. Barto. Reinforcement Learning - An Introduction. MIT Press. 2nd Edition. 2018.
- Richard S. Sutton and Andrew G. Barto, "Reinforcement learning: An introduction", Second Edition, MIT Press, 2019

REFERENCES

1. Li, Yuxi. "Deep reinforcement learning." arXiv preprint arXiv:1810.06339 (2018).
2. Wiering, Marco, and Martijn Van Otterlo. "Reinforcement learning." Adaptation, learning, and optimization 12 (2012)
3. Russell, Stuart J., and Peter Norvig. "Artificial intelligence: a modern approach." Pearson Education Limited, 2016.
4. Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. "Deep learning." MIT press, 2016.

WEB REFERENCES:

1. David Silver's course on Reinforcement Learning (link)
2. Video Lectures by Prof. David Silver
3. Video Lectures by Prof. B.Ravindran

PRACTICAL - VI :

III YEAR / V SEM

CHOOSE PRACTICAL OF ELECTIVE - I-A/ELECTIVE - I-B/ELECTIVE - I-C
WHICH WAS SELECTED IN THEORY

PRACTICAL -VI-A :

III YEAR / V SEM

NATURAL LANGUAGE PROCESSING LAB

OBJECTIVES:

- To understand the algorithms available for the processing of linguistic information and computational properties of natural languages.
- To conceive basic knowledge on various morphological, syntactic and semantic NLP tasks.
- To familiarize various NLP software libraries and datasets publicly available.
- To develop systems for various NLP problems with moderate complexity.
- To learn various strategies for NLP system evaluation and error analysis.

OUTCOMES:

- Describe the concepts of morphology, syntax, semantics, discourse & pragmatics of natural language.
- Demonstrate understanding of the relationship between NLP and statistics & machine learning.
- Discover various linguistic and statistical features relevant to the basic NLP task, namely, spelling correction, morphological analysis, parts-of-speech tagging, parsing and semantic analysis.
- Develop systems for various NLP problems with moderate complexity

LIST OF EXERCISES

1. How to tokenize a given text?
2. How to get the sentences of a text document ?
3. How to tokenize text with stop words as delimiters?
4. How to remove stop words and punctuations in a text ?

5. How to perform stemming?
6. How to lemmatize a given text ?
7. How to extract usernames from emails ?
8. How to find the most common words in the text excluding stopwords?
9. How to do spell correction in a given text ?
10. How to classify a text as positive/negative sentiment?
11. How to extract Noun and Verb phrases from a text ?
12. How to find the ROOT word of any word in a sentence?
13. Write a Python program to load the iris data from a given csv file into a dataframe and print the shape of the data, type of the data and first 3 rows.
14. Write a Python NLTK program to find the sets of synonyms and antonyms of a given word.
15. Write a Python NLTK program to print the first 15 random combine labeled male and labeled female names from names corpus.

TEXT BOOKS:

1. Jurafsky Dan and Martin James H. “Speech and Language Processing” ,3rd Edition, 2018.

REFERENCE BOOKS:

1. Jurafsky D. and Martin J. H., “Speech and language processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition”, 2nd Edition, Upper Saddle River, NJ: Prentice-Hall, 2008.
2. Goldberg Yoav “A Primer on Neural Network Models for Natural Language Processing”.
3. Natural Language Processing with Python, Steven Bird, Ewan Klein, and Edward Loper

PRACTICAL -VI-B :

III YEAR / V SEM

MOBILE APPLICATION DEVELOPMENT FOR MACHINE LEARNING - LAB

OBJECTIVES:

- Discuss the components and different Layout for mobile application development framework for android
- Design Simple GUI application with the Use of Built in components and widgets.
- Define a solution for complex problems

OUTCOMES:

- Develop components and different Layout for mobile application development framework for android
- Develop Simple GUI application with the Use of Built in components and widgets.
- Create databases to store application data locally.
- Test and Show the results on emulators or on physical handheld devices

LIST OF EXERCISES:

1. Calculator App – Setting up the UI using Linear Layouts
2. Calculator App – Using OnClick and implementing CLR functionality
3. Calculator App – Adding the OnDecimalPoint and OnOperator functionalities
4. Calculator App – Adding Subtraction using split and substring methods
5. Calculator App – Adding other required operators and finalizing the App
6. Quiz App – Setting up the UI using cardview
7. Quiz App – Creating the Question Model and preparing the Questions
8. Quiz App – Setting up the Question UI and Question Activity
9. Quiz App – Selecting right and wrong answers and displaying subsequent Question
10. Quiz App – Preparing UI for the Result Activity
11. TensorFlow Regression App – Editing the Layout File
12. TensorFlow Regression App – Adding TensorFlow Lite Dependencies
13. TensorFlow Regression App – Adding TensorFlow Lite Model
14. TensorFlow Regression App – Writing Activity code for inference

TEXT BOOKS:

1. Ryan Cohen and Tao Wang, “GUI Design for Android Apps”, Apress, 2014.
2. John Horton, “Android Programming for Beginners”, Second Edition, Packt, 2018.
3. Ted Hagos, “Learn Android Studio with Kotlin: Efficient Android App Development”, Apress, 2018.
4. Laurence Moroney, “AI and ML for Coders”, O’Reilly, 2020.

PRACTICAL -VI-C :

III YEAR / V SEM

REINFORCEMENT LEARNING - LAB**OBJECTIVES:**

- To understand basic and advanced reinforcement learning techniques
- To understand application of right technique at different problem situations
- To understand the importance of Domain Knowledge

OUTCOMES:

- Knowledge of basic and advanced reinforcement learning techniques.
- Identification of suitable learning tasks to which these learning techniques can be applied.
- Appreciation of some of the current limitations of reinforcement learning techniques.
- Formulation of decision problems, set up and run computational experiments, evaluation of results from experiments.

LIST OF EXERCISES:

1. Solving Frozen Lake Problem with Value Iteration
2. Solving Frozen Lake Problem with Policy Iteration
3. Every-visit MC Prediction with Blackjack Game
4. First-visit MC Prediction with Blackjack Game
5. Implementing on-policy MC control
6. Computing Optimal Policy using SARSA
7. Computing Optimal Policy using Q Learning
8. Implementing epsilon-greedy algorithm
9. Implementing Softmax Exploration strategy
10. Implementing Upper Confidence Bound Algorithm
11. Implementing Thomson Sampling strategy
12. Finding the Best Advertisement Banner using Bandits

BOOKS:

1. R. S. Sutton and A. G. Barto. Reinforcement Learning - An Introduction. MIT Press. 2nd Edition. 2018.
2. Sudharsan Ravichandran, Deep Reinforcement Learning with Python, PACKT, 2020.
3. Video Lectures by Prof. David Silver
4. Video Lectures by Prof. B.Ravindran

CORE - VIII :

III YEAR / VI SEM

CLLOUD COMPUTING

OBJECTIVES:

- To understand the concepts in Cloud Computing and its Security
- To understand the evolving computer model, cloud computing.
- To introduce the various levels of services that can be achieved by cloud.

OUTCOMES:

- To explain and apply levels of services of Cloud
- To describe the security aspects in the cloud.

UNIT - I

Cloud Computing Foundation: Introduction to Cloud Computing – Move to Cloud Computing – Types of Cloud – Working of Cloud Computing

UNIT - II

Cloud Computing Architecture : Cloud Computing Technology – Cloud Architecture – Cloud Modeling and Design - Virtualization : Foundation – Grid, Cloud and Virtualization – Virtualization and Cloud Computing

UNIT - III

Data Storage and Cloud Computing : Data Storage – Cloud Storage – Cloud Storage from LANs to WANs – Cloud Computing Services : Cloud Services – Cloud Computing at Work

UNIT - IV

Cloud Computing and Security : Risks in Cloud Computing – Data Security in Cloud – Cloud Security Services – Cloud Computing Tools : Tools and Technologies for Cloud – Cloud Mashups – Apache Hadoop – Cloud Tools

UNIT - V

Cloud Applications – Moving Applications to the Cloud – Microsoft Cloud Services – Google Cloud Applications – Amazon Cloud Services – Cloud Applications

TEXT BOOK:

- 1.A.Srinivasan and J.Suresh, “*Cloud Computing – A Practical Approach for Learning and Implementation*”, Pearson India Publications 2014.

REFERENCE BOOK:

1. Rajkumar Buyya, James Broberg, Andrzej , “*Cloud Computing: Principles and Paradigms*”, Wiley India Publications 2011.
2. Arshdeep Bahga and Vijay Madiseti , “*Cloud Computing – A Hands on Approach*”, Universities Press (India) Pvt Ltd. 2014.

WEB REFERENCES:

1. NPTEL & MOOC courses titled Cloud computing <https://nptel.ac.in/courses/106105167/>

CORE - IX :

III YEAR / VI SEM

INTERNET OF THINGS

OBJECTIVES:

- To understand the concepts of Internet of Things and the application of IoT.
- To Determine the Market Perspective of IoT.
- To Understand the vision of IoT from a global context

OUTCOMES:

- Use of Devices, Gateways and Data Management in IoT.
- Design IoT applications in different domain and be able to analyze their performance
- Implement basic IoT applications on embedded platforms.

UNIT – I

IoT & Web Technology, The Internet of Things Today, Time for Convergence, Towards the IoT Universe, Internet of Things Vision, IoT Strategic Research and Innovation Directions, IoT Applications, Future Internet Technologies, Infrastructure, Networks and Communication, Processes, Data Management, Security, Privacy & Trust, Device Level Energy Issues, IoT Related Standardization, Recommendations on Research Topics.

UNIT - II

M2M to IoT – A Basic Perspective– Introduction, Some Definitions, M2M Value Chains, IoT Value Chains, An emerging industrial structure for IoT, The international driven global value chain and global information monopolies. M2M to IoT-An Architectural Overview– Building an architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations.

UNIT - III

IoT Architecture -State of the Art – Introduction, State of the art, Architecture. Reference Model-Introduction, Reference Model and architecture, IoT reference Model, IoT Reference Architecture-Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views.

UNIT - IV

IoT Applications for Value Creations Introduction, IoT applications for industry: Future Factory Concepts, Brownfield IoT, Smart Objects, Smart Applications, Four Aspects in your Business to Master IoT, Value Creation from Big Data and Serialization, IoT for Retailing Industry, IoT For Oil and Gas Industry, Opinions on IoT Application and Value for Industry, Home Management, eHealth.

UNIT - V

Internet of Things Privacy, Security and Governance Introduction, Overview of Governance, Privacy and Security Issues, Contribution from FP7 Projects, Security, Privacy and Trust in IoT-Data-Platforms for Smart Cities, First Steps Towards a Secure Platform, Smartie Approach. Data Aggregation for the IoT in Smart Cities, Security

TEXT BOOK:

1. Vijay Madiseti and ArshdeepBahga, “*Internet of Things: (A Hands-on Approach)*”, Universities Press (INDIA) Private Limited 2014, 1st Edition.

REFERENCE BOOKS:

1. Michael Miller, “*The Internet of Things: How Smart TVs, Smart Cars, Smart Homes, and Smart Cities Are Changing the World*”, Pearson Education 2015.
2. Francis da Costa, “*Rethinking the Internet of Things: A Scalable Approach to Connecting Everything*”, Apress Publications 2013, 1st Edition.
3. Walteneus Dargie, Christian Poellabauer, “*Fundamentals of Wireless Sensor Networks: Theory and Practice*”, Wiley 2014.
4. CunoPfister, “*Getting Started with the Internet of Things*”, O’Reilly Media 2011.

WEB REFERENCES:

1. <https://github.com/connectIOT/iottoolkit>
2. <https://www.arduino.cc/>
3. <http://www.zettajs.org/>

MACHINE LEARNING**OBJECTIVES**

- To understand basic concepts of machine learning
- Understand how to evaluate models generated from data
- Discover how to build machine learning algorithms, prepare data, and use different techniques using Python

OUTCOMES

- Implement different machine learning algorithm techniques.
- Apply the algorithms to a real-world problem, optimize the models learned and report on the expected accuracy that can be achieved by applying the models.
- Apply appropriate data sets to the Machine Learning algorithms.
- Identify and apply Machine Learning algorithms to solve real world problems.

UNIT – I: Introduction

Machine Learning Foundations – Overview – Design of a Learning System – Types of Machine Learning – Supervised Learning and Unsupervised Learning – Applications of Machine Learning – Tools Overview for ML.

UNIT – II: Supervised Learning - I

Simple Linear Regression – Multiple Linear Regression – Polynomial Regression – Ridge Regression – Lasso Regression – Evaluating Regression Models – Model Selection – Bagging – Ensemble Methods.

UNIT – III: Supervised Learning - II

Classification – Logistic Regression – Decision Tree Regression and Classification – Random Forest Regression and Classification – Support Vector Machine Regression and Classification - Evaluating Classification Models.

UNIT – IV: Unsupervised Learning

Clustering – K-Means Clustering – Density-Based Clustering – Dimensionality Reduction – Collaborative Filtering.

UNIT – V: Association Rule Learning

Association Rule Learning – Concepts related to ARL – ARL Algorithms - Apriori – Eclat – Concepts and Algorithms.

TEXT BOOK:

1. Kevin P. Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press, 2012.

REFERENCE BOOK:

1. Ethem Alpaydin, “Introduction to Machine Learning”, MIT Press, Third Edition, 2014.
2. Tom Mitchell, "Machine Learning", McGraw-Hill, 1997.
3. Sebastian Raschka, Vahid Mirjalili, “Python Machine Learning and deep learning”, 2nd edition, kindle book, 2018
4. Carol Quados, “Machine Learning with python, scikit-learn and Tensorflow”, Packet Publishing, 2018
5. Gavin Hackling, “Machine Learning with scikit-learn”, Packet publishing, O'Reilly, 2018

WEB REFERENCES:

1. Stanford Lectures of Prof. Andrew Ng on Machine Learning

MACHINE LEARNING - LAB**OBJECTIVES:**

- Understand the mathematical and statistical perspectives of machine learning algorithms through python programming.
- Understand the basic concepts of deep neural network models and design the same.

OUTCOMES:

- Design and evaluate the unsupervised models through python in built functions.
- Evaluate the machine learning model algorithms by python programming.
- Design and apply various reinforcement algorithms to solve real time complex problems.
- Design and develop the code for the recommender system using Natural Language processing.

LIST OF EXERCISES:

1. Write a program to implement the Simple and Multiple Linear Regression
2. Write a program to implement the Polynomial Regression
3. Write a program to implement the Bagging Technique
4. Write a program to implement the Adaboost Methods
5. Write a program to implement Logistic Regression algorithm
6. Write a program to demonstrate the workflow of Decision Tree Classification
7. Write a program to implement the Random Forest Classification
8. Write a program to implement the SVM Classification
9. Write a program to perform the K Means Clustering
10. Write a program to perform the Density based Clustering
11. Write a program to implement the Apriori algorithm for market basket analysis
12. Write a program to compare the Supervised Machine Learning algorithms.

REFERENCE BOOK:

1. Kevin P. Murphy, "Machine Learning: A Probabilistic Perspective", MIT Press, 2012.
2. Ethem Alpaydin, "Introduction to Machine Learning", MIT Press, Third Edition, 2014.
3. Tom Mitchell, "Machine Learning", McGraw-Hill, 1997.
4. Sebastian Raschka, Vahid Mirjalili, "Python Machine Learning and deep learning", 2nd edition, kindle book, 2018
5. Carol Quados, "Machine Learning with python, scikit-learn and Tensorflow", Packet Publishing, 2018
6. Gavin Hackeling, "Machine Learning with scikit-learn", Packet publishing, O'Reilly, 2018
7. Stanford Lectures of Prof. Andrew Ng on Machine Learning

ELECTIVE - II :

III YEAR / VI SEM

CHOOSE FROM THE GIVEN ELECTIVE - II-A/ELECTIVE - II-B/ELECTIVE - II-C

ELECTIVE - II-A :

III YEAR / VI SEM

INFORMATION SECURITY**OBJECTIVES:**

- To acquire knowledge of cryptography and network security
- To acquire knowledge of security management and incident response
- To acquire knowledge of security in software and operating systems
- To acquire knowledge of data security and secure system development
- To acquire knowledge of privacy and data protection

OUTCOMES:

- Test and evaluate security in systems and networks
- Use methods for planning and designing secure systems

- Apply techniques and tools for secure system deployment and operation
- Perform continuous testing, assessment and updating of system security

Unit I

Introduction to Information Security : Attacks, Vulnerability, Security Goals, Security Services and mechanisms - Conventional Cryptographic Techniques : Conventional substitution and transposition ciphers, One-time Pad

Unit II

Block cipher and Stream Cipher, Steganography - Symmetric and Asymmetric Cryptographic Techniques : DES, AES, RSA algorithms
Authentication and Digital Signatures : Use of Cryptography for authentication,

Unit III

Secure Hash function, Key management – Kerberos - Program Security : Nonmalicious Program errors – Buffer overflow, Incomplete mediation, Time-of-check to Time-of-use Errors, Viruses, Trapdoors

Unit IV

Salami attack, Man-in-the-middle attacks, Covert channels - Security in Networks : Threats in networks, Network Security Controls – Architecture, Encryption, Content Integrity, Strong

Unit V

Authentication , Access Controls, Wireless Security, Honeypots, Traffic flow security, Firewalls – Design and Types of Firewalls, Personal Firewalls, IDS, Email Security – PGP,S/MIME

TEXT BOOKS:

1. “Information Systems Today, Managing in the Digital World”, Third Edition by Leonard
2. M. Jessup; Joseph S. Valacich, Publisher: Prentice Hall
3. “Introduction to Information Technology”, V. Rajaraman, PHI

REFERENCE BOOKS:

1. “Information Systems Management in Practice” Barbara C. McNurlin, Ralph H. Sprague,
2. and Publisher: Pearson Education.
3. Security in Computing, Fourth Edition, by Charles P. Pfleeger, Pearson Education
4. Cryptography And Network Security Principles And Practice, Fourth or Fifth Edition, William Stallings, Pearson
5. Modern Cryptography: Theory and Practice, by Wenbo Mao, Prentice Hall.
6. Network Security Essentials: Applications and Standards, by William Stallings. Prentice Hall.

ELECTIVE - II-B :

III YEAR / VI SEM

NETWORK SECURITY

OBJECTIVES:

- Understand security issues related to networking vulnerabilities, firewalls, intrusion detection systems
- Identify infrastructure components including devices, topologies, protocols, systems software, management and security
- Design and develop solutions for technical issues related to networking and security problems.
- Apply footprinting, scanning, enumeration and similar techniques to discover network and system vulnerabilities
- Analyze performance and risk factors of enterprise network systems

OUTCOMES:

- Describe network security services and mechanisms.
- Symmetrical and Asymmetrical cryptography.
- Data integrity, Authentication, Digital Signatures.
- Various network security applications, IPSec, Firewall, IDS, Web security, Email security, Malicious software etc.

Unit-I

Introduction: Attack - Services and Mechanism - Model for Internetwork Security - Cryptography - Notion of Plain Text- Encryption Key - Ciphertext - Decryption and Cryptanalysis - Public Key Encryption - Digital Signatures and Authentication

Unit-II

Network Security: Authentication Application - Kerberos - Directory Authentication Service - Pretty Good Privacy

Unit-III

IP Security Architecture: Authentication header - Encapsulating Security Payload combining Security Associations - Key Management

Unit-IV

Web Security: Requirement - Secure Sockets Layer - Transport Layer Security - Secure Electronic Transactions, Network Management Security: Overview of SNMP Architecture - SMMPV11 Communication Facility - SNMPV3

Unit-V

System Security: Intruders - Viruses and Related Threats - Firewall Design Principles - Comprehensive examples using available software platforms/case tools - Configuration Management

TEXT BOOKS:

1. William Stallings, "Cryptography and Network Security, Principles and Practices", Pearson Education, Third Edition
2. Charlie Kaufman, Radia Perlman and Mike speciner, "Network security, Private communication in a Public World"
3. Atul Kahate, "Cryptography and Network Security", TMH, Third Edition.
4. V. K. Pachghare "Cryptography and Information Security", PHI

REFERENCE BOOKS:

1. Christopher M. King, "Security architecture, design deployment and operations", Curtis patton and RSA Press.
2. Stephen Northcatt, Leny Zeltser, "INSIDE NETWORK Perimeter Security", Pearson Education Asia.
3. Robert Bragge, Mark Rhodes, Heith stragggberg, "Network Security the Complete Reference", Tata McGraw Hill Publication.

WEB RESOURCES:

1. <http://nptel.iitm.ac.in/courses/106105031/>
2. <http://www.cert.org/>
3. http://www.howard.edu/csl/research_crypt.htm
4. http://www.cs.purdue.edu/homes/ninghui/courses/426_Fall10/lectures.html
5. <http://www.cs.uwp.edu/staff/lincke/infosec/>
6. <http://www.cisa.umbc.edu/courses/cmsc/426/fall06/>
7. <http://www.cs.northwestern.edu/~ychen/classes/cs395-w05/lectures.html>

ELECTIVE - II-C :

III YEAR / VI SEM

BLOCKCHAIN TECHNOLOGY

OBJECTIVES:

- To understand the concepts of block chain technology
- To understand the consensus and hyper ledger fabric in block chain technology.

OUTCOMES:

- State the basic concepts of block chain
- Paraphrase the list of consensus and Demonstrate and Interpret working of Hyper ledger Fabric

- Implement SDK composer tool and explain the Digital identity for government

UNIT - I

History: Digital Money to Distributed Ledgers -Design Primitives: Protocols, Security, Consensus, Permissions, Privacy- : Block chain Architecture and Design-Basic crypto primitives: Hash, Signature-Hash chain to Block chain-Basic consensus mechanisms.

UNIT - II

Requirements for the consensus protocols-Proof of Work (PoW)-Scalability aspects of Block chain consensus protocols: Permissioned Block chains-Design goals-Consensus protocols for Permissioned Block chains.

UNIT - III

Decomposing the consensus process-Hyper ledger fabric components-Chain code Design and Implementation: Hyper ledger Fabric II:-Beyond Chain code: fabric SDK and Front End-Hyper ledger composer tool.

UNIT - IV

Block chain in Financial Software and Systems (FSS): -Settlements, -KYC, -Capital markets-Insurance-Block chain in trade/supply chain: Provenance of goods, visibility, trade/supply chain finance, invoice management/discounting.

UNIT - V

Block chain for Government: Digital identity, land records and other kinds of record keeping between government entities, public distribution system / social welfare systems: Block chain Cryptography: Privacy and Security on Block chain.

TEXT BOOKS:

1. Mark Gates, “*Block chain: Ultimate guide to understanding block chain, bit coin, crypto currencies, smart contracts and the future of money*”, Wise Fox Publishing and Mark Gates 2017.
2. Salman Baset, Luc Desrosiers, Nitin Gaur, Petr Novotny, Anthony O'Dowd, Venkatraman Ramakrishna, “*Hands-On Block chain with Hyper ledger: Building decentralized applications with Hyperledger Fabric and Composer*”, 2018.
3. Bahga, Vijay Madisetti, “*Block chain Applications: A Hands-On Approach*”, Arshdeep Bahga, Vijay Madisetti publishers 2017.

REFERENCE BOOKS :

1. Andreas Antonopoulos, “*Mastering Bitcoin: Unlocking Digital Crypto currencies*”, O'Reilly Media, Inc. 2014.
2. Melanie Swa, “*Block chain*”, O'Reilly Media 2014.

WEB REFERENCES:

1. NPTEL & MOOC courses titled blockchain technology
2. blockgeeks.com/guide/what-is-block-chain-technology
3. <https://nptel.ac.in/courses/106105184/>

, if with in Campus, if with in Campus, if with in Campus, if with in Campus, if with in Campus, if with in Campus

MINI PROJECT

OBJECTIVES:

The aim of the mini project is that the student has to understand the real time software development environment. The student should gain a thorough knowledge of the problem he/she has selected and the language / software he/she is using.

Project planning:

B.Sc (Computer Science / Software Application)/BCA Major Project is an involved exercise, which has to be planned well in advance. The topic should be chosen in the beginning of the final year itself. Related reading training and discussions of the first internal project viva voce should be completed in the first term of the final year.

I Selection of the project work

Project work could be of three types.

a) Developing solution for real life problem

In this case a requirement for developing a computer-based solution already exists and the different stages of the system development life cycle is to be implemented successfully. Examples are accounting software for a particular organization, computerization of administrative functions of an organization, web based commerce etc.

b) System Software Project

Projects based on system level implementation. An example is a Tamil language editor with spell checker, compiler design.

b) Research level project

These are projects which involve research and development and may not be as structured and clear cut as in the above case. Examples are Tamil character recognition, neural net based speech recognizer etc. This type of project provides more challenging opportunities to students.

II Selection of team

To meet the stated objectives, it is imperative that a major project is done through a team effort. Though it would be ideal to select the team members at random and this should be strongly recommended, due to practical consideration students may also be given the choice of forming themselves into teams with three members. A team leader shall be selected. Team shall maintain the minutes of meeting of the team members and ensure that tasks have been assigned to every team member in writing. Team meeting minutes shall form a part of the project report. Even if students are doing projects as groups, each one must independently take different modules of the work and must submit the report.

III Selection of Tools

No restrictions shall be placed on the students in the choice of platform/tools/languages to be utilized for their project work, though open source is strongly recommended, wherever possible. No value shall be placed on the use of tools in the evaluation of the project.

IV Project management

The Head of the Department / Principal of the college should publish the list of the student's project topic, internal guide and external organization and teams agreed before the end of July. Changes in this list may be permitted for valid reasons and shall be considered favorably by the Head of the department / Principal of the college any time before commencement of the project. Students should submit a fortnightly report of the progress, which could be an indication of percentage of completion of the project work. The students should ideally keep a daily activity book. Team meetings should be documented and the same should be submitted at the end of the project work.

V Documentation

Three copies of the project report must be submitted by each student (one for the department library, one for the organization where the project is done and one for the student himself/herself). The final outer dimensions of the project report shall be 21cm X 30 cm. The color of the flap cover shall be light blue. Only hard binding should be done. The text of the report should be set in 12 pt, Times New Roman, 1.5 spaced.

Headings should be set as follows: CHAPTER HEADINGS 16 pt, Arial, Bold, All caps, Centered.

1. Section Headings 14 pt Bookman old style, Bold, Left adjusted.

1.1 Section Subheading 12 pt, Bookman old style.

Title of figures tables etc are done in 12 points, Times New Roman, Italics, centered.

Content of the Project should be relevant and specific particularly with reference to the work. The report should contain the requirement specification of the work, Analysis, Design, Coding, testing and Implementation strategies done.

- Organizational overview (of the client organization, where applicable)
- Description of the present system
- Limitations of the present system
- The Proposed system - Its advantages and features
- Context diagram of the proposed system
- Top level DFD of the proposed system with at least one additional level of expansion
- Program List (Sample code of major functions used)
- Files or tables (for DBMS projects) list. List of fields or attributes (for DBMS projects) in each file or table.
- Program – File table that shows the files/tables used by each program and the files are read, written to, updated, queried or reports were produced from them.
- Screen layouts for each data entry screen.
- Report formats for each report.

Some general guidelines on documentation are:

1. Certificate should be in the format: **”Certified that this report titled... is a bonafide record of the project work done by Sri/ Kum under our supervision and guidance, towards partial fulfillment of the requirement for award of the Degree of B.Sc Computer Science/B.Sc Computer Science with Artificial Intelligence /B.Sc Computer Science with Data Science /BCA of XXXXXXXX College”** with dated signature of internal guide, external guide and also Head of the Department/ College.
2. If the project is done in an external organization, another certificate on the letterhead of the organization is required: **“Certified that his/her report titledis a bonafide record of the project work done by Sri/Kum.....under my supervision and guidance, at thedepartment of..... (Organization) towards partial fulfillment of the requirement for the award of the Degree of B.Sc (Computer Science)/B.Sc Computer Science with Artificial Intelligence /B.Sc Computer Science with Data Science /BCA of XXX College.**
3. Page numbers shall be set at right hand bottom, paragraph indent shall be set as 3.
4. Only 1.5 spaces need be left above a section or subsection heading and no space may be left after them.
5. References shall be IEEE format (see any IEEE magazine for detail) While doing the project keep note of all books you refer, in the correct format and include them in alphabetical order in your reference list.

VI Project Evaluation:

Internal Assessment

There shall be six components that will be considered in assessing a project work with weightage as indicated.

1. Timely completion of assigned tasks as evidenced by team meeting minutes 20%
2. Individual involvement, teamwork and adoption of industry work culture 10%
3. Quality of project documentation (Precision, stylistics etc) 10%
4. Achievement of project deliverables 20%
- 5 Effective technical presentation of project work 10%
6. Viva 30%

Based on the above 6 components, internal marks (40) can be awarded.

External Assessment

Dissertation/Project submitted at the end of third year shall be valued by two examiners appointed by the Controller for the conduct of practical exams. The board of examiners shall award 40 marks based on the following components.

1. Achievement of project deliverables - 20 Marks
2. Effective technical presentation of project work - 20 Marks
3. Project Viva - 20 Marks

There shall be a common written examination conducted for all the candidates in each group together for a minimum of 10 minutes.

- (i) Requirement Specification of Project
- (ii) Design of Project
- (iii) Testing and Implementation of Project