# SNDT Women’s University, Mumbai

## Master of Science in Data Science

## M.Sc. (D.S.)

*as per NEP-2020*

# Syllabus

**(2025-26)**

| | |
|---|---|
| **Programme** | **Master of Science in Data Science (M.Sc.-DS)** |
| **Preamble** | In unwavering commitment to the principles outlined herein, the (M.Sc.-DS.) program steadfastly aims to shape an educational experience that empowers its students to excel as adept scholars, critical thinkers, and responsible leaders within their respective fields. By articulating these guiding principles, we underscore our dedication to fostering a transformative learning environment that goes beyond the acquisition of technical skills to cultivate holistic and forward-thinking professionals. |
| **Programme Outcomes (POs)** | After completing this programme, Learner will be able to<br>● **Advanced Knowledge-**Demonstrated proficiency in fundamental and specialized computer science concepts, encompassing algorithms, data structures, artificial intelligence, machine learning, and relevant domains.<br>● **Critical Thinking and Problem Solving**- Exhibited ability to analyze intricate problems, synthesize information, and apply critical thinking skills for the creation of innovative and effective solutions within the computer science field<br>● **Advanced Technical Skills-**Possession of advanced technical skills in programming languages, software development, system design, and other pertinent areas, enabling the creation of robust and efficient computing solutions<br>● **Ethical Considerations-**Displayed a strong understanding of ethical considerations in computer science, encompassing privacy, security, intellectual property, and societal impact. Graduates are equipped to make informed and ethical decisions in their professional practice.<br>● **Lifelong Learning-**Embraced a commitment to lifelong learning, showcasing the ability to stay current with emerging technologies, industry trends, and advancements in computer science through self- directed learning and ongoing professional development. |
| **Programme Specific Outcomes (PSOs)** | Programme Specific Outcomes (PSOs) for an (M.Sc.-DS) specify the particular skills, knowledge, and abilities that students are expected to gain upon completion of the program. |

| | |
|---|---|
| | ● **Advanced Technical Proficiency-**Demonstrate mastery of advanced concepts in computer science, including algorithms, data structures, databases, and software engineering, to design and implement complex computing solutions.<br>● **Specialized Knowledge in Focus Areas-**Develop expertise in specific focus areas within computer science such as artificial intelligence, machine learning, cyber security, or data science, showcasing advanced knowledge and skills in these specialized domains.<br>● **Research and Innovation-**Conduct independent research, including formulating research questions, designing experiments, and analyzing results, contributing to the creation of new knowledge and advancements in the field.<br>● **Advanced Programming Skills-**Exhibit proficiency in various programming languages and paradigms, enabling the development of efficient and scalable software solutions.<br>● **System Design and Architecture-**Design and architect complex computing systems, demonstrating an understanding of system-level considerations, scalability, and performance optimization.<br>● **Effective Communication of Technical Information-**Communicate complex technical information effectively to both technical and non-technical audiences through written reports, presentations, and documentation.<br>● **Innovation and Entrepreneurship-**Foster innovation and entrepreneurial thinking, demonstrating the ability to identify opportunities, propose creative solutions, and potentially contribute to startups or innovative projects. |
| Eligibility Criteria for Programme | A woman Graduate in BSc.(Physics), BSc.(Maths.), BSc.(Elect.), BSc.(IT),B.Sc.(CS) or BCA or any engineering graduate in allied subject from the recognized university with an aggregate marks not less than 50% (Open Category) and 45%(Reserved category). |
| Intake | 60 |
| Duration | 4 semesters (2 years) |

## Master of Science in Data Science (M.Sc.-DS)

### Year I

| Code | Subjects | Type of Course | Credits | Marks | Int. | Ext. |
|---|---|---|---|---|---|---|
| | **Semester-I** | | | | | |
| 115661 | Mathematics for Computer Science | Major (Core) | 4 | 100 | 50 | 50 |
| 115662 | Advance Data Structure and Algorithms | Major (Core) | 4 | 100 | 50 | 50 |
| 115663 | Data Communication and Networking | Major (Core) | 4 | 100 | 50 | 50 |
| 115604 | Object-Oriented Programming with Java | Major (Core) | 2 | 50 | 50 | 0 |
| | Elective -I | Major (Elective) | 4 | 100 | 50 | 50 |
| 135681 | Research Methodology | Minor stream | 4 | 100 | 50 | 50 |
| | | | 22 | 550 | 300 | 250 |
| | **Semester-II** | | | | | |
| **Code** | **Subjects** | **Type of Course** | **Credit** | **Marks** | **Int.** | **Ext.** |
| 215661 | Artificial Intelligence | Major (Core) | 4 | 100 | 50 | 50 |
| 215662 | Advance DBMS / Next generation databases | Major (Core) | 4 | 100 | 50 | 50 |
| 215663 | Data Science Using Pythion | Major (Core) | 4 | 100 | 50 | 50 |
| 215664 | Full Stack Development | Major (Core) | 4 | 100 | 50 | 50 |
| | Elective - II | Major (Elective) | 4 | 100 | 50 | 50 |
| 145681 | OJT | OJT | 2 | 50 | 0 | 50 |
| | | | 22 | 550 | 250 | 300 |

Exit option(44 credits):
Post Graduate Diploma in Computer Science

## Year II

| Code | Subjects | Type of Course | Credit | Marks | Int. | Ext. |
|---|---|---|---|---|---|---|
| | **Semester-III** | | | | | |
| 315661 | Cloud Computing (Web Services) | Major (Core) | 4 | 100 | 50 | 50 |
| 315662 | Big Data Analytics | Major (Core) | 4 | 100 | 50 | 50 |
| 315663 | Machine Learning | Major (Core) | 4 | 100 | 50 | 50 |
| 315604 | Natural Language Processing | Major (Core) | 2 | 50 | 50 | 0 |
| | Elective - III | Major (Elective) | 4 | 100 | 50 | 50 |
| 355681 | Research Proposal | RP | 4 | 100 | 50 | 50 |
| | | | 22 | 550 | 300 | 250 |
| | **Semester-IV** | | | | | |
| | Swayam / MOOC / Elective | Major (Elective) | 4 | 100 | 50 | 50 |
| 415661 | Interactive & Advanced Data Visualization (with Python) | Major (Core) | 4 | 100 | 50 | 50 |
| 415662 | Advanced Natural Language Processing | Major (Core) | 4 | 100 | 50 | 50 |
| 415663 | Deep learning | Major (Core) | 4 | 100 | 50 | 50 |
| 455681 | Research Dissertation | Internship | 6 | 150 | 50 | 100 |
| | | | 22 | 550 | 250 | 300 |

| Code | Elective-I | Code | Elective-II |
|---|---|---|---|
| 125661 | 1. Cyber Security | 225661 | 1. Software Testing and Quality Assurance |
| 125662 | 2. Software Engineering and Project Management | 225662 | 2. Fuzzy Logic & Neural networks |
| 125663 | 3. Theory of Computation and Applications | 225663 | 1. Ethical Hacking |
| 125664 | 4. UI/UX Design | 225664 | 4. Natural Language Processing |

| Code | Elective-III | Code | Elective-IV |
|---|---|---|---|
| 325661 | 1. Generative AI and LLMs | 425661 | 1. Applied Data Analysis (XAI) |
| 325662 | 2. Graph Neural Network (GNN) | 425662 | 2. Social Network Analysis |
| 325663 | 3. Image and Video Analytics | 425663 | 3. Explainable Artificial Intelligence (XAI) |
| 325664 | 4. Mobile App Development Technologies | 425664 | 4. Internet of Things (IOT) |

M.Sc. (Data Science) Semester I(NEP pattern)

| | |
|---|---|
| **Course Title** | **Mathematics for Computer Science** |
| **Course Credits** | **4 Credits** |
| **Course Outcomes** | **CO1:** To Study Set and Operations. |
| | **CO2:** To study logic and Boolean Algebra |
| | **CO3**: To enable the students to use nonlinear and linear methods to analyze and understand problems in the same |
| | **CO4:** To demonstrate how the mathematics help to develop thinking ability |
| | **CO5:** To demonstrate how the graph and logic helps to develop thinking ability |
| **Module 1 (Credit 1) Set Theory** | |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | To Understand the set theory and operations<br>To facilitate applications of set |
| **Course Outline** | Set definition, set builder form. Operations on set, Venn diagram , DeMorgan's Law |
| **Module 2 (Credit 1) Logic and Boolean Algebra** | |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Logic operations |
| | Boolean operations |
| **Course Outline** | Logic gates, truth tables, Boolean expression, Laws |
| **Module 3 (Credit 1) Solution to nonlinear equation** | |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | To Understand the methods of nonlinear equation |
| **Course Outline** | **Solution to nonlinear equation:**<br>Bisection method,<br>Newton Raphson Method,<br>Regula Falsi method |
| **Module 4 (Credit 1) Graph** | |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Understand Basics of Graph |
| **Course Outline** | Graph Definition basic terminology, Matrix representation of graphs, Types of graphs, Hand Shaking Lemma |

**Reference Books:**

(1) S.S. Sastry; Introductory Methods of Numerical Analysis, 3rd edition, Prentice Hall of India,
(2) H.C. Saxena; Finite differences and Numerical Analysis, S. Chand and Company.
(3) Kolman and Ross - Discrete Mathematical Structures
(4) B.S. Vatssa:-Discrete Mathematics
(5) Set Theory: A First Course by Daniel W. Cunningham:

## Assessment:

**Internal Assessment – 50 Marks**

| **The Rubric will have the following Evaluation Parameters:** | | |
|---|---|---|
| **Evaluation Parameters** | **Description / Evaluation Points** | **Marks** |
| Interpretation | Accuracy in interpreting the example | 15 |
| Representation | Translating problems | 15 |
| Calculation | To perform correct numerical calculations | 15 |
| Logic Development | To enhance logical thinking | 5 |

Sample Projects

Project 1: Find real world applications of graphs
Project 2 :Find real world the applications of logic
Project 3: Prepare the model for Set operations
Project 4: Write program for non linear equations

**External Assessment: (Marks 50)**

End Semester examination of 50 marks for 2 hours duration will be conducted.

| | |
|---|---|
| **Course Title** | **Advance Data Structure and Algorithms**<br>**Major (Core) Theory** |
| **Course Credits** | **4 Credits** |
| **Course Outcomes** | **CO1: Understand** and apply linear data structures-List |
| | **CO2:** Understand and apply nonlinear data structures graphs and trees. |
| | **CO3**: Analyze the time and space complexities of basic data structure operations. |
| | **CO4:** Apply advanced data structures to solve specific computational problems. |
| **Module 1 (Credit 1)** | **Analysis of Algorithm** |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Analyze the time and space complexities of algorithms. · |
| | Apply Big-O notation to express the upper bounds of algorithmic performance. |

| | |
|---|---|
| | Evaluate the efficiency and suitability of data structures in different scenarios. |
| **Course Outline** | **Introduction:**<br>ADT- stack and queue, data structure: Definition & classification<br>**Space and time complexity** |

| | |
|---|---|
| | Graphical understanding of the relation between different functions of n, examples of linear loop, logarithmic, quadratic loop etc.<br>Analysis of algorithms with emphasis on best case, average case and worst case.<br>Best, Worst, Average case analysis, Asymptotic notations (Big O, Omega Ω, Theta Θ**)** |
| **Module 2 (Credit 1)** | **Linear Data structures** |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Identify and classify linear data structures. |
| | Analyze the advantages and limitations of each linear data structure. |
| **Course Outline** | **Stack and queue:** Introduction,<br>implementation using array.<br>**Linked Lists-** insert, delete and update<br>Singly Linked List- insert, delete and update<br>Circularly Linked List- insert, delete and update<br>Doubly Linked lists- insert, delete and update<br>Linked list - Polynomial Manipulation. |
| **Module 3 (Credit 1)** | **Non-Linear Data Structures -Tree** |
| **Learning Outcomes** | Understand tree structures, including binary trees and n-ary trees. |
| | Representation of binary Tree |
| | Implement tree traversal algorithms (e.g., in-order, pre-order, post-order). |
| | Working of a balanced tree. |

| | |
|---|---|
| | **Types of Binary trees**<br>Binary tree, skewed tree, strictly binary tree, full binary tree, complete binary tree, expression tree, binary search tree.<br>**Representation of binary trees.**<br>Implementation and Operations on Binary Search Tree - Create, Insert, Delete, Search<br>**Tree traversals**– preorder, inorder, postorder ( recursive implementation), Level-order traversal using queue<br>Max-Heap, Min-Heap<br>Huffman Algorithm Binary search tree<br>**Balanced Trees**<br>AVL Tree- concept and rotations<br>Red Black trees -, insertion and deletion(concept)<br>Multi-way search tree - B and B+ tree - Insertion, Deletion (concept) |

| **Module 4 (Credit 1)** | **Hashing** |
|---|---|
| **Learning Outcomes** | Explain the concept of hashing and hash functions. |
| | Implement and analyze hash tables for efficient data retrieval |
| | **Hash Table**<br>Concept of hashing: Terminologies<br>Hash table, Hash function, Bucket, Hash address, collision, overflow etc.<br>Properties of good hash function<br>**Collision resolution techniques**<br>Open Addressing - Linear probing, quadratic probing, rehashing<br>Chaining - Coalesced, separate chaining |

**References:**

1. Weiss, M. A. (2003). Data Structures and Algorithm Analysis in C (2nd ed.). Pearson Education.
2. Pai, G. A. V. (2008). Data Structures and Algorithms: Concepts, Techniques, and Applications (1st ed.).
3. Horowitz, E., Sahni, S., & Anderson-Freed, S. (2007). Fundamentals of Data Structures in C (2nd ed.). University Press.
4. Tremblay, J.-P., & Sorenson, P. G. (2007). An Introduction to Data Structures with Applications (2nd ed.). Tata McGraw-Hill.
5. SNDTWU Faculty of Science and Technology: M.Sc.-CS. Syllabus 2023-24
6. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2003). Introduction to Algorithms (2nd ed.). PHI.
7. Dave, P., & Dave, H. (2008). Design and Analysis of algorithms. Pearson Education.

8. Tanenbaum, A. S. (PHI). Data Structures Using C & C++.
9. Goodrich, M., &Tamassia, R. (Wiley). Algorithm Design: Foundation, Analysis & Internet Examples.
10. Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1983). Data Structures & Algorithms. Addison-Wesley Publishing.
11. Lipschutz, S. (2017). Data Structures with C. McGraw-Hill.
    12.Kanetkar, Y. (BPB publications). Data Structures Through
    C.

## Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

## Module 1

**Comparative Complexity Study**

- Compare two algorithms for the same problem
- Analyze based on: Time Complexity, Space Complexity
- Practical performance for different input sizes.

## Module 2

**Stack and Queue Using Arrays:**

**Description**: Implement stack and queue data structures using arrays.

**Objective**: Understand LIFO (stack) and FIFO (queue) behavior.

**Evaluation**: Functional implementation, correct use of push/pop or enqueue/dequeue.

## Group Activity: Linked List Concept Mapping

**Description**: In small groups, draw and explain how different types of linked lists work (singly, doubly, circular).

**Objective**: Develop conceptual understanding through peer learning.

**Evaluation**: Diagram accuracy, explanation clarity, teamwork.

## Module 3

**Assignment: AVL Tree Rotations**

**Description**: Write code or explain the different types of AVL rotations (LL, RR, LR, RL).

**Objective**: Understand self-balancing trees and the impact of imbalance.

**Evaluation Focus**: Rotation logic, balance factor calculation, height updates.

## Module 4

**Assignment: Hash Function Design**

**Description**: Design simple hash functions for different key types (integers, strings) and analyze their effectiveness.

**Objective**: Understand the role of a good hash function and collision minimization.

**Evaluation Focus**: Creativity, correctness, and justification of design.

**Assessment**

## Internal Assessment – 50 Marks

**Evaluation Scheme: -** Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

| S. No. | Component | Criteria | Marks |
|---|---|---|---|
| 1 | **Assignments / Lab Work** | Correctness and completeness of implementation | 5 |
| | | Use of appropriate advanced data structures | 3 |

|  |  |  |  |
|---|---|---|---|
|  |  | Code efficiency and readability | 2 |
|  |  | **Subtotal** | **10** |
| **2** | **Internal Test / Viva** | Understanding of concepts and logic | 5 |
|  |  | Problem-solving approach | 3 |
|  |  | Communication during viva or written clarity | 2 |
|  |  | **Subtotal** | **10** |
| **3** | **Mini Project / Lab Task** | Relevance of problem and solution design | 5 |
|  |  | Use of appropriate advanced data structures (trees,list, graphs) | 5 |
|  |  | Functionality and correctness | 3 |
|  |  | **Subtotal** | **13** |
| **4** | **Seminar / Presentation** | Clarity of explanation and content organization | 4 |
|  |  | Depth of technical understanding | 3 |
|  |  | Presentation skills (PPT/diagrams/Q&A) | 3 |
|  |  | **Subtotal** | **10** |
| **5** | **Attendance & Participation** | Attendance, punctuality, and class engagement | 2 |
|  |  | **Subtotal** | **2** |
|  |  | **Total Internal Marks** | **50** |

## External Assessment: (Marks 50)

End Semester examination of 50 marks for 2 hours duration will be conducted.

| **Course Title** | **Data Communication and Computer Networks** |
|---|---|
| **Course Credits** | **4 Credits** |
| **Course Outcomes** | **CO1:** Define and explain key concepts in data communications. |
| | **CO2:** Understand the OSI (Open Systems Interconnection) model and TCP/IP protocol suite. |
| | **CO3:** Describe and differentiate between various networking protocols. |
| | **CO4:** Analyze the functions and characteristics of key protocols such as TCP, UDP, IP, and HTTP. |
| | **CO5:** Design and implement basic computer networks based on specific requirements**.** |
| | **CO6:** Evaluate and select appropriate networking topologies and configurations. |
| **Module 1 (Credit 1)** | **Introduction to Computer Networks** |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Understand the concept of computer networks and their significance in modern computing. |
| | Explain the purposes and advantages of connecting computers in a networked environment. |
| | Identify and differentiate between common network topologies. |
| | Analyze the strengths and weaknesses of various topologies in different scenarios. |
| | Explore various applications of computer networks in different domains (e.g., business, education, healthcare). |
| | Analyze case studies to understand how networks are utilized to meet specific organizational needs. |

| | |
|---|---|
| **Content Outline** | **Introduction:**<br>Computer Networks and its uses, Network categorization and Hardware: Broadcast and point-to-point networks, Local Area Network (LAN), Metropolitan Area Network (MAN), Wide Area Networks (WAN), Inter networks, Topologies, Wireless Networks, Network Software: Protocols, Services, network architecture, design issues, OSI Reference model, TCP/IP Reference model, Comparison of OSI and TCP/IP Models. Introduction to Example Networks: Internet, Connection-Oriented Networks–X.25, Frame Relay, ATM<br>**Data Communication Model:**<br>Digital and Analog data and signals, bit rate, baud, bandwidth, Nyquist bit rate, Guided Transmission Media – Twisted Pair, Coaxial cable, Optical fiber; wireless transmission–Radio waves, microwaves, infrared waves; Satellite Communication. |
| **Module 2 (Credit 1)** | **Switching, Error Detection and Correction** |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Explain the concept of circuit switching in telecommunication networks. |
| | Differentiate circuit switching from other switching techniques. |
| | Describe the process of establishing a circuit in a circuit-switched network. |
| | Describe how packets are routed and forwarded in a packet-switched network. |
| | Explore routing algorithms used in packet switching. |
| **Content Outline** | **Switching:**<br>Circuit Switching, Packet switching; Multiplexing: Frequency Division Multiplexing, Time Division Multiplexing, Synchronous and Asynchronous TDM, Modems, Transmission impairments, Manchester and differential Manchester encoding<br>**Error Detection and Correction:**<br>Types of errors Redundancy, Detection Versus Correction, Error Detection, Error Correction, Hamming Code, Cyclic Redundancy Check, Check sum and Its idea. |
| **Module 3 (Credit 1)** | **Data Link Layer Design issues** |

|  |  |
|---|---|
| **Learning Outcomes** | **After learning the module, learners will be able to** |
|  | Explain the role of the Data Link Layer in the OSI model. |
|  | Differentiate between the functions of the Physical Layer and the Data Link Layer. |
|  | Discuss various framing techniques used in the Data Link Layer. |
|  | Implement framing algorithms for efficient data encapsulation and transmission |
| **Content Outline** | Data Link Layer Design issues: Framing, error control, Flow Control, Error Detection and correction; Elementary Data Link Protocols, Sliding Windows Protocols; Medium Access Control: Aloha, CSMA protocols, Collision free protocols, Limited Contention Protocols; Wave length division Multiple access protocol, Wireless LAN Protocol: MACA; IEEE 802.3Ethernet, IEEE 802.4 Token Bus; IEEE 802.5 Token ring, Binary Exponential Back off algorithm, Digital Cellular, Radio : Global System for Mobile, Communication (GSM), Code Division Multiple Access (CDMA) |
| **Module 4 (Credit 1)** | **Network Layer, Design issues** |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
|  | Explain the purpose and functions of the Network Layer in the OSI model. |
|  | Differentiate between the responsibilities of the Network Layer and other layers. |
|  | Understand the concepts of addressing and routing at the Network Layer. |
|  | Design and implement addressing schemes for efficient packet routing. |
|  | Define virtual circuit switching and its advantages. |
|  | Compare and contrast virtual circuit switching with other switching techniques. |
| **Content Outline** | Network Layer, Design issues Virtual circuit and Datagram Subnet, Routing Algorithms, Optimality principle, shortest path routing, Flooding, Distance Vector Routing, Link State Routing, Hierarchical |

| | |
|---|---|
| | Routing, Broadcast and Multi Cast Routing, Routing for Mobile hosts, Routing in Ad hoc Networks, congestion Control Algorithm, General Principals Traffic Shaping, Leaky Bucket, Token Bucket, choke packets, Load Shedding |

**Reference Books**

1. Forouzan, B. A. (2007). Data Communications and Networking (4th ed.). McGraw Hill. ISBN: 0-07-296775-7.
2. Stallings, W. (2013). Data and Computer Communications (10th ed.). Pearson.
3. Tanenbaum, A. S. (2010). Computer Networks (5th ed.). Pearson.

## Classroom Activities

## Activities towards Comprehensive Continuous Evaluation (CCE)

### Module 1:

1. Define a computer network and explain the applications of computer networks.
2. Explain LAN, MAN, WAN.
3. What is the OSI model? Explain the functions of each layer in the OSI model.
4. Explain the network topologies.
5. Describe the guided and unguided transmission media.

### Module 2:

1. Differentiate between circuit switching and packet switching.
2. What is multiplexing? Explain various types of multiplexing.
3. Explain Manchester and differential Manchester encoding.
4. Describe Hamming code and CRC.

### Module 3:

1. What are the design issues of the data link layer?
2. Explain the sliding window protocol.
3. Describe the IEEE token ring and token bus protocols.
4. Explain the concept of GSM.

### Module 4:

1. What are the design issues of the network layer?
2. Explain the link state routing protocol.
3. Describe the distance vector routing protocol.
4. Explain the various congestion control techniques.

**Assessment**

## Internal Assessment – 50 Marks

Evaluation Scheme:

| Component | Weight | Typical Criteria |
|---|---|---|
| Mid-term Exam (CIA I) | ~25% | Coverage of core theory, routing, protocols, addressing |
| Assignments & Quizzes (CIA II & III) | ~10–20% | Case studies, seminars, problem-solving |
| Lab Reports | ~10% | Clarity, accuracy, visuals, completeness, timely submission |
| Group Project / Final Report | ~20% | Design, topology, addressing, cost, analysis, presentation |
| Attendance & Participation | ~5% | Active engagement throughout the course |

**External Assessment: (Marks 50)**

End Semester examination of 50 marks for 2 hours duration will be conducted.

| | |
|---|---|
| **Course Title** | **Object Oriented Programming with Java** |
| **Course Credits** | **2 Credits** |
| | **CO1: To** introduce the object-oriented programming system concepts |
| | **CO2:** To develop modular programs using Java |
| | **CO3**: To setup JDK environment to create, debug and run Java programs |
| | **CO4:** To introduce syntax and semantics of Java programming language as well as JDBC |
| **Module 1 (Credit 1) Introduction to Java Programming** | |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Apply Object-Oriented Programming Concepts |
| | Demonstrate Proficiency with Java Syntax and Language Fundamentals |
| | Implement Control Flow and Decision-Making in Java Programs |
| **Course Outline** | **Introduction to Java:** Basics of Java programming, Datatypes, Variables, Operators, Control structures including selection, Looping, Java methods, Overloading, Math class, Arrays in java.<br>**Inheritance in java:** Super and sub class, Overriding, Object class, Polymorphism, Dynamic binding, Generic programming, Casting objects, Instance of operator, Abstract class, Interface in java, Package in java, UTIL package, Multithreading, Exception Handling<br>Introduction to JDBC: What is JDBC.<br>Database connectivity, JDBC Architecture, JDBC drivers, Using JDBC API – Loading a Driver, connecting and executing JDBC statement, Handling SQL Exceptions. Accessing Result Sets, method of Result Set interface, Methods of Prepared Statement interface, retrieving row, inserting row, Managing Database Transactions, creating and calling stored procedures in JDBC, using Metadata in JDBC. |
| **Module 2 (Credit 1) JSP, Servlet, Hibernet, Spring MVC** | |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | 1)Understanding of advanced web concepts associated with JAVA. |

| Course Outline | **Introduction to servlets:** Servlet vs CGI, Servlet API overview, Servlet Life cycle, Generic servlet, HTTP Servlet, ServletConfig, Servlet Context, Handling HTTP Request and response –GET / POST method, request dispatching, Using cookies, Session tracking.<br>**Web development using JSP:** Introduction to JSP, JSP Architecture, JSP Directives, JSP scripting elements, Default objects in JSP, JSP Actions, JSP with beans and JSP with Database, Error handling in JSP, tracking techniques in JSP, Introduction to custom tags, JSTL tags in detail. Introduction to jQuery, JS, JS JSON, jQuery vs JS.<br>**Introduction to Spring Framework:** Spring Architecture, Spring Aspect of Object-Oriented Concepts – Join Point and Point Cuts. Spring web applications with Spring MVC. Features of the Spring Boot. Use of Spring Boot to create and configure a Spring application.<br>Customize Spring Boot features. REST web services with Spring. Spring Data Secure with Spring Security. JMS- Introduction, requirement, JMS Programming model. JMS support of Spring.<br>**Introduction to Hibernate:** Problem with JDBC - paradigm mismatch, ORM. Different components of Hibernate, How to persist objects using Hibernate, mapping files in hibernate, configuration files and Session object, Instance states, Implementing Inheritance in Hibernate, Transactions in Hibernate, Querying with HQL (Hibernate Query Language) |
|---|---|

**Reference Books**

1. The Complete Reference, Third Edition, by Patrick Naughton and Herbert Schildt, Tata McGraw Hill Edition 1999.
2. Java Enterprise in a Nutshell: A Desktop Quick Reference (Nutshell Handbook) or any other

book with similar contents.
3. Mastering Java2 J2SE1.4 by John Zukouski PBP Publication
4. JavaTM How to Program Sixth Edition by H.M Deitel, P.J. Deitel
5. Core Servlets & JavaServer Pages by Marty Hall, Larry Brown
6. Spring Boot in Action 1st Edition by Craig Walls
7. Beginning Hibernate Second Edition by Jeff Linwood, Dave Minter is the third book for Hibernate beginners.

## Assessment Internal-50 Marks

Internal Assessment to be done on the basis of Project (20M), Unit test (20M) and assignment

## Project 1

Using jdk design a Bank Account System which allows users to
create accounts, deposit money, withdraw money, and check balances, Examples of JDBC

## Project 2

Assignments of-Servlet, JSP, JMS Support of Spring, Introduction to Hibernet and Querying with HQL.

| **The Rubric will have the following Evaluation Parameters:** | | |
|---|---|---|
| **Evaluation Parameters** | **Description / Evaluation Points** | **Marks** |
| Code Implementation | ● Accuracy in writing and executing the Java program, correct logic, syntax | 15 |
| Problem Understanding | ● Clarity of problem interpretation and correct approach | 10 |
| Output Correctness | ● Program compiles without errors and gives correct output | 10 |
| Code Quality & Structure | ● Proper use of indentation, comments, meaningful variable names, modularity | 5 |
| Debugging Skills | ● Ability to identify and fix logical/syntax errors during execution | 5 |
| Lab Record/Journal | ● Completeness, correctness, formatting, and timely submission | 5 |

## Major (Elective - I)

| | |
|---|---|
| **Course Title** | **Cyber Security (Elective-I) Theory** |
| **Course Credits** | **4 Credits** |
| | **CO1: Provide** an overview of the field of Cyber Security,<br>including its challenges, constraints, and the role of Internet governance. |
| | **CO2:** Understand the imperative for a comprehensive Cyber Security policy, the establishment of a nodal authority, and the importance of an international convention on Cyberspace. |
| | **CO3**: Identify vulnerabilities in software, system administration, network architectures, data access, authentication, broadband communications, and poor awareness. |
| | **CO4:** Identify intrusion types, such as physical theft, privilege abuse, unauthorized access, malware infection, and implement techniques including anti-malware software, network-based intrusion detection/prevention systems with its types. |
| **Module 1 (Credit 1) Introduction to Cyber Security** | |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Demonstrate the ability to conduct security audits, identifying potential, weaknesses in systems and networks. |
| | Explain the role of cryptography in Cyber Security and apply cryptographic techniques to secure data communication. |
| | Understand the concept of ethical hacking and its role in proactively identifying and addressing vulnerabilities. |
| **Course Outline** | **Introduction to Cyber Security**<br>Overview of Cyber Security,<br>Internet Governance–Challenges and Constraints, Cyber Threats: - Cyber Warfare-Cyber Crime-Cyber, terrorism-Cyber Espionage, Need for a Comprehensive Cyber Security Policy, Need for a Nodal Authority, Need for an International<br>convention on Cyberspace<br><br>**Cyber Security Vulnerabilities and Safeguards**<br>Cyber Security Vulnerabilities Overview, vulnerabilities in<br>software, System administration, Complex Network Architectures, Open Access to Organizational Data, Weak Authentication,<br>Unprotected Broadband |

| | |
|---|---|
| | communications, Poor Cyber Security Awareness. Cyber Security Safeguards<br>Overview, Access control, Audit, Authentication, Biometrics, Cryptography, Deception, Denial of Service Filters,<br>Ethical Hacking, Firewalls, Intrusion Detection Systems, Response, Scanning, Security policy, Threat Management. |
| **Module 2 (Credit 1) Securing Web Application** | |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Define and explain the roles of services and servers in the context of web applications. |
| | Understand identity management principles and implement secure identity practices within web services. |
| | Apply security measures to prevent and detect unauthorized access attempts by external entities. |
| **Course Outline** | **Securing Web Application**<br>Services and Servers Introduction, Basic security for HTTP Applications and Services, Basic Security for SOAP Services, Identity Management and Web Services, Authorization Patterns,<br>Security Considerations,<br>Challenges.<br><br>**Intrusion Detection and Prevention**<br>Intrusion, Physical Theft, Abuse of Privileges, Unauthorized Access by Outsider, Malware infection, Intrusion detection and Prevention Techniques,<br>Anti-Malware software, Network based Intrusion detection Systems, Network based Intrusion Prevention Systems, Host based Intrusion prevention Systems, Security Information Management, Network Session Analysis, System Integrity Validation |
| **Module 3 (Credit 1) Cryptography and Network Security** | |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Define cryptography and explain its role in securing information and communication. |
| | Differentiate between symmetric and asymmetric key cryptography, and understand their applications in securing data. |
| | Understand and apply cryptography in various applications, demonstrating proficiency in securing data in different contexts. |
| **Course Outline** | **Cryptography and Network Security:** |

| | |
|---|---|
| | Introduction to Cryptography, Symmetric key Cryptography, Asymmetric key Cryptography, Message Authentication, Digital Signatures, Applications of Cryptography. Overview of Firewalls- Types of Firewalls, User Management, VPN Security, Security Protocols:-security at the Application Layer-PGP and S/MIME, Security at Transport Layer-SSL And TLS, Security at Network Layer-IPSec. |
| **Module 4 (Credit 1) Cyber jurisprudence & Forensic** | |
| **Learning Outcomes** | Analyze the specificities of the Indian cyber space, including its regulatory framework, challenges, and initiatives. |
| | Define Cyber Forensics and understand its significance in investigating cybercrimes and digital incidents. |
| | Demonstrate the ability to initiate and conduct preliminary investigations in response to suspected cyber incidents, ensuring the preservation of digital evidence. |
| **Course Outline** | **Cyberspace and the Law:**<br>Introduction, Cyber Security Regulations, Roles of International Law, the state and Private Sector in Cyber space, Cyber Security Standards. The INDIAN Cyber space, National Cyber Security Policy 2013.<br>**Cyber Forensics**<br>Introduction to Cyber Forensics, Handling Preliminary Investigations, Controlling an Investigation, conducting disk-based analysis, Investigating Information-hiding, Scrutinizing E-mail, Validating E-mail header information, Tracing Internet access, Tracing memory in real-time. |
| **Activities to be done in the class towards Comprehensive Continuous Evaluation** | |
| 1. These assignments and activities are designed to engage students in practical applications of cyber security concepts, fostering critical thinking and skill development throughout the course.<br>2. Divide students into groups and assign each group a specific cyber security vulnerability (e.g., software vulnerabilities, weak authentication). Have them research, analyze, and present strategies to mitigate the assigned vulnerability.<br>3. Organize a workshop where students present and demonstrate various cyber security safeguards. This can include access control, encryption, firewalls, and intrusion detection systems. Encourage hands-on activities and practical demonstrations. | |

4. Assign students a case study involving a web application. They should conduct a security assessment, identify vulnerabilities, and propose safeguards. Emphasize securing HTTP and SOAP services, identity management, and authorization patterns.

5. Provide case studies related to intrusion incidents. Students should analyze each case, identify the type of intrusion, and propose effective prevention and detection techniques. Encourage discussion on ethical hacking and security policy enforcement.

6. Provide legal cases related to cyber space and cyber security. Students should analyze the legal implications, court decisions, and the role of international law and regulations.

7. Assign students to review and critique the National Cyber Security Policy of 2013. They should assess its effectiveness, identify areas for improvement, and propose updated recommendations.

8. Encourage students to write research papers on cyber security and spreading awareness among society.

**Bibliography:**

1. Cybersecurity Essentials Charles J. Brooks et al.,Publisher: Wiley, 2022.
2. Cryptography and Network Security Atul Kahate ,Publisher: McGraw Hill Education.
3. Cybersecurity in Context: Technology, Policy, and Law Chris Jay Hoofnagle & Golden G. Richard III ,Publisher: Wiley, 2024 1st Edition.
4. Digital Forensics and Incident Response Gerard Johansen,Publisher: Packt Publishing, 3rd Edition, 2023.
5. Digital Privacy and Security Using Windows: A Practical Guide Nihad Hassan,Rami Hijazi, Apress.
6. Securing the Future: The Role of AI in Cybersecurity Nimeshkumar Patel.

**Assessment**

Internal Assessment – 50 Marks

**Evaluation Scheme:**

Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

| **The Rubric will have the following Evaluation Parameters:** | | |
|---|---|---|
| **Evaluation Parameters** | **Description / Evaluation Points** | **Marks** |
| Understanding of Cyber Security Fundamentals | Demonstrates in-depth understanding of cyber threats, governance, policies, and global conventions. | 10 |

| | | |
|---|---|---|
| Securing Web Applications | Implements identity/access management principles; prevents unauthorized access efficiently. | 10 |
| Problem Solving, Ethical Hacking, IDS/IPS Implementation | Applies ethical hacking tools & techniques proficiently; accurately implements IDS/IPS with justifications. | 10 |
| Application of Cryptographic Techniques | Effectively applies cryptographic algorithms and differentiates between symmetric/asymmetric cryptography. | 10 |
| Cyber Forensics and Legal Awareness | Demonstrates ability to analyze digital evidence and explain Indian and global cyber laws precisely. | 10 |

Given below are two sample case study, but it is expected to work on similar sort of Case study

**Case study 1:-**The Data Breach at Any Bank,Identify the cyber vulnerabilities that led to the breach.Discuss what kind of Intrusion Detection/Prevention Systems should have been used. Propose a Cyber Security policy and technical safeguards to prevent such attacks.

**Case study 2:-**E-commerce Website Under DDoS AttackExplain the working of a DDoS attack and how it affects server availability.Suggest network-level and application-level safeguards to mitigate DDoS risks.Recommend changes in server architecture to prevent similar issues in the future.

External Assessment: (Marks 50)

End Semester examination of 50 marks for 2 hours duration will be conducted by University

Major (Elective)

| | |
|---|---|
| **Course Title** | **Software Engineering and Project Management** |
| **Course Credits** | **4 Credits** |
| | **CO1**- Understand structured development methodologies and various models like agile or waterfall. |
| | **CO2**- Recognize the pivotal role of Software Requirements Specification (SRS) in documenting software needs. |
| | **CO3** -Estimate costs, create timelines, allocate resources efficiently, implement quality assurance, and manage risks. |

| | |
|---|---|
| | **CO4**- Adhere to effective coding, thorough verification, and engage in testing methodologies. |
| | **CO5** - Demonstrate knowledge beyond development, covering maintenance, risk management, and project management concepts. |
| **Module 1 (Credit 1)**<br>**Software Processes, Software Requirement Analysis and Specification** | |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | 1. Understand the concepts of software processes, projects, and products. |
| | 2. Examine component software processes and their roles in the development lifecycle. |
| | 3. Investigate the software configuration management process and its importance. |
| | 4. Identify the characteristics of a software process and how they influence project outcomes. |
| | 5. Understand other modeling approaches, such as prototyping, and their relevance to requirement analysis. |
| | 6. Define software requirements and recognize the need for Software Requirement Specification (SRS). |
| **Content Outline** | ● **Software Processes:**<br>Processes projects and products, Component software processes, characteristics of a software process, software Development Process, project management process, software configuration management process, software configuration management process, and process management process<br>● **Software requirement Analysis and Specification:**<br>Software requirement, need for SRS, requirement process, problem analysis, analysis issues. Informal approach, structured analysis, object-oriented modelling, other modelling approaches, prototyping, |
| **Module 2 (Credit 1)**<br>**Planning Software Project and Coding** | |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Explore verification techniques in coding and Identify size measures in the context of coding and programming. |
| | Understand the fundamentals of software testing. |
| | Explore code walk-throughs and inspections in the testing process. |

| | |
|---|---|
| | Understand different testing strategies and the associated issues. |
| **Content Outline** | **Module Contents:**<br>**Planning Software Project:**<br>Cost estimation, uncertainties in cost estimation, building cost estimation models, on size estimation, COCOMO model, project scheduling, average duration estimation, project scheduling and milestones, staffing and personnel planning, ray leigh curve, personnel plan, team structure, software configuration management plans, quality assurance plans, verification and validation, project monitoring plans, risk management.<br>**Coding:**<br>Programming practice, verification, size measures, complexity analysis, coding standards. Testing– fundamentals, white box testing, control structure testing, black box testing, basis path testing, code walk-throughs and inspection, testing strategies-Issues, Unit testing, integration testing, Validation testing, System testing |
| **Module 3 (Credit 1)**<br>**Maintenance** | |
| **Learning Outcomes** | **After learning the module, learners will be able to**<br>Understand the overall process of software maintenance and its significance in the software development lifecycle. |
| | Identify and differentiate between various types of maintenance activities. |
| | Define software risks and recognize their impact on software projects. |
| | Understand the fundamental concepts of project management |
| **Content Outline** | **Maintenance:**<br>Overview of maintenance process, types of maintenance.<br>**Risk management:** software risks-risk identification-risk monitoring and management. Project Management concept: People–Product-Process-Project. |
| **Module 4 (Credit 1)**<br>**Protection and Security** | |
| **Learning Outcomes** | **After learning the module, learners will be able to**<br>Understand the basic concepts of project scheduling and tracking in software development. |
| | Understand the basics of software configuration management (SCM) |

| | |
|---|---|
| | Understand the basics of CASE tools and their role in the software development lifecycle. |
| | Learn about the building blocks of CASE tools and how they contribute to the development process |
| **Content Outline** | **Project scheduling and tracking:**<br>Basic concepts-relation between people and effort-defining task set for the software project-selecting, software engineering task<br>**Software configuration management:**<br>Basics and standards User interface design-rules. Computer aided software engineering tools-CASE building blocks, taxonomy of CASE tools, integrated CASE environment. |

**Assignments towards Comprehensive Continuous Evaluation**

**Module 1:**

1. What are some common software process models?
2. What is the difference between traditional and agile processes in handling uncertainty?
3. How can you customize the development of a software process?
4. What are functional and non-functional requirements?
5. Choose a software project scenario and develop a project schedule, considering task dependencies and the allocation of resources. Discuss the challenges and benefits of the chosen scheduling approach.

**Module 2:**

1. What is the project's goal and purpose?
2. How will communication be managed between stakeholders and the development team?
3. What is the overall architecture of the software?
4. What coding standards and best practices will be followed?
5. What is the process for handling bug fixes and updates after the initial release?
6. How will the team handle changes to requirements during development?

**Module 3:**

1. What is the difference between preventive and predictive maintenance?
2. How is data analysis used to improve maintenance effectiveness and predict potential issues?
3. What is the role of technology in supporting maintenance and risk management?
4. How are risks assessed and prioritized?
5. What is the process for reviewing and updating risk assessments?

**Module 4:**

1. What are the project milestones?
2. What are the project dependencies?
3. What is a Work Breakdown Structure (WBS)?
4. What is critical path analysis?
5. What is Software Configuration Management (SCM)?
6. What is the difference between SCM and Version Control?
7. Why is SCM important in software development?

**Reference Books :**

1. Pressman, Roger. (Year). "Software Engineering: A Practitioner's Approach." Tata McGrawHill, New Delhi.
2. Jalote, Pankaj. (Year)."An Integrated Approach to Software Engineering."Narosa, New Delhi.
3. Fairley, R. E. (1985). "Software Engineering Concepts." McGraw-Hill, Inc. Poyce.(Year)."Software Project Management."Addison-Wesley.
4. Sommerville.(Year)."Software Engineering."Addison-Wesley.

**Assessment:**
**Internal Assessment – 50 Marks**

**The Rubric will have the following Evaluation Parameters:**

| Criteria | Description | Marks | Assessment Type |
|---|---|---|---|
| 1. Understanding of Concepts | Clarity in core software engineering principles and project management fundamentals (SDLC, models, tools, planning, risk management) | 10 | Written exam/quiz/assignment |
| 2. Application of Methodologies | Application of software development methodologies (Agile, Waterfall, DevOps) and project management techniques (Gantt charts, CPM, etc.) | 8 | Case study/assignment/practical |

| | | | |
|---|---|---|---|
| 3. Practical Tool Usage | Proficiency in using software engineering and project management tools (JIRA, MS Project, Git, etc.) | 6 | Lab/practical/project |
| 4. Project Planning and Execution | Ability to plan, schedule, and execute a software project, including resource allocation and risk mitigation | 8 | Project report/presentation |
| 5. Teamwork and Communication | Collaboration skills and communication effectiveness in project scenarios | 6 | Peer evaluation/presentation |
| 6. Innovation and Problem-Solving | Creative solutions and handling project challenges effectively | 6 | Project outcome and report |
| 7. Report and Documentation | Quality and clarity of project documentation and software engineering artifacts | 6 | Project report/documentation |

**External Assessment: (Marks 50)**
End Semester examination of 50 marks for 2 hours duration will be conducted.

**Major (Elective)**

| | |
|---|---|
| **Course Title** | **Theory of Computation and Application** |
| **Course Credit** | **4 Credits** |
| **Course Outcomes** | To give an overview of the theoretical foundations of computer science from the perspective of formal languages |
| | **CO1**. To give an overview of the theoretical foundations of computer science from the perspective of formal languages. |
| | **CO2**. To illustrate finite state machines to solve problems in computing |
| | **CO3**. Prove properties of languages and automata, including their limitations. |

| | |
|---|---|
| | **CO4**. To familiarize Regular grammars, context frees grammar. |
| | **CO5**. Understand the theoretical foundations of computation and its limitations. |
| **Module 1 (Credit 1)** | **Foundations of Theory of Computation** |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | 1. Understanding fundamental concepts in automata theory, formal languages, and computability |
| **Content Outlines** | 1. Define alphabet, string, language<br>2. Operations on languages. Introduce the concept of regular expressions and their relationship to regular languages.<br>3. Discuss deterministic finite automata (DFA), non-deterministic finite automata (NFA), and their equivalence. Explore regular languages and their representation using FA. |
| **Module 2 (Credit 1)** | Regular Languages and Regular Grammars |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Understand the fundamental concepts of formal languages and their relationship to automata |
| **Content Outlines** | 1. Introduce regular grammars and their relationship to regular languages and finite automata.<br>2. Cover pumping lemma for regular languages. |
| **Module 3 (Credit 1)** | Context-Free Languages (CFL) and Pushdown Automata (PDA) |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Understand the fundamental concepts of formal language theory, including the differences between regular and context-free languages, and the power of using a stack for recognizing patterns in languages that finite automata cannot handle |

| | |
|---|---|
| **Content Outlines** | 1. Introduce context-free grammars and their relationship to pushdown automata. Cover pumping lemma for CFLs.<br>2. Turing Machines<br>3. Concepts of Decidability and Undecidability, Computational Complexity, NP-Completeness |
| **Module 4**<br>**(Credit 1)** | Applications of Theory of Computation |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Design and analyze different types of automata, understand formal languages, and grasp the limitations of computation |
| **Content Outlines** | **1. Compiler Desig**n-Explain how concepts like regular expressions, context-free grammars, and pushdown automata are used in lexical analysis, parsing, and syntax checking.<br>**2. Short explanation about Database Theory**,<br>Formal Verification, Bioinformatics, Natural Language Processing, Cryptography |

**Evaluation Scheme:**

**Internal Assessment: (Marks 50)**

This involves a combination of tests, assignments, quizzes, and attendance. where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the internal assessment depending on evaluation rubrics given below

Tests:-One or more theory tests are conducted to assess the students' understanding of the course material.

**Assignments**:-Students are assigned problems and exercises related to the course content, which are evaluated at regular intervals.

Attendance:-Regular attendance in lectures is often a component of the internal assessment.

The Rubric will have the following Evaluation Parameters:

| Component | Mark s | Evaluation Criteria |
|---|---|---|

| | | |
|---|---|---|
| 1. Unit Tests / Internal Exams (2 Tests) | 20 | Problem solving (DFA/NFA/PDA/CFG/RE/TM) – 10 ,Theoretical questions & proofs – 10 |
| 2. Assignments | 10 | DFA/PDA/CFG design tasks – 5.Presentation, accuracy, and submission – 5 |
| 3. Practical | 10 | Correct implementation of automata models – 5 Output explanation – 5 |
| 4. Viva / Oral Assessment | 5 | Concept clarity (definitions, conversions) 3,Communication – 2 |
| 5. Attendance & Class Participation | 5 | Regularity and engagement |

**Classroom Activities**

Activities towards Comprehensive Continuous Evaluation(CCE)

## Module 1:

1. **Prove or disprove:** The relation R={(a,b)∣a≤b}R = \{(a, b) \mid a \leq b\}R={(a,b)∣a≤b} on the set of integers is reflexive, antisymmetric, and transitive.
2. **Construct a truth table** for the logical expression: (P→Q)∧(¬Q→¬P)(P \rightarrow Q) \land (\neg Q \rightarrow \neg P)(P→Q)∧(¬Q→¬P) and determine whether it is a tautology.
3. **Use proof by contradiction** to show that 2\sqrt{2}2 is an irrational number.

   Given the sets

   A={1,2,3},B={2,3,4}A = \{1, 2, 3\}, B = \{2, 3, 4\}A={1,2,3},B={2,3,4},
   compute A∪B,A∩B,A−B,B−AA \cup B, A \cap B, A - B, B - AA∪B,A∩B,A−B,B−A.
4. Determine whether the function

   f:Z→Zf: \mathbb{Z} \rightarrow \mathbb{Z}f:Z→Z, defined by f(x)=2x+1f(x) = 2x + 1f(x)=2x+1, is one-to-one and/or onto. Justify your answer.

## Module 2:

**Design DFA for a given language:-** Design a Deterministic Finite Automaton (DFA) over the alphabet Σ={0,1}\Sigma = \{0,1\}Σ={0,1} that accepts all strings ending with "01".

1. **Convert Regular Expression to NFA and then to DFA:-**Given the regular expression R=(a+b)∗abR = (a + b)^*abR=(a+b)∗ab, construct an NFA using Thompson’s construction and then convert it to an equivalent DFA.
2. **Prove a language is not regular using Pumping Lemma:-** Use the Pumping Lemma to prove that the language L={anbn∣n≥0}L = \{ a^n b^n \mid n \geq 0 \}L={anbn∣n≥0} is not regular.

3. **Minimize a given DFA:-** Consider any DFA with redundant states, apply the state minimization algorithm to find an equivalent minimized DFA.
4. **Construct a Regular Grammar:-** Construct a right-linear grammar (regular grammar) that generates the language L={w∈{0,1}∗|w ends with 11}L = \{ w \in \{0,1\}^* \mid w \text{ ends with } 11 \}L={w∈{0,1}∗|w ends with 11}.

## Module 3:

**1. Construct a Context-Free Grammar (CFG) :-** Design a context-free grammar (CFG) for the language L={anbn|n≥1}L = \{ a^n b^n \mid n \geq 1 \}L={anbn|n≥1}. Also, derive the string "aaabbb" using your grammar.
**2. Convert CFG to Chomsky Normal Form (CNF):-** Convert the following CFG to Chomsky Normal Form:S→aSb|abS \rightarrow aSb \mid abS→aSb|ab
**3. Design a PDA for a Given Language:-** Construct a Pushdown Automaton (PDA) that accepts the language L={anbn|n≥0}L = \{ a^n b^n \mid n \geq 0 \}L={anbn|n≥0} by empty stack or final state (specify your choice).
**4. Prove a Language is Not Context-Free:-** Use the Pumping Lemma for CFLs to show that the language L={anbncn|n≥1}L = \{ a^n b^n c^n \mid n \geq 1 \}L={anbncn|n≥1} is not context-free.
**5. Parse Tree Construction and Ambiguity Check:-** Given the CFG:S→SS|aS \rightarrow SS \mid aS→SS|a Draw two different parse trees for the string "aaa" and explain whether the grammar is ambiguous.

## Module 4:

**1. Essay/Report: Application of Automata in Compiler Design:-** Write a short essay or report explaining how Finite Automata are used in lexical analysis during compiler design. Include an example of token recognition using a DFA.

**2. Case Study: TOC in Natural Language Processing (NLP): -** Choose one real-world application of TOC in NLP (e.g., syntactic parsing, grammar checking). Describe how context-free grammars (CFGs) are used to model sentence structure in natural languages.
**3. Assignment: TOC in Artificial Intelligence / Machine Learning: -** Explain the role of formal languages and automata in the field of AI/ML. Give examples where automata or grammar-based systems are used in intelligent systems.
**4. Research Activity: TOC in Network Protocols / Cybersecurity: -** Investigate how regular expressions and automata are applied in validating network protocols or filtering malicious traffic. Submit a 1–2 page research summary with examples.
**5. Presentation: Real-Life Use Cases of Turing Machines: -** Prepare a group presentation on the theoretical concept of the Turing Machine and its relevance to modern-day computing devices or algorithmic problem solving. Use at least one real-life analogy or simulation.

**TEXT BOOKS:**1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman (2007), Introduction to Automata Theory, 2. Languages and Computation, 3rd edition, Pearson Education, India.

REFERENCE BOOKS:

1. K. L. P Mishra, N. Chandrashekaran (2003), Theory of Computer Science-Automata Languages and Computation, 2nd edition, Prentice Hall of India, India

Minor Stream

| | |
|---|---|
| **Course Title** | **Research Methodology** |
| **Course Credit** | **4 Credits** |
| **Course Outcome** | **CO1**: Demonstrate an understanding of the key steps involved in the research process, including problem formulation, literature review, data collection, analysis, and interpretation. |
| | **CO2**: Formulate clear and focused research questions and hypotheses based on a thorough review of existing literature and identification of research gaps. |
| | **CO3**: Conduct comprehensive literature reviews to identify relevant studies, theories, and methodologies within a specific research domain. |
| | **CO4**: Apply basic statistical techniques for data analysis, interpretation, and drawing meaningful conclusions from research findings |
| **Module 1(Credit 1)** | **Introduction to Research methodology** |
| **Learning Outcome** | **After learning the modules, learners will be able to** |
| | **Understand the fundamentals of research** – Explain the meaning, objectives, and significance of research in various disciplines. |
| | **Identify and define research problems** – Formulate clear and researchable problems and hypotheses based on real-world issues. |
| | **Conduct literature reviews** – Search, analyze, and synthesize existing research to support new investigations. |
| | **Apply appropriate data collection methods** – Use surveys, interviews, observations, and experiments effectively in gathering primary data. |
| | **Differentiate between qualitative and quantitative research** – Understand the applications, strengths, and limitations of both approaches. |
| **Content Outline** | ● **Research methodology**<br>An Introduction Objectives of Research, Types of Research, Research Methods and Methodology, defining a Research Problem, Techniques Involved in Defining a Problem.<br>● **Research Design** |

| | |
|---|---|
| | Need for Research Design, Features of Good Design, Different Research Designs, Basic Principles of Experimental Designs, Sampling Design, Steps in Sampling Design, Types of Sampling Design, Sampling Fundamentals, Estimation, Sample size Determination, Random sampling. |
| **Module2 (Credit 1)** | **Measurement and Scaling Techniques** |
| **Learning Outcome** | **After learning the modules, learners will be able to** |
| | **Understand the concepts of measurement in research** – Explain the meaning, importance, and levels of measurement (nominal, ordinal, interval, and ratio). |
| | **Understand the principles of scaling** – Explain the purpose and use of different scaling techniques in behavioral and social science research. |
| | **Interpret scaled data effectively** – Analyze and draw conclusions from data collected through various scaling methods. |
| **Content Outline** | ● **Measurement in Research**<br>Measurement Scales, Sources in Error, Techniques of Developing Measurement Tools, Scaling, Meaning of Scale, Scale Construction Techniques<br>● **Methods of Data Collection and Analysis**<br>Collection of Primary and Secondary Data, Selection of appropriate method Data Processing Operations, Elements of Analysis, Statistics in Research, Measures of Dispersion, Measures of Skewness, Regression Analysis, Correlation. |
| **Module3 (Credit 1)** | **Techniques of Hypotheses** |
| **Learning Outcome** | **After learning the modules, learners will be able to** |
| | **Understand the concept of a hypothesis** – Define hypothesis, its role, and importance in the research process. |
| | **Formulate testable hypotheses** – Develop clear, specific, and researchable hypotheses based on the research problem and literature review. |
| | **Select appropriate statistical tests** – Choose suitable parametric or non-parametric tests based on the type of data and research design. |

| | |
|---|---|
| **Content Outline** | ● **Techniques of Hypotheses**<br>Parametric or Standard Tests Basic concepts, Tests for Hypotheses I and II, Important parameters limitations of the tests of Hypotheses, Chi-square Test, Comparing Variance, as a non-parametric Test, Conversion of ChitoPhi, Caution in using Chi-square test. |
| **Module 4(Credit 1)** | **Analysis of Variance and Co-variance** |
| **Learning Outcome** | **After learning the modules, learners will be able to** |
| | **Understand the concepts of ANOVA and ANCOVA** – Explain the purpose, assumptions, and applications of variance and covariance analysis in research. |
| | **Apply One-way and Two-way ANOVA** – Analyze data involving one or more independent variables to test for significant differences between group means. |
| | **Interpret the results of ANOVA and ANCOVA** – Understand F-statistics, significance values, and post-hoc test results to make valid inferences. |
| **Content Outline** | ● **Analysis of Variance and Co-variance ANOVA**<br>One way ANOVA, Two Way ANOVA, ANOCOVA Assumptions in ANOCOVA, Multivariate Analysis Technique Classification of Multivariate Analysis, factor Analysis, R-type Q Type, factor Analysis, Path Analysis |

**References:**

1. Kothari, C. R., & Garg, G. (2019). Research methodology: Methods and techniques (4th ed.). New Age International Publishers.
2. Kumar, R. (2019). Research methodology: A step-by-step guide for beginners (5th ed.). SAGE Publications.
3. Cooper, D. R., & Schindler, P. S. (2014). Business research methods (12th ed.). McGraw-Hill Education.
4. Creswell, J. W., & Creswell, J. D. (2018). Research design: Qualitative, quantitative, and mixed methods approaches (5th ed.). SAGE Publications.
5. Zikmund, W. G., Babin, B. J., Carr, J. C., & Griffin, M. (2012). Business research methods (9th ed.). Cengage Learning.
6. Krishnaswamy, K. N., Sivakumar, A. I., & Mathirajan, M. (2009). Management research methodology: Integration of principles, methods and techniques. Pearson Education India.
7. Dawson, C. (2002). Practical research methods: A user-friendly guide to mastering research techniques and projects. How to Books.

**Classroom Activities**

## Activities towards Comprehensive Continuous Evaluation(CCE)

## Module 1:

**Research Proposal Development:**
Assignment: Ask students to develop a research proposal for a hypothetical research study. The proposal should include a clear research problem statement, objectives, literature review, research questions/hypotheses, methodology, and potential challenges.

### Critical Analysis of Research Articles:

Activity: Provide students with a set of research articles from different disciplines. Ask them to critically analyse the methodology section, identifying strengths and weaknesses, and discussing how they would improve the research design.

## Module 2:

**Survey Design and Implementation:**
Assignment: Have students design a survey on a topic of their choice.They should consider question wording, response options, and survey structure. After designing the survey, ask them to administer it to a small sample and analyse the results.

## Module 3:

Activity: Assign students a qualitative research design task. They can choose a research question and develop a plan for data collection (e.g.,interviews, focus groups, observation).

## Module 4:

**Peer Review Simulation:**
Activity: Have students conduct a peer review of a research proposal or a manuscript. This can include evaluating the clarity of the research question, appropriateness of methodology, and overall rigor of the study.

**Assessment**

## Internal Assessment – 50 Marks

## Evaluation Scheme:

Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

| **The Rubric will have the following Evaluation Parameters:** | | |
|---|---|---|
| **Evaluation Parameters** | **Description / Evaluation Points** | **Marks** |
| Understanding of Research Concepts | Demonstrates understanding of RM concepts, terms, and techniques | 10 |

| | | |
|---|---|---|
| Problem Formulation & Hypothesis Framing | Ability to identify and define a research problem and objectives | 10 |
| Research Design & Method Selection | Selection and justification of research design, tools, sampling techniques | 10 |
| Data Collection & Analysis | Quality and relevance of collected data; proper statistical treatment | 10 |
| Report Writing & Documentation | Structure, clarity, referencing, and academic tone of report | 5 |
| Presentation & Communication Skills | Ability to present findings clearly and confidently | 5 |

(Given below are two sample projects but it is expected to work on similar sort of projects)

Project 1:

**Research Presentation:**

Assignment: Ask students to create a presentation summarizing a research paper. They should highlight the key elements of the study, discuss the methodology, and present the findings. Encourage a focus on effective communication.

**Project 2:**

Case Study Analysis:

Activity: Provide students with a research-related case study involving methodological challenges. Ask them to analyze the case, identify issues, and propose solutions based on their understanding of research methodology.

External Assessment: (Marks 50)

End Semester examination of 50 marks for 2 hours duration will be conducted.

Semester II (NEP pattern)

Major Core

| **Course Title** | **Artificial Intelligence** |
|---|---|
| **Course Credit** | **4 Credits** |
| **Course Outcome** | **CO1**: Understand the fundamental concepts, history, and evolution of Artificial Intelligence, including key domains and applications. |
| | **CO2**: Apply problem-solving techniques such as search algorithms (uninformed and informed searches) and game playing strategies for developing intelligent solutions. |
| | **CO3**: Demonstrate the ability to implement knowledge representation methods like logic-based models, semantic networks, and ontologies for AI problem domains. |
| **Module 1(Credit 1)** | **Introduction to Artificial Intelligence** |
| **Learning Outcome** | **After learning the modules, learners will be able to** |
| | Define Artificial Intelligence and explain its scope, history, and evolution. |
| | Identify key areas and applications of AI across various industries like healthcare, education, finance, and automation. |
| | Illustrate real-world examples of AI systems and discuss how they improve efficiency and decision-making. |
| **Content Outline** | **Definition, History & Evolution of AI**<br>Types of AI: Narrow AI, General AI, Super AI<br>Applications of AI in various sectors<br>**Intelligent Agents:** Types, Structures, and Environments<br>Solving by searching:<br>Travelling salesman Problem, Monkey Banana Problem, Tower of Hanoi and Block World. |
| **Module2 (Credit 1)** | **Problem solving by Searching Methods** |
| **Learning Outcome** | **After learning the modules, learners will be able to** |
| | Explain the role of search in problem-solving within Artificial Intelligence. |
| | Differentiate between various search strategies, including uninformed (blind) and informed (heuristic) search techniques. |

| | |
|---|---|
| | Apply informed search techniques like Greedy Best-First Search and A* Search for efficient problem-solving. |
| **Content Outline** | **Uninformed Search Techniques**<br>Breadth-First Search (BFS)<br>Depth-First Search (DFS)<br>Iterative Deepening Search<br>**Informed Search Techniques**<br>Best-First Search<br>A* Search<br>Hill Climbing<br>**Adversarial Search**<br>Game Playing: Minimax Algorithm<br>Alpha-Beta Pruning |
| **Module3 (Credit 1)** | **Knowledge Representation and Reasoning** |
| **Learning Outcome** | **After learning the modules, learners will be able to** |
| | Describe the importance of knowledge representation in building intelligent systems. |
| | Apply logic-based methods (Propositional and Predicate Logic) to represent facts, rules, and relationships in problem domains. |
| | Implement reasoning techniques such as forward chaining, backward chaining, and resolution to infer new knowledge from existing information. |
| **Content Outline** | **Knowledge Representation Techniques**<br>Propositional Logic<br>Predicate Logic<br>Semantic Networks<br>Frames and Ontologies<br>**Reasoning Methods**<br>Forward Chaining<br>Backward Chaining<br>Resolution<br>**Reasoning under Uncertainty**<br>Bayesian Networks<br>Fuzzy Logic and Reasoning |
| **Module 4(Credit 1)** | **Knowledge Representation** |
| **Learning Outcome** | **After learning the modules, learners will be able to** |
| | Explain various knowledge representation techniques such as logic-based models, semantic networks, frames, and ontologies for structuring information in AI systems. |

| | |
|---|---|
| | Describe the architecture and components of Expert Systems, including knowledge base, inference engine, and user interface. |
| | Identify challenges and limitations in representing knowledge, reasoning under uncertainty, and processing natural language. |
| **Content Outline** | **Structured Knowledge Representation:**<br>Associative Networks, Semantic Nets, Frames Structures, Conceptual, Dependencies & Scripts, Learning – Concept of Learning – Learning Automata, Learning by induction.<br>**Natural Language Processing:**<br>Overview of Linguistics, Grammars and Languages, basic Parsing techniques, semantic analysis, and representation structures. Natural Language generation and Natural Language Systems.<br>**Expert System:**<br>Knowledge acquisition methods, knowledge engineering process, goals in knowledge system development, basic architecture of expert systems, problem domain versus knowledge domain, Development of ES and life cycle of ES. Advantages of expert system, structure of Rule based expert system, characteristics of conventional system and expert system. |

**References**:

1. Introduction to AI and Expert Systems - Patterson.
2. Artificial Intelligence A Modern Approach- Stuart Rushell.
3. Artificial Intelligence - Rich E and Knight K.
4. Principles of Artificial Intelligence - Nilsson.
5. Artificial Intelligence – An Engineering Approach – Schalkoff R J.
6. Introduction to Expert System - Peter Jackson.
7. Artificial Intelligence – Janakiraman.

Evaluation:
Internal Evaluation: 50 Marks

## External Assessment: (Marks 50)

End Semester examination of 50 marks for 2 hours duration will be conducted.

Major (Core)

| | |
|---|---|
| **Course Title** | **Advance DBMS** |
| **Course Credits** | **4 Credits** |
| **Course Outcomes** | **CO1**: -Students will grasp the theoretical foundations of advanced database systems, including object-oriented databases, parallel databases, and distributed databases. |
| | **CO2**: -Students will be able to understand the basic concepts of transaction management, including various concurrency control protocols and recovery mechanisms. |
| | **CO3**: - Students will be able to analyze the architecture and design of distributed and parallel databases, and understand data distribution, query processing, and system coordination. |
| | **CO4:** -Students will be able to differentiate NoSQL databases from traditional RDBMS, and explain the need for NoSQL in handling big data, horizontal scalability, and schema flexibility. |
| **Module 1 (Credit 1) Transaction Management and Concurrency Control** | |
| **Learning Outcomes** | **After learning the modules, Learners will be able to** |
| | 1. Understand the concept of Transactions. |
| | 2. Explain the need for concurrency control in multiuser database environments. Understand and apply locking protocol. |
| | 3.Determine the serializability of transaction schedules using conflict and view serializability. |
| **Content Outline** | **Introduction to Transaction**: - Define the purpose of Transaction, ACID property, Tra state.<br>**Transaction Processing**: - Transaction life cycle, System and user-define transaction, se vs concurrent execution<br>**Concurrency Control**: - Need for concurrency in DBMS, Problems in concurrency, S and their type.<br>**Serializability**: - Conflict serializability, view serializability.<br>**Concurrency Control Protocol**: - Lock-based protocol Binary locks Shared and exclusive locks, Two-phase locking (2PL) protocol, Strict 2PL and rigorous 2PL,Timestamp-based protocol. |
| **Module 2 (Credit 1) Introduction to Advance Database Model** | |
| **Learning Outcomes** | **After learning the modules, Learners will be able to** |
| | 1.Compare and contrast advanced data models with traditional relational models. |

| | |
|---|---|
| | 2. Develop solutions involving object-oriented, temporal, or spatial data using relevant tools or extensions. |
| | 3. Analyze use-cases and select appropriate advanced database models for different application domains |
| **Content Outline** | **Introduction to Object-Oriented Databases**: - Features of OODB, Object Oriented Data Models (e.g. ODMG, ORDBMS extensions<br>**Introduction to Temporial Databases**: - Difference between conventional databases and temporal databases. Types of temporal data: valid time, transaction time, bitemporal, temporal Data Model – time-stamped tuples, attribute-version, Event-based models.<br>**Introduction to Spatial Databases: -** Definition and importance of spatial databases, Applications: GIS, urban planning, environmental monitoring, LBS (location-based services), Spatial data types: Points, Lines, Polygons, Raster vs. Vector data models<br>**Introduction to Deductive Databases: -** Definition and motivation, Differences between relational and deductive databases , Applications of deductive databases (rule-based systems, expert systems, knowledge bases) |
| **Module 3 (Credit 1) Distributed and Parallel Databases** | |
| **Learning Outcomes** | **After learning the modules**, **Learners will be able to**<br>1.Understand the need for distributed and parallel database systems. |
| | 2.Differentiate between centralized, distributed, and parallel database systems. |
| | 3.Describe the architectures of distributed and parallel database systems and identify the challenges and trade-offs associated with distributed and parallel database systems. |
| **Content Outline** | **Introduction to Distributed Databases**: - Centralized vs. Distributed vs. Parallel Systems.<br>Need for distributed and parallel databases. Performance metrics (e.g., speedup, scale-up). Basic concepts: fragmentation, replication, and distribution.<br>**Distributed Database Architectures**: - Shared-nothing, shared-disk, and shared-memory architectures. Examples of distributed database systems.<br>**Introduction to Distributed Database Design**: - Data fragmentation (horizontal, vertical, derived). Data allocation strategies. |
| **Module 4 (Credit 1) NoSQL Databases** | |
| **Learning Outcomes** | **After learning the module, learners will be able to**<br>1. Explain the differences between NoSQL and traditional relational databases. |
| | 2. Describe the need for NoSQL in handling big data, scalability, and flexible schemas. |
| | 3.Understand BASE properties and how they differ from ACID properties |

| | |
|---|---|
| **Content Outline** | **Introduction to NoSQL: -** What is NoSQL? Comparison with relational databases, **Characteristics of NoSQL databases**: - Schema-free design, Horizontal scalability, BASE vs. ACID properties<br>**Types of NoSQL Data Models**: - Classification of NoSQL databases: Key-Value Stores (e.g., Redis, Riak); Document Stores (e.g., MongoDB, CouchDB); Column-Family Stores (e.g., Cassandra, HBase); Graph Databases (e.g., Neo4j) |

References:-

1. "Database System Concepts" Authors: Abraham Silberschatz, Henry F. Korth, and S. Sudarshan, Publisher: McGraw-Hill
2. "Fundamentals of Database Systems" Authors: Ramez Elmasri and Shamkant B. Navathe, Publisher: Pearson
3. "Advanced Database Management System" Author: Dr. Rini Chakravarty Publisher: Dreamtech Press
4. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence" Authors: Pramod J. Sadalage and Martin Fowler, Publisher: Addison-Wesley
5. Principles of Distributed Database Systems Authors: M. Tamer Özsu and Patrick Valduriez Publisher: Springer

**Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)**

**<u>Classroom Activities:</u>**

**Activities towards Comprehensive Continuous Evaluation (CCE)**

**Module 1:**

1. Use of SQL commands for managing transactions: BEGIN, COMMIT, ROLLBACK, SAVEPOINT.

Module 2:

1. Case study on design the OOAD schema.
2. Short quiz or viva on spatial data types , deductive database, temporal database.

Module 3:

1.Design a simple distributed database schema for a fictional organization with multiple branch offices (e.g., a university or retail chain). Include: Schema design with tables, Distribution strategy (horizontal/vertical fragmentation or replication) , Justification for your design choices.

Module 4:

1. Set up and interact with at least one NoSQL database tool (e.g., MongoDB, Redis, or Neo4j).
2. Introduced and explored any one database tools for handling any advance database e.g. explore MongoDB .

**Assessment**

Internal Assessment – 50 Marks

Evaluation Scheme: -

Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

| **The Rubric will have the following Evaluation Parameters:** | | |
|---|---|---|
| **Evaluation Parameters** | **Description / Evaluation Points** | **Marks** |
| Remembering | The basic definition of the transaction analysis, ACID concept, other advance concept about the advance databases | 10 |
| Understanding | Understand the schema design of the different type of databases | 10 |
| Applying | The student can design the database schema for different type of databases through case study | 10 |
| Creating | Install the MongoDB and operate it. | 15 |
| Submission | Completeness, correctness, formatting, and timely submission | 5 |
| | Total | 50 |

(Given below are two sample projects but it is expected to work on similar sort of projects)

Project 1:

1. Using MongoDB create the Database schema and execute queries on it.

Project 2:

2. Design the case study for creation of database Schema for OORDBMs, Spital database.

External Assessment: (Marks 50)

End Semester examination of 50 marks for 2 hours duration will be conducted.

Major (Core)

| | |
|---|---|
| **Course Title** | **Data Science Using Python** |
| **Course Credits** | **4 Credits** |
| **Course Outcomes** | **CO1**: Students will be able to define the concept, scope, and evolution of Data Science and differentiate between Data Science, Machine Learning, and Artificial Intelligence. |
| | **CO2**: Students will be able to explain the phases of the Data Science lifecycle, including data collection, preprocessing, modeling, evaluation, and deployment. |
| | **CO3:** Students will be able to apply descriptive statistics and visualization techniques to summarize and interpret datasets, and identify patterns, trends, and anomalies in data. |
| | **CO4:** Students will be able to describe basic machine learning algorithms (e.g., regression, classification, clustering) and explain the principles of model training, testing, and evaluation. |
| | **CO5**: Students will be able to discuss issues related to data privacy, bias, and fairness, and evaluate the importance of ethical data handling and responsible AI. |
| **Module 1 (Credit 1) Introduction to Data Science and Python** | |
| **Learning Outcomes** | **After learning the modules**, **Learners will be able to** |
| | 1. Understand the basic concepts, scope, and components of Data Science. |
| | 2. Familiarize with data types, sources, and preprocessing techniques. |
| | 3.Learn the foundational tools and applications of data science in real-world problems. |
| **Content Outline** | **Overview of Data Science**: What is data science, its applications,<br>Difference between Data Science, Data Analytics, and Business Intelligence, The Data Science lifecycle (CRISP-DM or OSEMN model) ,Roles in data science: Data Scientist, Analyst, Engineer, etc.<br>Applications across industries (healthcare, finance, education, etc.)<br>**Data Science Components:** - Structured, Semi-structured, and Unstructured data , Introduction to Big Data, Data Collection: Sources and Methods, Data Warehousing and Data Lakes, Data preprocessing: Cleaning, Transformation, Normalization |
| **Module 2 (Credit 1) Data Manipulation, Analysis and Visualization with Python** | |
| **Learning Outcomes** | **After learning the modules**, **Learners will be able to** |
| | 1. Understand the structure and advantages of NumPy |
| | 2. Create and manipulate Pandas Series and DataFrames for structured data. |

| | |
|---|---|
| | 3. Create basic visualizations such as line plots, bar charts, and scatter plots using Matplotlib |
| **Content Outline** | **NumPy**: Working with arrays, numerical operations, linear algebra, and array manipulation.<br>**Pandas**: Data structures like Series and DataFrames, data cleaning, filtering, and transformation.<br>**Data Input/Output**: Reading and writing data from various formats (CSV, Excel, etc.).<br>**Matplotlib**: Creating basic plots, charts, and graphs to visualize data.<br>**Seaborn**: Advanced visualization techniques, statistical plots, and creating visually appealing charts.<br>Exploratory Data Analysis (EDA) :-Descriptive statistics (mean, median, mode, std. dev.), Data visualization techniques: Histograms, Box plots, Scatter plots ,Correlation and Causation Outlier detection and missing value handling. |
| **Module 3 (Credit 1) Statistical Concepts** | |
| **Learning Outcomes** | **After learning the modules**, **Learners will be able to** |
| | . 1. Define and compute measures of central tendency (mean, median, mode) to summarize data. Calculate measures of dispersion (range, variance, standard deviation) to describe data spread. |
| | 2.Interpret data distribution using graphical techniques (histograms, boxplots) and numerical summaries. |
| | 3. Explain the concept of population vs. sample and the need for inferential statistics. |
| | 4.Understand fundamental probability concepts including independent, dependent events, and conditional probability. |
| **Content Outline** | **Descriptive Statistics**: Measures of central tendency, dispersion, and distribution. **Inferential Statistics**: Hypothesis testing, confidence intervals, and statistical significance. **Probability:** Basic probability concepts, distributions (normal, binomial, etc.). |
| **Module 4 (Credit 1) Machine Learning model Development** | |
| **Learning Outcomes** | **After learning the modules**, **Learners will be able to** |
| | 1. Use Scikit-learn to build machine learning pipelines including data preprocessing, model training, and evaluation. |
| | 2. Build and evaluate linear regression and logistic regression models for predictive tasks. Apply classification algorithms like Decision Trees and Support Vector Machines (SVM) to labeled datasets. |
| | 3. Implement K-Means and Hierarchical clustering using Scikit-learn.<br>Interpret clustering results and visualize groupings effectively. |

| | |
|---|---|
| **Content Outline** | **Introduction to Scikit-learn**: - Using the Scikit-learn library for building and evaluating machine learning models.<br>**Supervised Learning**: - Linear regression, logistic regression, classification algorithms (e.g., decision trees, SVM).<br>**Unsupervised Learning**: -Clustering (k-means, hierarchical clustering).<br>**Model Evaluation**: -Metrics for evaluating model performance (e.g., accuracy, precision, recall). |

**References:-**

1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, Author: Aurélien Géron, Publisher: O'Reilly Media
2. Data Science from Scratch: First Principles with Python , Author: Joel Grus, Publisher: O'Reilly Media
3. Think Stats: Probability and Statistics for Programmers ,Author: Allen B. Downey,Publisher: O’Reilly / Green Tea Press
4. Practical Statistics for Data Scientists , Authors: Peter Bruce, Andrew Bruce, and Peter Gedeck Publisher: O'Reilly Media
5. Introduction to Machine Learning with Python Authors: Andreas C. Müller & Sarah Guido Publisher: O'Reilly Media

## Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

## Classroom Activities

### Module 1: Python Basics for Data Science

Activity 1. Assignment on using lists, tuples, dictionaries, and string manipulations in a dataset-like context.

Activity 2: Implement NumPy operations (array creation, slicing, reshaping, and broadcasting).

Activity 3: Use Pandas to load a CSV dataset, clean missing values, and perform filtering and aggregation.

### Module 2: Data Visualization

Activity 1: Create plots using Matplotlib – line chart, bar graph, scatter plot with proper labels and titles.

Activity 2: Use Seaborn to generate correlation heatmaps, boxplots, and pairplots for a given dataset.

### Module 3: Exploratory Data Analysis (EDA)

Activity 1: Perform EDA on a real-world dataset (e.g., Titanic, Iris, or custom). Summary statistics, distribution, missing values, correlation matrix.

## Module 4 : Machine Learning with Scikit-learn

Activity 1: Build and evaluate a linear regression model using Scikit-learn.

Activity 2: Classification using Decision Tree or SVM; evaluate using accuracy, precision, recall.

Activity 3: Apply K-means clustering on a dataset and visualize the results.

Evaluation: Algorithm implementation, model accuracy, justification of choice of algorithm.

### Assessment

## Internal Assessment – 50 Marks

**Evaluation Scheme: -** Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

| **The Rubric will have the following Evaluation Parameters:** | | |
|---|---|---|
| **Evaluation Parameters** | **Description / Evaluation Points** | **Marks** |
| Remember- Syntax Knowledge (NumPy/Pandas) | · Students recall and use correct syntax of NumPy and Pandas.<br>· Ability to write syntactically correct Python code using NumPy and Pandas. | 10 |
| Understand Statistical Concept | · Students understand core statistical concepts and know when to apply them to a given problem.<br>· Correct identification of suitable statistical techniques (mean, std, correlation).<br>· Justification for method choice in context. | 10 |
| Apply | · Students apply Python libraries and statistical techniques on real-world case studies or datasets.<br>· Implement end-to-end data preprocessing, EDA, and insight extraction. | 10 |
| Analyze / Evaluate | · Students develop machine learning models and evaluate their performance using appropriate metrics.<br>· Model selection, training, and testing.<br>· Evaluation using accuracy, precision, recall, etc. | 15 |
| | · Completeness, correctness, formatting, and timely submission | 5 |
| | Total | 50 |

(Given below are sample projects but it is expected to work on similar sort of projects)

**Project Work (Summative)**

- Select a dataset (Kaggle/UCI).
- Perform EDA, build ML models, visualize results.
- Submit report + present findings.
- Evaluation Criteria:
- Problem statement clarity
- Code and data analysis
- Interpretation and insights
- Oral presentation & report

Major (Elective II)

| **Course Title** | **Software Testing and Quality assurance** |
|---|---|
| **Course Credits** | **4 Credits** |
| | **CO1**- Understand the fundamental principles and concepts of software testing, including its purpose, objectives, and the role it plays in software development. |
| | **CO2-** Apply various testing methods such as functional, non-functional, unit, integration, system, regression, and acceptance testing to diverse software systems. |
| | **CO3**- Explain software quality concepts and quality assurance practices, models, and standards like ISO and CMMI. |
| | **CO4**- Enable students to evaluate software quality using metrics, process audits, and quality improvement strategies. |
| | **CO 5**- Understand quality assurance principles and best practices, emphasizing the importance of testing in the software development lifecycle. |
| **Module 1 (Credit 1)**<br>**Fundamentals of Software Testing** | |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Gain proficiency in software testing terminologies, methodologies, and life cycles. |
| | Describe key testing concepts, including error, fault, failure, defect, and bug. |

| | |
|---|---|
| | Differentiate between software testing types such as manual vs automated testing, static vs dynamic testing. |
| | Recognize the principles and objectives of testing, including early testing, defect clustering |

| | |
|---|---|
| | Understand and illustrate the phases of the Software Testing Life Cycle (STLC) |
| **Content Outline** | **Module Contents:**<br>**Introduction to Software Testing**<br>● Goals and Principles of Testing<br>● Software Development Life Cycle (SDLC) vs Software Testing Life Cycle (STLC)<br>● V-Model and Agile Testing Overview<br>● Types of Testing: Manual vs Automated, Static vs Dynamic<br>● Defect Lifecycle, Severity & Priority |
| **Module 2 (Credit 1)**<br>**Testing Techniques** | |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Differentiate between black-box and white-box testing techniques and identify when to apply each in real-world scenarios. |
| | How to apply black-box testing techniques and white-box testing techniques |
| | Analyze code and specifications to design appropriate test cases using both black-box and white-box methods. |
| | Explain the significance of static techniques, including code reviews and walkthroughs, in improving software quality. |
| | Select suitable testing techniques based on software requirements and risk analysis |

| | |
|---|---|
| **Content Outline** | **Module Content:**<br>**Black Box Techniques**<br>● Equivalence Partitioning<br>● Boundary Value Analysis<br>● Cause-Effect Graphing<br>● Decision Table Testing<br>**White Box Testing Techniques**<br>● Statement, Branch, Condition, and Path Coverage<br>● Cyclomatic Complexity<br>● Code Reviews and Walkthroughs |
| **Module 3 (Credit 1)** | |
| **Test Management** | |
| **Learning Outcomes** | **After learning the module, learners will be able to**<br>Understand the components of test management, including test planning, scheduling, monitoring, and control. |
| | Create a comprehensive test plan, including scope, objectives, resources, schedules, and deliverables. Design and document effective test cases and test scenarios based on software requirements. |
| | Plan and perform test closure activities, including final reporting, test summary creation, and lessons learned |
| | Understand configuration and version control and their importance in maintaining test artifacts. |
| | Coordinate testing efforts within teams, ensuring collaboration, communication, and adherence to deadlines |
| **Content Outline** | ● Test Planning and Documentation<br>● Test Strategy, Test Plan Components<br>● Test Case Design and Test Data Preparation<br>● Test Environment Setup<br>● Test Execution and Defect Reporting<br>● Test Closure Activities<br>● Configuration Management and Version Control |
| **Module 4 (Credit 1)**<br>**Software Quality Assurance (SQA) and Automation and Tools** | |
| **Learning** | **After learning the module, learners will be able to** |

| | |
|---|---|
| **Outcomes** | Identify and apply key software quality attributes, including functionality, reliability, usability, efficiency, maintainability, and portability. |
| | Apply SQA practices such as reviews, audits, and inspections to improve software quality and reduce defects. |
| | Differentiate between process assurance and product assurance, and explain their relevance in a quality framework |
| | Analyze software process improvement models, such as CMMI, Six Sigma, and Total Quality Management (TQM), and their impact on software projects. |
| | Differentiate between various automation frameworks, such as linear, modular, data-driven, keyword-driven, and hybrid frameworks. |
| | Use popular automation tools (e.g., Selenium, JUnit/TestNG) to write, execute, and manage automated test scripts.Apply bug-tracking tools (e.g., Bugzilla, Mantis) to report, manage, and track defects throughout the test lifecycle. |

| | |
|---|---|
| | Critically evaluate organizational quality assurance strategies, and recommend improvements based on industry best practices. |
| **Content Outline** | **Concepts of Software Quality**<br>● Quality Factors (McCall's and ISO 9126)<br>● SQA Activities and Audits<br>● Reviews: Formal Technical Reviews, Inspections<br>● Software Process Improvement Models:<br>1. Capability Maturity Model Integration (CMMI)<br>2. ISO 9001:2015<br>3. Six Sigma in Software Engineering<br>**Automation and Tools**<br>Basics of Test Automation<br>Benefits and Challenges of Automation<br>Frameworks: Linear, Modular, Data-Driven, Keyword-Driven, Hybrid<br>Overview of Tools:<br>● Selenium<br>● JUnit/TestNG<br>● Bug Tracking: Bugzilla, Mantis<br>● CI Tools: Jenkins<br>Introduction to Performance Testing (LoadRunner, JMeter) |
| **Assignments towards Comprehensive Continuous Evaluation** | |

**Module 1:**

1. Explain the software testing life cycle (STLC) with diagrams.
2. Describe the role of a test plan. What are its key components?
3. Explain the significance of early testing in SDLC.
4. Provide a case study of an organization's testing process and have students analyze and propose improvements based on maturity model principles.
5. Case study -A software product is nearing completion. As a test engineer, what approach will you take to ensure it meets quality standards?

**Module 2:**

1. Differentiate between black-box and white-box testing techniques.
2. What is boundary value analysis? Why is it important?
3. State any two advantages of using decision table-based testing
4. Explain the following black-box testing techniques with examples.
   a) Equivalence Class Partitioning
   b) Boundary Value Analysis
5. Describe the following white-box testing techniques with examples:
   a) Statement Coverage
   b) Branch Coverage
6. Design test cases using equivalence partitioning and boundary value analysis for a function that accepts age input between 18 and 60.
7. Given a simple program to calculate grades based on percentage, create a decision table and generate test cases.

**Module 3:**

1. What is test management? Define the role of a test plan in test management.
2. Differentiate between test case and test scenario.
3. Describe the test management activities throughout the Software Testing Life Cycle (STLC).
4. What are the types of risks in testing? How are they managed?
5. Discuss the importance of traceability in test management. How is a traceability matrix prepared and used?
6. Describe the challenges in test management for large-scale enterprise software and how they can be addressed.
7. Evaluate the role of a test manager in Agile and DevOps environments.
8. You are the test lead for an e-commerce project. Outline how you will manage:
   - Test planning
   - Test team coordination
   - Defect reporting
   - Test metrics tracking
9. Given a project with limited time and resources, how would you prioritize and manage test activities effectively?

**Module 4:**

1. Define Software Quality Assurance. How does it differ from software testing?
2. What are the different types of software quality standards? Give examples and explain their importance.
3. Discuss the role of metrics in SQA. What types of metrics are commonly used to measure software quality?
4. Describe the difference between defect prevention and defect detection in the context of SQA.
5. What is test automation? Explain its benefits and limitations in software testing.
6. Explain the difference between automated testing and manual testing. When should automation be preferred?
7. What is Continuous Integration (CI), and how do automated tests fit into CI pipelines?
8. Describe the process of selecting an automation tool for a software project. What factors should be considered?

**Reference Books :**

1. Software Testing Principles and Practices – Srinivasan Desikan, Gopalaswamy Ramesh
2. Foundations of Software Testing – Dorothy Graham, Erik van Veenendaal, Isabel Evans
3. Software Engineering – Ian Sommerville (Relevant QA chapters)
4. Effective Software Testing – Elfriede Dustin
5. Software Quality Assurance – Daniel Galin
6. Testing Computer Software – Cem Kaner

**Assessment:**
**Internal Assessment – 50 Marks**

**The Rubric will have the following Evaluation Parameters:**

| Criteria | Excellent (9-10 marks) | Good (7-8 marks) | Satisfactory (5-6 marks) | Needs Improvement (0-4 marks) | Marks |
|---|---|---|---|---|---|
| **1. Understanding of Concepts** | Deep understanding of testing fundamentals, QA principles, methodologies; clear explanation with examples. | Good understanding with minor gaps; explains most concepts well. | Basic understanding; some misconceptions or incomplete explanations. | Poor understanding; many misconceptions or unclear explanations. | /10 |
| **2. Application of Testing Techniques** | Correctly applies unit, integration, system, acceptance testing techniques in practical scenarios. | Applies most techniques correctly with minor errors or omissions. | Applies basic techniques but misses advanced or appropriate application. | Fails to apply correct testing techniques or misunderstands them. | /10 |
| **3. Quality Assurance Processes** | Clearly explains and applies QA processes including planning, defect management, metrics, standards. | Explains QA processes with minor inaccuracies or omissions. | Basic description of QA processes but lacks depth or practical understanding. | Limited or incorrect understanding of QA processes. | /10 |
| **4. Use of Tools & Automation** | Proficient use of testing tools (Selenium, JIRA, JUnit) and automation frameworks; explains advantages. | Uses tools adequately but with limited depth or some errors. | Basic use of tools; mostly manual testing or unclear on automation. | No or incorrect use of testing tools or automation. | /10 |

| | | | | | |
|---|---|---|---|---|---|
| **5. Critical Analysis & Reporting** | Produces clear, structured test reports; critically analyses results, defects, suggests,improvements. | Produces acceptable reports; some analysis present but lacks depth. | Reports are basic with limited analysis or unclear presentation. | Poor or no reports; lacks analysis and critical thinking. | /10 |

**External Assessment: (Marks 50)**

End Semester examination of 50 marks for 2 hours duration will be conducted.

Major (Elective)

| | |
|---|---|
| **Course Title** | **Fuzzy Logic and Neural Networks** |
| **Course Credits** | **4 Credits** |
| **Course Outcomes** | **CO1**: Understand the fundamental concepts of fuzzy logic systems |
| | **CO2**: Understand the structure and functioning of biological and artificial neural networks |
| | **CO3:** Implement learning algorithms for single-layer and multi-layer neural networks |
| | **CO4** : Analyze and compare the performance of neural and fuzzy systems for various applications |

| **Module 1 (Credit 1)** | **Fundamentals of Fuzzy Logic** |
|---|---|
| **Learning Outcomes** | **After learning the modules**, **Learners will be able to** |
| | 1. **Differentiate** between classical (crisp) sets and fuzzy sets in terms of uncertainty handling. |
| | 2. **Explain** the concepts of membership functions and perform fuzzy set operations like union, intersection, and complement. |
| | 3.**Construct** fuzzy relations and apply fuzzy composition techniques. |
| **Content Outline** | Crisp sets vs Fuzzy sets, Fuzzy set operations (union, intersection, complement), Properties of fuzzy sets, Fuzzy relations and fuzzy composition, Membership functions: types and properties, Fuzzification and defuzzification techniques, Applications of fuzzy logic in control systems |

| | |
|---|---|
| **Module 2 (Credit 1) Fuzzy Inference and Reasoning Systems** | |
| **Learning Outcomes** | **After learning the modules**, **Learners will be able to** |
| | 1. **Explain** the structure and components of a fuzzy rule-based system including linguistic variables and fuzzy rules. |
| | 2. **Implement** fuzzy rule evaluation and approximate reasoning to derive conclusions from imprecise inputs. |
| | 3. **Analyze** and compare the behavior of different inference mechanisms. |
| **Content Outline** | Linguistic variables and fuzzy rules, Mamdani and Sugeno fuzzy models, Inference mechanism and rule evaluation, Approximate reasoning, Design of fuzzy logic controllers, Real-world applications: Industrial control, smart appliances |
| **Module 3 (Credit 1) Fundamentals of Neural Networks** | |
| **Learning Outcomes** | **After learning the modules**, **Learners will be able to** |
| | 1.**Explain** the structure and functioning of biological and artificial neurons. |
| | 2.**Differentiate** between various neural network architectures such as feedforward and feedback networks. |
| | 3.**Describe** the working of basic neural network models like Perceptron, ADALINE, and MADALINE. |
| | 4.**Solve** simple classification and pattern recognition problems using basic neural models. |
| **Content Outline** | Biological neuron vs artificial neuron, Neural network models: Perceptron, ADALINE, MADALINE, Activation functions: Sigmoid, Tanh, ReLU, Learning rules: Hebbian, Delta, Perceptron rule, Multi-layer perceptron (MLP), Backpropagation algorithm: derivation and working |
| **Module 4 (Credit 1) Advanced Neural Networks and Neuro-Fuzzy Systems** | |

| | |
|---|---|
| **Learning Outcomes** | **After learning the modules, Learners will be able to** |
| | 1. **Describe** the structure and working of advanced neural network models such as Self-Organizing Maps (SOM), Radial Basis Function (RBF) Networks, and Hopfield Networks. |
| | 2. **Explain** the concept and architecture of Neuro-Fuzzy systems like ANFIS (Adaptive Neuro-Fuzzy Inference System). |
| | 3. **Apply** advanced neural and neuro-fuzzy models for solving complex problems in classification, control systems, and prediction. |
| **Content Outline** | Self-organizing maps (SOM), Radial basis function networks, Hopfield networks (brief overview), Introduction to Neuro-Fuzzy systems (ANFIS), Hybrid systems combining fuzzy logic and neural networks, Case studies and real-world applications (classification, prediction) |

**References:**

1. Ross, T. J. (2010). *Fuzzy logic with engineering applications* (3rd ed.). Wiley.
2. Haykin, S. (2009). *Neural networks and learning machines* (3rd ed.). Pearson Education.
3. Rajasekaran, S., & Vijayalakshmi Pai, G. A. (2003). *Neural networks, fuzzy logic and genetic algorithms: Synthesis and applications*. PHI Learning.
4. Kosko, B. (1992). *Neural networks and fuzzy systems: A dynamical systems approach to machine intelligence*. Prentice Hall.
5. Jang, J. S. R., Sun, C. T., & Mizutani, E. (1997). *Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence*. Pearson Education.

6. Fausett, L. (1994). *Fundamentals of neural networks: Architectures, algorithms and applications*. Prentice Hall.

**Assignments towards Comprehensive Continuous Evaluation**

**Module 1:**

1. Define fuzzy sets and explain how they differ from classical sets with examples.
2. Draw and describe different types of membership functions (triangular, trapezoidal, Gaussian).
3. Perform fuzzy set operations (union, intersection, complement) on given data.
4. Design a fuzzification and defuzzification process for a washing machine temperature control system.

**Module 2:**

1. Construct a Mamdani fuzzy inference system for a traffic control problem using at least three linguistic variables.
2. Implement a simple Sugeno model using Python/Matlab/Excel.
3. Create a fuzzy rule base for an air conditioner control system.
4. Discuss the role of approximate reasoning with a real-world scenario (e.g., medical diagnosis or flood prediction).

**Module 3:**

1. Explain the difference between biological and artificial neurons with diagrams.
2. Implement a single-layer perceptron model in Python to solve the AND/OR/XOR problem.
3. Use Python to simulate the working of an ADALINE model with a small dataset.
4. Analyze the effect of different activation functions on neural output with graphical representation.

**Module 4:**

1. Develop a simple Self-Organizing Map (SOM) using a given 2D dataset and visualize the clustering.
2. Implement an Adaptive Neuro-Fuzzy Inference System (ANFIS) using any tool (Python, MATLAB, etc.) for a classification problem.
3. Write a comparative analysis of neural networks vs fuzzy logic vs neuro-fuzzy systems.
4. Case Study: Design and document a hybrid model for medical diagnosis, weather prediction, or stock trend prediction.

## Assessment

Internal Assessment – 50 Marks

**Evaluation Scheme: -** Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below..

| **The Rubric will have the following Evaluation Parameters:** | | |
|---|---|---|
| **Evaluation Parameters** | **Description / Evaluation Points** | **Marks** |
| Understanding of Concepts | Ability to clearly explain theoretical concepts related to fuzzy logic or neural networks. | 10 |
| Problem Solving / System Design | Design and structuring of fuzzy inference systems or neural models as per problem statements. | 10 |
| Code / Implementation Accuracy | Functionality and correctness of the developed model, code execution, and adherence to logic. | 10 |
| Report and Documentation | Clarity of report writing, logical structure, diagrams, input-output explanation, formatting. | 10 |
| Presentation & Timely Submission | Neatness, formatting, adherence to submission deadline, and overall presentation of work. | 10 |

Sample Projects

### 1. XOR Problem Using Single Layer and Multi-Layer Perceptron

- Objective: Train and compare a Perceptron and a Backpropagation Neural Network to solve the XOR logic problem.
- Tools: Python (TensorFlow, Keras, NumPy)
- Learning Outcome: Explore the limitations of single-layer networks and the significance of non-linearity and hidden layers.

### 2. Handwritten Digit Recognition using Neural Networks

- Objective: Build a neural network to classify handwritten digits (e.g., using the MNIST dataset).
- Tools: Python (Keras, TensorFlow, or PyTorch)
- Learning Outcome: Apply concepts of MLP, activation functions, backpropagation, and accuracy metrics.

## External Assessment: (Marks 50)

End Semester examination of 50 marks for 2 hours duration will be conducted.

## Major (Elective)

| | |
|---|---|
| **Course Title** | **Ethical Hacking** |
| **Course Credits** | **4 Credits** |

| | |
|---|---|
| | **CO1**- To introduce key concepts of ethical hacking and cybersecurity. |
| | **CO2**- To identify and analyze various security threats and vulnerabilities. |
| | **CO3**-To develop hands-on skills with ethical hacking tools and techniques. |
| | **CO4**- To understand legal frameworks and ethical responsibilities of security professionals. |
| **Module 1 (Credit 1)**<br>**Introduction to Ethical Hacking and Cyber Laws** | |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Grasp the fundamentals of ethical hacking, cyber threats, attack vectors, and hacker methodologies. |
| | Use tools and techniques for reconnaissance, vulnerability assessment, and basic penetration testing in a legal environment. |
| | Identify and implement appropriate security measures and countermeasures such as firewalls, IDS/IPS, and encryption. |
| | Understand key cyber laws, ethical issues, and the legal responsibilities of ethical hackers in national and global contexts. |
| **Content Outline** | ● Fundamentals of hacking: types of hackers, attack vectors<br>● Phases of ethical hacking: reconnaissance to covering tracks<br>● Cybersecurity principles: CIA triad, threat modeling<br>● Legal, professional, and ethical issues<br>● Overview of IT Act 2000 and amendments<br>● Cybercrime case studies (India and global) |
| **Module 2 (Credit 1)**<br>**Reconnaissance, Scanning & Vulnerability Assessment** | |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Distinguish between active and passive reconnaissance and explain their role in the ethical hacking lifecycle. |
| | Utilize tools and methods for port scanning, network mapping, and service detection to gather target system information. |
| | Identify, analyze, and classify system vulnerabilities using automated and manual assessment tools. |
| | Evaluate scan and assessment results to determine security risks and prepare actionable reports for remediation. |

| | |
|---|---|
| **Content Outline** | **Module Contents:**<br>● Footprinting techniques: passive & active reconnaissance<br>● Tools: Whois, Nslookup, Recon-ng, Maltego, Google Dorking<br>● Network scanning: types of scans (TCP, SYN, UDP), ping sweep<br>● Banner grabbing, OS fingerprinting<br>● Enumeration techniques: SNMP, NetBIOS, LDAP<br>● Vulnerability assessment using Nessus, OpenVAS |
| **Module 3 (Credit 1)**<br>**System Exploitation and Post-Exploitation Techniques** | |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Explain common exploitation techniques and vulnerabilities used to gain unauthorized access to systems. |
| | Use tools like Metasploit and other frameworks to exploit known vulnerabilities in a simulated environment. |
| | Demonstrate methods for privilege escalation, maintaining access, and data exfiltration after initial compromise. |
| | Evaluate the impact of successful exploitation and generate professional reports outlining findings and recommendations. |
| **Content Outline** | ● Exploiting system vulnerabilities: buffer overflow, password attacks<br>● Privilege escalation techniques<br>● Malware: trojans, viruses, rootkits<br>● Backdoors and maintaining access<br>● Covering tracks: log deletion, anti-forensics<br>● Tools: Metasploit, Hydra, John the Ripper |
| **Module 4 (Credit 1)**<br>**Web Application & Wireless Network Hacking** | |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Identify typical security flaws such as SQL injection, XSS, weak encryption, and insecure Wi-Fi protocols. |
| | Use appropriate tools and techniques to test web apps and wireless networks for vulnerabilities ethically. |

| | |
|---|---|
| | Recommend and apply security controls like input validation, secure configurations, and strong wireless encryption. |
| | Evaluate attack outcomes and prepare clear reports with mitigation strategies for stakeholders. |
| **Content Outline** | ● Web application vulnerabilities: SQLi, XSS, CSRF, file inclusion<br>● Secure coding practices<br>● Web vulnerability scanning tools: OWASP ZAP, Burp Suite<br>● Wireless security: WEP/WPA/WPA2 vulnerabilities<br>● Wireless hacking tools: Aircrack-ng, Kismet<br>● Network sniffing with Wireshark<br>● Introduction to penetration testing reports and documentation |

**Reference Books :**

1. Matt Walker, All-In-One-CEH-Certified-Ethical-Hacker-ExamGuide.
2. Tutorials Point Professionals, Ethical Hacking by Tutorials Point.
3. BIBLIOGRAPHY • Kimberly Graves(26th-April-2010), "CEH Certified Ethical Hacker Study Guide" 1st Edition, ISBN-13: 978-0470525203, ISBN-10: 0470525207, Sybex-Wiley Publishing. •
4. Sean-Philip Oriyano, Sybex, Certified Ethical Hacker Study Guide v9, Study Guide Edition, 2016.

**Assignments towards Comprehensive Continuous Evaluation**

**Module 1:**

5. Define ethical hacking. How does it differ from malicious hacking?
6. List and explain the phases of ethical hacking.
7. Discuss the legal and ethical considerations an ethical hacker must keep in mind before conducting a security test.
8. Discuss some real-world examples where ethical hacking helped prevent cyber attacks.
9. Explain the key provisions of the Information Technology Act, 2000
10. What is data protection law? How does it impact individuals and organizations?
11. What are the responsibilities of internet service providers (ISPs) under cyber law?
12. Explain the legal challenges in prosecuting cybercrimes across international borders.

**Module 2:**

1. What is reconnaissance in ethical hacking, and why is it considered the first step in the hacking process?
2. Differentiate between active and passive reconnaissance with suitable examples.
3. Explain the purpose of port scanning and name three tools used for this task.
4. What are the different types of scanning techniques used to identify vulnerabilities in a target system?
5. Define vulnerability assessment and explain how it helps in improving cybersecurity.
6. Describe the difference between vulnerability assessment and penetration testing.
7. What is the role of Common Vulnerabilities and Exposures (CVE) in vulnerability management?

**Module 3:**

1. What is system exploitation in ethical hacking? Explain its significance in penetration testing.
2. Describe the common types of system exploits used by ethical hackers.
3. What are buffer overflow attacks? How can they be exploited to gain system access?
4. Explain the concept of privilege escalation and its importance in post-exploitation.
5. What are backdoors and rootkits? How do they aid attackers after gaining system access?
6. Discuss the techniques used to maintain access to a compromised system during post-exploitation.
7. What is data exfiltration, and what methods do attackers use to extract sensitive information during post-exploitation?

**Module 4:**

1. What are the common vulnerabilities found in web applications? Explain with examples.
2. Describe the SQL Injection attack and how it can be prevented in web applications.
3. Explain Cross-Site Scripting (XSS) and its impact on web security.
4. What tools and techniques are commonly used for hacking wireless networks?
5. Discuss the differences between WEP, WPA, and WPA2 encryption in wireless security.
6. What is a man-in-the-middle (MITM) attack in wireless networks and how can it be mitigated?
7. Explain how session hijacking works in web applications and the measures to prevent it.

**Assessment:**
**Internal Assessment – 50 Marks**

**The Rubric will have the following Evaluation Parameters:**

| Criteria | Excellent (9-10 marks) | Good (7-8 marks) | Satisfactory (5-6 marks) | Needs Improvement (0-4 marks) | Marks |
|---|---|---|---|---|---|
| 1. Understanding of Ethical Hacking Concepts | Clear, in-depth understanding of ethical hacking principles, methodologies, legal and ethical considerations | Good understanding with minor gaps; some awareness of ethical/legal issues | Basic understanding, some misconceptions about ethical/legal aspects | Poor or incorrect understanding of ethical hacking concepts | 10 |
| 2. Practical Skills and Tools Usage | Proficient use of multiple hacking tools (e.g., Nmap, Metasploit), demonstrating advanced techniques | Effective use of common tools with some advanced techniques | Basic usage of standard tools; limited application of techniques | Inadequate or incorrect use of tools; unable to demonstrate skills | 10 |
| 3. Vulnerability Analysis & Reporting | Thorough and detailed vulnerability assessment with clear, actionable, and well-documented reports | Good vulnerability analysis with acceptable documentation | Basic identification of vulnerabilities; report lacks depth | Poor or incomplete vulnerability assessment and reporting | |
| 4. Problem Solving & Critical Thinking | Demonstrates excellent analytical skills; proposes innovative and effective mitigation | Good problem-solving ability; mitigation strategies mostly effective | Identifies problems but solutions lack depth or feasibility | Unable to analyze or suggest practical solutions | 10 |

| | | | | | |
|---|---|---|---|---|---|
| | strategies | | | | |
| 5. Ethical & Legal Compliance Awareness | Demonstrates strong awareness of ethical standards and legal frameworks; adheres strictly to guidelines | Adequate awareness and compliance with ethical/legal standards | Some understanding but inconsistent application | Little or no awareness of ethical/legal considerations | 10 |
| 6. Communication & Presentation | Presents work clearly and professionally with excellent organization and technical language | Clear presentation with minor clarity or organization issues | Basic communication; some ambiguity or lack of detail | Poor communication; unclear or disorganized presentation | 10 |

**External Assessment: (Marks 50)**

End Semester examination of 50 marks for 2 hours duration will be conducted.

Major (Elective)

| | |
|---|---|
| **Course Title** | **Natural Language Processing** |
| **Course Credits** | **4 Credits** |
| **Course Outcomes** | **CO1**:- Understand the fundamentals of natural language processing and its role in artificial intelligence and machine learning. |
| | **CO2**:-Explain the structure of language using linguistic concepts such as morphology, syntax, semantics, and pragmatics. |
| | **CO3**:- Apply basic NLP techniques such as tokenization, stemming, lemmatization, POS tagging, and parsing to process raw text data. |
| | **CO4**:- Implement and evaluate classical NLP models including n-gram models, TF-IDF, and word embeddings (e.g., Word2Vec, GloVe). |

| | |
|---|---|
| | **CO5:**- Build machine learning models for NLP tasks such as text classification, sentiment analysis, and named entity recognition using libraries like NLTK, SpaCy, and Scikit-learn. |
| **Module 1 (Credit 1)** | **Introduction and Foundational Concepts** |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | •Define the significance and necessity of natural language processing (NLP) in modern computing and communication systems.<br>• How the text preprocessing is perform<br>• Understand the construction of regular expression<br>• Understand corpus and utilization of corpus in NLP |
| **Content Outline** | **What is NLP?** - Definition, applications, and the role of NLP in the broader field of artificial intelligence and computational linguistics.<br>**Text Processing**: -Tokenization, stemming, lemmatization, and stop word removal.<br>**Regular Expressions**:-Understanding and using regular expressions for text manipulation.<br>**Corpora and Resources**:- Introduction to commonly used text corpora and NLP resources.<br>**Python Libraries**:- Overview of Python libraries like NLTK, spaCy, and others commonly used in NLP. |
| **Module 2 (Credit 1 ) Core NLP Techniques** | |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | 1. Differentiate between various levels of language processing (phonetics, syntax, and semantics) and comprehend their role in NLP.<br>2. Identify and analyze the key issues, challenges, and limitations in natural language processing.<br>3. Analyze and discuss diverse applications of NLP across various domains such as machine translation, sentiment analysis, information retrieval, catboats, and more. |

| | |
|---|---|
| **Content Outline** | **Language Modeling**: - Statistical language modeling, n-grams, smoothing techniques, and perplexity.<br>**Part-of-Speech (POS) Tagging**: - Algorithms for assigning grammatical tags (noun, verb, etc.) to words. Type of POS Tagging<br>**Parsing**:-Understanding syntactic structures, constituency parsing (phrase structure trees), and dependency parsing.<br>**Types of parsing**:<br>**Constituency and Dependency Parsing.**<br>**Constituency Parsing**:- Top-down parsing (e.g., recursive descent, predictive parsing), Bottom-up parsing (e.g., shift-reduce parsing).<br>**Dependency Parsing:-**Dependency relations and their representation, Transition-based dependency parsing (e.g., Arc-Standard, Arc-Eager).<br>Graph-based dependency parsing.<br>**Semantic Analysis**:- Lexical semantics, word sense disambiguation, distributional semantics (word embedding), and semantic role labeling. |
| **Module 3 (Credit 1)** | **Morphology and Techniques of POS Tagging** |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | 1. Define the morphology, the different types of morphology, and its process and differentiate between inflectional and derivational morphology, recognizing their significance in understanding word formation and structure.<br>2. Define and categorize word classes, recognizing their significance in linguistic analysis and natural language understanding |

| | |
|---|---|
| **Content Outline** | **Introduction to Morphology**: Definition of morphology, its role in NLP, and its relationship to other linguistic levels.<br>**Morphemes and Allomorphs**: Understanding the concept of morphemes (the smallest meaningful units of language) and allomorphs (different forms of the same morpheme).<br>**Morphological Processes:**<br>**Inflectional Morphology**: Changes in word form to express grammatical features (e.g., tense, number)<br>**Derivational Morphology**: Creation of new words with different meanings or grammatical categories<br>**Compositional Morphology**: Combining words to form new words (e.g., compound nouns)<br>**Techniques for POS Tagging:**<br>o Rule-based POS tagging, Statistical POS tagging:-Using Hidden Markov Models (HMMs).<br>o Maximum Entropy Markov Models (MEMMs) and Conditional Random Fields (CRFs). |
| **Module 4 (Credit 1)** | **Applications and Advanced Topics (Examples)** |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | Implement the NLP knowledge |
| **Content Outline** | **Information Retrieval**: Building search engines and understanding information retrieval techniques.<br>**Text Classification**: Classifying text into different categories (e.g., spam detection, sentiment analysis). |

**References:-**

1. Foundations of Statistical Natural Language Processing  Authors: Christopher D. Manning and Hinrich Schütze,  Publisher: MIT Press
2. Natural Language Processing with Python (a.k.a. the NLTK Book), Authors: Steven Bird, Ewan Klein, and Edward Loper, Publisher: O'Reilly Media

3. Natural Language Processing: A Practical Guide for Beginners, Author: Rajesh Arumugam, Rajalingappaa Shanmugamani, Publisher: Packt Publishing
4. Practical Natural Language Processing ,Authors: Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, Harshit Surana ,Publisher: O'Reilly Media
5. Transformers for Natural Language Processing , Author: Denis Rothman,Publisher: Packt Publishing

## Assignments/Activities towards Comprehensive Continuous Evaluation (CCE) Classroom Activities

**Module 1: Introduction to NLP**

Activity 1**:** Short quiz or MCQs on NLP basics, applications, and language models. Activity 2: Group presentation: "Applications of NLP in real-world domains (health,
finance, etc.)"

Activity 3: **Crossword** or flashcards for terminology (tokens, corpus, lexicon, etc.)

### Module 2: Text Preprocessing and Linguistic

Activity 1: Write a Python program using NLTK or SpaCy to tokenize, clean, and preprocess raw text

Activity 2: Annotate a paragraph using POS tagging and lemmatization using SpaCy Activity 3: Compare outputs of stemming vs lemmatization on a given dataset

### Module 3: Language Modeling and Feature Extraction

Activity 1: Implement unigram, bigram, and trigram models for a corpus and compute TF-IDF values.

Activity 2: Visualize and interpret word frequency and TF-IDF scores for a news or review dataset

### Module 4 : Text Classification & Sentiment Analysis

Activity 1**:** Build a text classification model (e.g., spam vs ham, positive vs negative) using Scikit-learn

Activity 2: Compare Naive Bayes vs Logistic Regression for the same dataset.

**Assessment**

Internal
Assessment – 50

Marks External
Assessment-50
Marks

**Evaluation Scheme: -** Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

| **The Rubric will have the following Evaluation Parameters:** | | |
|---|---|---|
| **Evaluation Parameters** | **Description / Evaluation Points** | **Mark S** |
| Apply | Code Implementation & Execution. Writing functional, modular code for NLP tasks | 10 |
| Understand And Apply | Text Preprocessing and NLP Techniques. Using tokenization, lemmatization, POS tagging, etc. appropriately | 10 |
| Apply | Use of NLP Libraries (NLTK, SpaCy, etc.) Applying functions and tools from standard NLP libraries | 10 |
| Analyze / Evaluate | Model Development (ML/NLP Models). Selecting and building appropriate models based on problem Evaluation Metrics & Interpretation - Using accuracy, precision, recall, F1-score and interpreting results | 15 |
| | Completeness, correctness, formatting, and timely submission | 5 |
| | Total | 50 |

(Given below are sample projects but it is expected to work on similar sort of projects)

**Project Work (Summative)**

- Select a dataset (Kaggle/UCI).
- Perform text cleaning , build ML models, visualize results.
- Submit report + present findings.
- Evaluation Criteria:
- Problem statement clarity
- Code and data analysis
- Interpretation and insights
- Oral presentation & report

# Second Year

## Semester: III

| | |
|---|---|
| **Course Code** | **3.1 (Major Core)** |
| **Course Title** | **Cloud Computing** |
| **Course Credits** | **4** |
| **Course Outcomes** | **After Completion of this Course, students will be able to**<br>• Identify security aspects of each cloud model.<br>• Develop a risk-management strategy for migrating to the Cloud. Implement a public cloud instance with a public cloud service provider.<br>• Apply a trust-based security model to different layers. |
| **Module 1: Introduction to Cloud Computing (Credit1)** | |
| **Learning Outcomes** | **After learning this module, learners will be able to**<br>• Define cloud computing and its key characteristics, service models, and deployment models.<br>• Provide an overview of the historical development of cloud computing and its evolution from traditional models.<br>• Compare major cloud service providers like AWS, Azure, and GCP.<br>• Identify and analyze potential security risks and challenges in cloud computing. |
| **Content Outline** | • Introduction to Cloud Computing<br>• Online Social Networks and Applications<br>• Cloud introduction and overview<br>• Different clouds, Risks, Novel applications of cloud computing |
| **Module 2: Cloud Computing Architecture, Cloud Deployment Models (Credit1)** | |
| **Learning Outcomes** | **After learning this module, learners will be able to**<br>• Define the requirements driving the emergence of cloud computing and explain CPU virtualization's role.<br>• Provide an overview of basic cloud computing principles, discuss hypervisors, and explain the SPI framework<br>• Identify key drivers motivating cloud adoption and assess the impact on end-users and businesses.<br>• Explore best practices for establishing effective governance structures in cloud environments. |

| | |
|---|---|
| **Content Outline** | • Cloud Computing Architecture: Requirements, Introduction Cloud computing architecture, On Demand Computing Virtualization at the infrastructure level, Security in Cloud computing environments, CPU Virtualization, A discussion on Hypervisors Storage Virtualization Cloud Computing Defined, The SPI Framework for Cloud Computing, The Traditional Software Model, The Cloud Services Delivery Model<br>• Cloud Deployment Models: Key Drivers to Adopting the Cloud, The Impact of Cloud Computing on Users, Governance in the Cloud, Barriers to Cloud Computing Adoption in the Enterprise |
| **Module 3: Security Issues in Cloud Computing and Access management (Credit1)** | |
| **Learning Outcomes** | **After learning the module, learners will be able to**<br>• Apply knowledge of infrastructure security in cloud computing to identify key components involved.<br>• Analyse network-level security measures and protocols relevant to cloud environments.<br>• Evaluate application-level security practices and challenges specific to cloud-based applications.<br>• Design strategies to ensure data security and storage in cloud computing environments. |
| **Content Outline** | • **Security Issues in Cloud Computing:** Infrastructure Security, Infrastructure Security: The Network Level, The Host Level, The Application Level, Data Security and Storage, Aspects of Data Security, Data Security Mitigation Provider Data and Its Security.<br>• **Identity and Access Management:** Trust Boundaries and IAM, IAM Challenges, Relevant IAM Standards and Protocols for Cloud Services, IAM Practices in the Cloud, Cloud Authorization Management. |
| **Module 4: Security Management in the Cloud, Privacy Issues (Credit1)** | |
| **Learning Outcomes** | **After learning the module, learners will be able to** |
| | • Apply security management standards relevant to cloud computing to ensure robust security measures.<br>• Analyse availability management practices for SaaS, PaaS, and IaaS to optimize service availability.<br>• Evaluate risk assessments specific to cloud security and propose effective mitigation strategies.<br>• Design and implement tailored incident response plans for cloud computing scenarios to ensure swift and effective responses. |
| **Content Outline** | • Security Management in the Cloud: Security Management Standards, Security Management in the Cloud, Availability Management: SaaS, PaaS, IaaS.<br>• Privacy Issues: Privacy Issues, Data Life Cycle, Key Privacy Concerns in the Cloud, Protecting Privacy, Changes to Privacy Risk Management and Compliance in Relation to Cloud Computing, Legal and Regulatory Implications, U.S. Laws and Regulations, International Laws and Regulations. |

**Activities towards Comprehensive Continuous Evaluation (CCE)**

**These assignments aim to apply theoretical concepts to practical application and critical thinking.**

**Module 1:**

- Research and analyze the evolution of cloud computing and its impact on modern technological landscapes. Identify and explore various online social networks and applications that leverage cloud computing technologies. Develop a comprehensive overview of cloud computing, highlighting different types of clouds, associated risks, and novel applications in diverse domains.

**Module 2:**

- Investigate the architecture of cloud computing systems, focusing on requirements and essential components. Explore virtualization at the infrastructure level and its role in cloud computing. Evaluate security measures implemented in cloud environments. Examine various cloud deployment models and discuss key drivers influencing cloud adoption.

**Module 3:**

- Conduct a comprehensive assessment of security issues inherent in cloud computing, covering infrastructure, network, host, and application levels. Explore data security and storage considerations, including mitigation strategies. Investigate identity and access management (IAM) challenges and relevant standards and protocols for cloud services.

**Module 4:**

- Examine security management standards and practices applicable to cloud computing environments. Analyze availability management for different cloud service models (SaaS, PaaS, IaaS). Investigate privacy concerns related to cloud computing, including data life cycle, key privacy concerns, and compliance with legal and regulatory frameworks.

**Evaluation Scheme:**

**Internal Assessment:**

Depending on the activities mentioned above a project should be developed of 50 marks. Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

The Rubric will have the following Evaluation Parameters:

| Evaluation Parameters | Marks |
|---|---|
| Functionality (Core Features) | 20 |
| Code Quality (Structure, Readability) | 10 |
| GUI Design & User Experience | 10 |
| Input Handling & Error Management | 5 |
| Report writing or Documentation | 5 |

The project Internal assessment will be done by conducting project presentation at the department level, where External Examiner appointed by the department will be evaluating the project.

One sample projects are given below and it is expected from students to choose a good and advance project topic that aligns with their interests and abilities to work on.

Project 1: Develop and deploy a simple web application (e.g., a student feedback system) on the cloud using AWS services. The app must be scalable, fault-tolerant, and include monitoring and logging.

**External Assessment:**

End Semester examination of 50 marks for 2 hours duration will be conducted by the University.

**References:-**

1. T., Mahmood, Z., &Puttini, R. (2013). Cloud Computing: Concepts, Technology & Architecture. Prentice Hall.
2. Reese, G. (2009). Cloud Application Architectures: Building Applications and Infrastructure in the Cloud. O'Reilly Media.
3. Mather, T., Kumaraswamy, S., & Latif, S. (2009). Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance. O'Reilly Media.
4. Bahga, A., & Madisetti, V. (2014). Cloud Computing: A Hands-On Approach.CreateSpace Independent Publishing Platform.

| | |
|---|---|
| **Course Code** | **3.2 (Major Core)** |
| **Course Title** | **BIG DATA ANALYTICS** |
| **Course Credits** | **4** |
| **Course Outcomes** | **After completion of this Course, the students will be able to**<br>• Apply big data analytics approaches, including conceptualization, summarization, and machine learning techniques.<br>• Analyze the characteristics of datasets to distinguish between trivial data and big data for various applications.<br>• Evaluate solutions for problems related to batch learning, online learning, and big data characteristics like high dimensionality and scalability issues.<br>• Design scalable solutions to manage and process dynamically growing big data. |
| **Module 1: Introduction (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Apply understanding of fundamental Big Data concepts to address modern data challenges effectively.<br>• Analyze traditional and Big Data approaches to select appropriate technologies for large dataset processing and analysis.<br>• Evaluate infrastructure requirements for efficient Big Data handling, focusing on data analytics for valuable insights extraction.<br>• Design strategies to overcome challenges inherent in Big Data, considering volume, velocity, and variety factors for effective data management and analysis |
| **Module Contents** | • Introduction to Big Data, Big Data Characteristics, Types of Big Data |

| | |
|---|---|
| | • Traditional Versus Big Data Approach, Technologies Available for Big Data<br>• Infrastructure for Big Data, Use of Data Analytics, Big Data Challenges, Desired Properties of a Big Data System, Case Study of Big Data Solutions |
| **Module 2: Analytical Theory and Methods (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Apply clustering algorithms, association rules, and the Apriori Algorithm to recognize patterns effectively across diverse datasets.<br>• Analyse real-world scenarios by applying association rules, mastering validation, and testing for model reliability.<br>• Evaluate linear and logistic regression models, implementing and interpreting regression analyses with proficiency.<br>• Design additional regression models to enhance analytical capabilities for addressing diverse data- driven challenges. |
| **Module Contents** | • Clustering and Associated Algorithms, Association Rules, Apriori Algorithm, Candidate Rules<br>• Applications of Association Rules, Validation and Testing, Diagnostics, Regression, Linear Regression, Logistic Regression, Additional Regression Models |
| **Module 3: Hadoop (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>Apply foundational principles to comprehend Hadoop, including its definition, core components, and the role of operating systems in Big Data processing.<br>• Analyse Hadoop architecture, ecosystem components, and technologies like Hive to develop proficiency and understanding.<br>• Evaluate the limitations of Hadoop while exploring practical applications, particularly in recommendation systems.<br>• Design practical strategies to address the limitations of Hadoop and optimize its application in real-world scenarios. |
| **Module Contents** | • Introduction, What is Hadoop?, Core Hadoop Components, Operating System for Big Data<br>• Concepts, Hadoop Architecture, Hadoop Ecosystem, Hive, , Hadoop Limitations , Recommendation Systems. |
| **Module 4: NoSQL (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Apply comprehensive learning to grasp NoSQL, covering its definition, business drivers, case studies, and data architectural patterns for informed decision-making in data management.<br>• Analyse the practical application of MapReduce within the new software stack, understanding its role and implementing algorithms for efficient big data processing.<br>• Evaluate the effectiveness of MapReduce in handling largescale data processing tasks, considering its scalability and performance. |

|  |  |
|---|---|
|  | - Design strategies to optimize MapReduce algorithms and workflows for specific big data processing requirements, ensuring efficient and effective data processing. |
| **Module Contents** | - What is NoSQL?, NoSQL Business Drivers, NoSQL Case Studies, NoSQL Data Architectural Patterns<br>- Variations of NoSQL Architectural Patterns, Using NoSQL to Manage Big Data<br>- Map Reduce: MapReduce and The New Software Stack, MapReduce, Algorithms Using MapReduce |

**Activities towards Comprehensive Continuous Evaluation**

These assignments aim to apply theoretical concepts to practical application and critical thinking.

**Module 1:**

- Students will conduct research on clustering algorithms, association rules, and the Apriori Algorithm.

**Module 2:**

- They will analyze real-world applications of association rules and perform validation and testing of these algorithms.

**Module 3:**

- Students will delve into regression analysis, including linear regression, logistic regression, and additional regression models.

**Module 4:**

- They will evaluate the effectiveness and limitations of regression models in different data-driven scenarios.
- Finally, students will design and implement regression models for specific analytical tasks, applying the learned theories and methods.

**Evaluation Scheme:**

**Internal Assessment:**

Depending on the activities mentioned above a project should be developed of 50 marks. Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

| The Rubric will have the following Evaluation Parameters: |  |
|---|---|
| **Evaluation Parameters** | **Marks** |
| Functionality (Core Features) | 20 |
| Code Quality (Structure, Readability) | 10 |
| GUI Design & User Experience | 10 |
| Input Handling & Error Management | 5 |
| Report writing or Documentation | 5 |

The project Internal assessment will be done by conducting project presentation at the department level, where External Examiner appointed by the department will be evaluating the project.

Two sample projects are given below and it is expected from students to choose a good and advance project topic that aligns with their interests and abilities to work on.

Project 1: Use Apache Spark to analyze a large crime dataset from a major city (e.g., Chicago Crime Data). Extract trends, frequent crimes, and visualize hotspots.

Project 2: Analyze a large dataset of COVID-19-related tweets to find trending hashtags, sentiment patterns, and regional discussions.

**External Assessment:**

End Semester examination of 50 marks for 2 hours duration will be conducted by the University.

**References:**

1. Shankarmani, R. (2016). Big Data Analytics. Wiley.
2. Chellappan, S., & Acharya, S. (2015). Big Data and Analytics. Wiley.
3. Prajapati, V. (2013). Big Data Analytics with R and Hadoop. Pack Publishing.
4. Dasgupta, N. (2018). Practical Big Data Analytics. Pack Publishing.

| | |
|---|---|
| **Course Code** | **3.3 (Major Core)** |
| **Course Title** | **Machine Learning** |
| **Course Credits** | **4** |
| **Course Outcomes** | **After Completion of this Course, students will be able to**<br>• Understand the foundational concepts and algorithms of Machine Learning.<br>• Implement binary, multiclass classification and regression models using real-world datasets.<br>• Apply logic-based and algebraic models to structured learning problems.<br>• Develop end-to-end ML solutions including preprocessing, model selection, evaluation, and reporting.<br>• Explore recent trends and advanced topics in Machine Learning. |
| **Module 1: Introduction (Credit 1)** | |
| **Learning Outcomes** | **After learning this module, learners will be able to**<br>• Explain basic concepts of supervised, unsupervised, and reinforcement learning.<br>• Identify real-world problems where ML can be applied.<br>• Understand data preprocessing techniques. |

| | |
|---|---|
| **Content Outline** | • What is Machine Learning? Types and Applications<br>• Supervised vs Unsupervised Learning<br>• Data Cleaning, Feature Scaling, Label Encoding<br>• Overfitting and Underfitting<br>• Bias-Variance Tradeoff<br>• Introduction to Scikit-Learn |
| **Module 2: Binary and Multiclass Classification (Credit 1)** | |
| **Learning Outcomes** | **After learning this module, learners will be able to**<br>• Build and evaluate classification models.<br>• Handle imbalanced datasets and multiclass problems. |
| **Content Outline** | • Binary Classification: Logistic Regression, Naïve Bayes, k-NN<br>• Multiclass Classification: One-vs-Rest, One-vs-One strategies<br>• Support Vector Machines (Linear & Kernel)<br>• Model Evaluation: Confusion Matrix, Accuracy, Precision, Recall, F1 Score, ROC-AUC<br>• Cross-Validation and Grid Search |
| **Module 3: Regression (Credit1)** | |
| **Learning Outcomes** | **After learning the module, learners will be able to**<br>• Develop and evaluate regression models for prediction tasks. |
| **Content Outline** | • Simple and Multiple Linear Regression<br>• Polynomial Regression<br>• Decision Tree Regression<br>• Evaluation Metrics: MSE, RMSE, MAE, R$^2$ Score<br>• Feature Selection & Regularization: Lasso, Ridge, ElasticNet |
| **Module 4: Logic Based and Algebraic Models, Trends in Machine Learning (Credit1)** | |
| **Learning Outcomes** | **After learning the module, learners will be able to**<br>• Understand logic-based approaches and ensemble methods.<br>• Identify and explore current trends in ML. |
| **Content Outline** | • Decision Trees, Rule-Based Systems<br>• Ensemble Methods: Bagging, Random Forest, Boosting (AdaBoost, XGBoost)<br>• Introduction to Clustering: K-Means, Hierarchical<br>• Dimensionality Reduction: PCA, t-SNE<br>• Trends: AutoML, Explainable AI (XAI), MLops Basics, Ethical AI |
| **Activities towards Comprehensive Continuous Evaluation (CCE)** | |
| **Module 1:** Preprocessing real datasets using Python (Mini-lab)<br>**Module 2:** Implement binary and multiclass classification models on UCI datasets<br>**Module 3:** Predict housing prices using regression techniques<br>**Module 4:** Compare ensemble models on real data and present findings | |

**Evaluation Scheme:**

**Internal Assessment:**

Depending on the activities mentioned above a project should be developed of 50 marks. Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

| The Rubric will have the following Evaluation Parameters: | |
|---|---|
| **Evaluation Parameters** | **Marks** |
| Functionality (Core Features) | 20 |
| Code Quality (Structure, Readability) | 10 |
| GUI Design & User Experience | 10 |
| Input Handling & Error Management | 5 |
| Report writing or Documentation | 5 |

The project Internal assessment will be done by conducting project presentation at the department level, where External Examiner appointed by the department will be evaluating the project.

Two sample projects are given below and it is expected from students to choose a good and advance project topic that aligns with their interests and abilities to work on.

Project 1: Disease Diagnosis using ML (Multi-class classification)

Project 2: Stock Price Prediction using Regression Techniques

**External Assessment:**

End Semester examination of 50 marks for 2 hours duration will be conducted by the University.

**References:-**

1. Tom Mitchell, Machine Learning, McGraw Hill
2. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow, O'Reilly
3. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning
4. Jason Brownlee, Machine Learning Mastery with Python
5. Python and Scikit-learn official documentation

| | |
|---|---|
| **Course Code** | **3.4 (Major Core)** |
| **Course Title** | **NATURAL LANGUAGE PROCESSING** |
| **Course Credits** | **2** |
| **Course Outcomes** | **After Completion of this Course, students will be able to**<br>• Provide an understanding of text processing for Natural Language Understanding and Natural Language Generation.<br>• Cover various techniques available for natural language processing.<br>• Introduce different approaches and algorithms for carrying out NLP |

| | |
|---|---|
| | tasks.<br>• Explore the know-hows, issues, and challenges in NLP applications and their relevance in classical and modern contexts. |
| **Module 1: INTRODUCTION TO NLP (Credit 1)** | |
| **Learning Outcomes** | **After learning this module, learners will be able to**<br>• Foster an understanding of Natural Language Understanding and Natural Language Generation.<br>• Equip learners with practical skills in NLTK and spaCy, focusing on computing fundamentals with languages.<br>• Master various text processing techniques, including Unicode, regular expressions, tokenization, stemming, lemmatization, segmentation, and formatting. |
| **Content Outline** | • Natural Language Processing: What is Natural Language Understanding and Natural Language Generation, Introduction to NLTK, spaCy, Computing with Languages – Text and Words, Searching Text counting vocabulary, List , Strings , Variable, Computing frequency Distribution<br>• Accessing Text Corpora, Lexical Resources and Processing Raw Text: Introduction to Corpora, Conditional Frequency Distribution, Lexical Resources, Accessing text from web, Text Processing using Strings : Unicode, Regular Expressions Normalizing Text :Tokenizing Text, Stemming, Lemmatization, Segmentation, Formatting |
| **Module 2:** TAG AND TEXT **(Credit 1)** | |
| **Learning Outcomes** | **After learning this module, learners will be able to**<br>• Develop expertise in categorizing and tagging words through taggers and Python dictionaries.<br>• Introduce the application of machine learning algorithms like Decision Trees and Naïve Bayes Classifier for text classification.<br>• Provide an understanding of supervised classification processes and skills to evaluate classifier performance. |
| **Content Outline** | • Categorizing and Tagging Words: Using a Taggers, Tagged Corpora Mapping words to properties using Python Dictionaries ,Tagging, How to determine category of a word<br>• Learning to Classify Text: Using Machine Learning Algorithms to create classifiers, Supervised Classification, Decision Tree, Naïve Bayes Classifier, and Evaluation of the Classifier. |
| **Module 3:** INFORMATIONAND SENTENCE ANALYSIS **(1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Cultivate expertise in extracting information from text through Information Extraction, Chunking, Named Entity Recognition, and Relation Extraction.<br>• Explore the complexities of analyzing sentence structures, including grammatical nuances, syntax utilization, contextfree grammar, parsing, and dependency grammar. |

| | |
|---|---|
| | • Provide practical skills in building and assessing chunkers and managing recursion in linguistic structure. |
| **Module Contents** | • Extracting Information from the Text: Information Extraction :Information Extraction, Chunking, Developing and Evaluating Chunkers, Recursion in Linguistic Structure, Named Entity Recognition, Relation Extraction<br>• Analysing, Sentence Structure: Grammatical Dilemmas, What's the use of syntax? Context free Grammar, Parsing with Context free Grammar, Dependency and Dependency Grammar |
| **Module 4:** Building feature based Grammar **(1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Enhance proficiency in constructing feature-based grammar, encompassing grammatical features and processing feature structures.<br>• Investigate sentence semantics, including Natural Language Understanding, Propositional Logic, First Order Logic (Predicate Logic), and Discourse Semantics.<br>• Foster comprehension of organizational learning and transformation, while extending feature-based grammar to augment linguistic analysis capabilities. |
| **Module Contents** | • Building feature based Grammar: Grammatical Features, Processing Feature Structures Organizational Learning and Transformation, Extending a FeatureBased Grammar<br>• Analysing the Meaning of Sentences: Natural Language Understanding, Propositional Logic, First Order Logic (Predicate Logic)The Semantics of English Sentences, Discourse Semantics |
| **Assignments/ Activities towards Comprehensive Continuous Evaluation** | |
| These assignments aim to apply theoretical concepts to practical application and critical thinking.<br>**Module 1:**<br>• Apply advanced Natural Language Processing (NLP) techniques to analyze and extract information from a given dataset.<br>**Module 2 :**<br>• Build a feature-based grammar for a specific language or domain of interest.<br>**Module 3:**<br>• Analyze the meaning of sentences using propositional logic.<br>• Investigate and apply discourse semantics to a set of interconnected sentences.<br>**Module 4:**<br>• Develop an NLU system for a specific application (e.g., sentiment analysis, information extraction). | |

**Evaluation Scheme:**

**Internal Assessment:**

Depending on the activities mentioned above a project should be developed of 50 marks. Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

| The Rubric will have the following Evaluation Parameters: | |
|---|---|
| **Evaluation Parameters** | **Marks** |
| Functionality (Core Features) | 20 |
| Code Quality (Structure, Readability) | 10 |
| GUI Design & User Experience | 10 |
| Input Handling & Error Management | 5 |
| Report writing or Documentation | 5 |

The project Internal assessment will be done by conducting project presentation at the department level, where External Examiner appointed by the department will be evaluating the project.

Two sample projects are given below and it is expected from students to choose a good and advance project topic that aligns with their interests and abilities to work on.

Project 1: Build a sentiment analysis system to classify customer reviews as positive, negative, or neutral using NLP techniques and supervised machine learning.

Project 2: Design a tool that extracts named entities (person, organization, location) and their relationships (e.g., "X is CEO of Y") from a set of news articles.

**External Assessment:**

End Semester examination of 50 marks for 2 hours duration will be conducted by the University.

**References:**

1. Indurkhya, N., & Damerau, F. J. (2010). Handbook of Natural Language Processing (2nd ed.). CRC Press Taylor and Francis Group.
2. Manning, C., & Schutze, H. (2009). Natural Language Processing With Python. Wiley Publications.

| | |
|---|---|
| **Course Code** | **3.5.1** |
| **Course Title** | **Generative AI and LLMs(Elective-III)** |
| **Course Credits** | **4** |
| **Course Outcomes** | **After Completion of this Course, students will be able to**<br>• Understand the foundations and architectures of Generative AI.<br>• Explore the workings of Transformer-based Large Language Models (LLMs).<br>• Implement text generation and fine-tuning tasks using pre-trained models.<br>• Evaluate generative models for creativity, coherence, and factual accuracy.<br>• Analyze challenges in LLMs including bias, ethics, and alignment. |
| **Module 1: Fundamentals of Generative AI (Credit 1)** | |
| **Learning Outcomes** | **After learning this module, learners will be able to**<br>• Understand the principles of generative modeling.<br>• Explore early generative models and their applications. |
| **Content Outline** | • Introduction to Generative Models: VAEs, GANs, Autoregressive Models<br>• Applications of Generative AI in Text, Image, Audio, and Video<br>• Evaluation Metrics: BLEU, ROUGE, Perplexity<br>• Introduction to Generative Tasks: Text Completion, Summarization, Image Generation |
| **Module 2: Transformer Architecture and LLMs (Credit 1)** | |
| **Learning Outcomes** | **After learning this module, learners will be able to**<br>• Explain the architecture of transformers.<br>• Understand the design and training of LLMs. |
| **Content Outline** | • Transformer Architecture: Self-Attention, Positional Encoding<br>• GPT, BERT, T5: Overview and Comparisons<br>• Pretraining and Fine-tuning Paradigms<br>• Transfer Learning and Zero-shot/Few-shot Learning<br>• Popular LLMs: GPT-3/4, PaLM, LLaMA, Falcon, Gemini |
| **Module 3: Practical Applications of LLMs (Credit1)** | |
| **Learning Outcomes** | **After learning the module, learners will be able to**<br>• Apply LLMs to real-world natural language tasks. |
| **Content Outline** | • Text Generation, Summarization, Translation<br>• Chatbots and Conversational AI<br>• Prompt Engineering Techniques<br>• Fine-tuning LLMs with Custom Datasets<br>• Tools: Hugging Face Transformers, LangChain, OpenAI API |
| **Module 4: Ethics, Challenges, and Future Directions (Credit1)** | |
| **Learning Outcomes** | **After learning the module, learners will be able to**<br>• Analyze risks and challenges associated with LLMs.<br>• Explore current research and future trends. |
| **Content Outline** | • Ethical Issues: Bias, Misinformation, Deepfakes<br>• Model Alignment and Interpretability |

| | |
|---|---|
| | - Scaling Laws and Model Optimization<br>- Responsible AI and Policy Regulations<br>- Future of Generative AI and Multimodal LLMs |
| **Activities towards Comprehensive Continuous Evaluation (CCE)** | |
| **Module 1:** Evaluate generative output quality from classic models<br>**Module 2:** Visualize transformer components using sample inputs<br>**Module 3:** Build a chatbot using GPT-3 or Hugging Face models<br>**Module 4:** Case study on ethical use of LLMs in social media moderation | |

**Evaluation Scheme:**

**Internal Assessment:**

Depending on the activities mentioned above a project should be developed of 50 marks. Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

| The Rubric will have the following Evaluation Parameters: | |
|---|---|
| **Evaluation Parameters** | **Marks** |
| Functionality (Core Features) | 20 |
| Code Quality (Structure, Readability) | 10 |
| GUI Design & User Experience | 10 |
| Input Handling & Error Management | 5 |
| Report writing or Documentation | 5 |

The project Internal assessment will be done by conducting project presentation at the department level, where External Examiner appointed by the department will be evaluating the project.

Two sample projects are given below and it is expected from students to choose a good and advance project topic that aligns with their interests and abilities to work on.

Project 1: Domain-specific Chatbot using LangChain and LLM APIs

Project 2: LLM-Based Code Generator for Python Tasks

**External Assessment:**

End Semester examination of 50 marks for 2 hours duration will be conducted by the University.

**References:-**

1. Ian Goodfellow et al., Deep Learning, MIT Press
2. Vaswani et al., Attention is All You Need (Transformer Paper)
3. OpenAI, Anthropic, Google DeepMind – LLM Research Papers
4. Hugging Face Course and Documentation
5. LangChain, OpenAI API, Cohere AI Developer Guides

| | |
|---|---|
| **Course Code** | **3.5.2** |
| **Course Title** | **Graph Neural Network (GNN) (Elective-III)** |
| **Course Credits** | **4** |
| **Course Outcomes** | **After Completion of this Course, students will be able to**<br>• Understand graph theory concepts and represent data using graphs.<br>• Implement basic graph algorithms and apply neural networks on graph-structured data.<br>• Analyze Graph Neural Network architectures such as GCN, GAT, GraphSAGE, and others.<br>• Design GNN-based models for node classification, link prediction, and graph classification tasks.<br>• Explore recent trends, tools, and applications of GNNs in real-world domains. |
| **Module 1: Foundations of Graph Theory and Graph Representation (Credit 1)** | |
| **Learning Outcomes** | **After learning this module, learners will be able to**<br>• Understand graph structures and representation techniques.<br>• Implement basic graph traversal and manipulation algorithms. |
| **Content Outline** | • Introduction to Graphs: Directed, Undirected, Weighted<br>• Adjacency Matrix, Adjacency List, Edge List<br>• Graph Traversal: BFS, DFS<br>• Graph Construction from Tabular and Text Data<br>• Introduction to NetworkX and PyTorch Geometric |
| **Module 2: Basics of Graph Neural Networks (Credit 1)** | |
| **Learning Outcomes** | **After learning this module, learners will be able to**<br>• Learn how GNNs process graph data.<br>• Understand convolution operations on graphs. |
| **Content Outline** | • Introduction to GNNs: Message Passing and Aggregation<br>• Graph Convolutional Networks (GCN)<br>• GraphSAGE: Sampling-based Approaches<br>• Neighborhood Aggregation and Pooling<br>• GNN Training and Challenges (Over-smoothing, Scalability) |
| **Module 3: Advanced GNN Architectures and Tasks (Credit1)** | |
| **Learning Outcomes** | **After learning the module, learners will be able to**<br>• Apply GNNs to real-world machine learning tasks.<br>• Compare performance of different GNN architectures. |
| **Content Outline** | • Graph Attention Networks (GAT)<br>• Relational GNNs and Heterogeneous GNNs<br>• Node Classification, Link Prediction, Graph Classification<br>• GNN Evaluation Metrics and Benchmarks<br>• Tools: DGL, PyG, StellarGraph |
| **Module 4: Applications, Tools, and Trends in GNN (Credit1)** | |
| **Learning Outcomes** | **After learning the module, learners will be able to**<br>• Explore applications of GNNs across various domains.<br>• Learn about recent trends and research directions in GNNs. |

| **Content Outline** | - Applications: Social Network Analysis, Recommendation Systems, Bioinformatics, NLP<br>- GNN Explainability and Visualization<br>- Temporal GNNs and Dynamic Graphs<br>- Federated and Distributed GNNs<br>- Current Research and Industry Use Cases |
|---|---|

**Activities towards Comprehensive Continuous Evaluation (CCE)**

**Module 1:** Build and visualize graphs using real-world datasets

**Module 2:** Implement GCN and GraphSAGE on citation networks

**Module 3:** Node classification or link prediction project using PyG or DGL

**Module 4:** Literature review or implementation of a recent GNN research paper

**Evaluation Scheme:**

**Internal Assessment:**

Depending on the activities mentioned above a project should be developed of 50 marks. Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

The Rubric will have the following Evaluation Parameters:

| **Evaluation Parameters** | **Marks** |
|---|---:|
| Functionality (Core Features) | 20 |
| Code Quality (Structure, Readability) | 10 |
| GUI Design & User Experience | 10 |
| Input Handling & Error Management | 5 |
| Report writing or Documentation | 5 |

The project Internal assessment will be done by conducting project presentation at the department level, where External Examiner appointed by the department will be evaluating the project.

Two sample projects are given below and it is expected from students to choose a good and advance project topic that aligns with their interests and abilities to work on.

Project 1: Social network link prediction using GraphSAGE

Project 2: Product recommendation system using GAT

**External Assessment:**

End Semester examination of 50 marks for 2 hours duration will be conducted by the University.

References:-

1. William L. Hamilton, Graph Representation Learning, Morgan & Claypool
2. Zhou et al., Graph Neural Networks: A Review, IEEE Transactions on Neural Networks
3. PyTorch Geometric Documentation
4. Deep Graph Library (DGL) Resources
5. Open-source tutorials from Stanford CS224W and DeepMind

| | |
|---|---|
| **Course Code** | **3.5.3** |
| **Course Title** | **Image and Video Analytics (Elective-III)** |
| **Course Credits** | **4** |
| **Course Outcomes** | **After completion of this Course, the students will be able to**<br>• Understand foundational image representation and basic operations from first principles.<br>• Build image classification pipelines using traditional and deep learning methods.<br>• Interpret and analyze motion, activities, and tracking in video data.<br>• Design simple, practical analytics applications using image and video data. |
| **Module 1: Introduction to Digital Images and Python-based Image Processing (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Understand how images are stored and represented in computers.<br>• Perform image operations using Python libraries like OpenCV and PIL.<br>• Apply basic transformations and visualizations. |
| **Module Contents** | • What is a digital image? Pixels, resolution, grayscale vs. color<br>• Introduction to image formats (JPEG, PNG, BMP)<br>• Python for image processing: OpenCV basics<br>• Loading, displaying, resizing, and cropping images<br>• Basic operations: brightness, contrast, rotation, flipping<br>• Image histograms and visualization |
| **Module 2: Foundational Image Analytics and Feature Extraction (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Extract basic features like edges, shapes, and textures.<br>• Understand the role of features in image classification.<br>• Preprocess and prepare image datasets. |
| **Module Contents** | • Concepts of edges and contours<br>• Introduction to filters: blur, sharpen<br>• Edge detection: Canny, Sobel (intuitive explanation and code)<br>• Introduction to features: color histograms, shape descriptors<br>• Simple image classifiers: KNN, Decision Tree with Scikit-learn<br>• Dataset preparation: resizing, grayscale conversion, flattening |
| **Module 3: Deep Learning for Image & Video Data (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Understand basics of CNNs and apply to image classification.<br>• Use transfer learning for real-world image tasks.<br>• Understand and process simple video data. |

| | |
|---|---|
| **Module Contents** | • Introduction to Neural Networks and CNNs (no math-heavy)<br>• Building image classifiers using Keras/TensorFlow<br>• Transfer Learning using VGG16 or MobileNet<br>• Introduction to video processing: Reading video files with OpenCV<br>• Extracting frames, saving frames<br>• Basics of object detection (YOLO/SSD overview only) |
| **Module 4: Applications of Image and Video Analytics (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Work with video streams for real-world use cases.<br>• Recognize activities or detect movement.<br>• Design a simple analytics system from end to end. |
| **Module Contents** | • Motion detection using frame differencing<br>• Background subtraction (MOG2, thresholding)<br>• Introduction to object tracking: Centroid tracking, basic Kalman filter (intuitive)<br>• Application domains: traffic monitoring, face detection, emotion recognition<br>• Capstone discussion: how to build an end-to-end pipeline<br>• Ethical aspects: surveillance, bias in AI, privacy |
| **Assignments/ Activities towards Comprehensive Continuous Evaluation** | |
| Module 1:<br>• Load different images and perform geometric operations<br>• Create a simple image collage using Python.<br>Module 2:<br>• Extract color and edge features from a dataset and classify using KNN.<br>• Compare classification accuracy with and without preprocessing.<br>Module 3:<br>• Train a CNN to classify images (e.g., dogs vs. cats).<br>• Extract frames from a video and convert them to grayscale.<br>Module 4:<br>• Detect motion in a surveillance-style video using frame differencing.<br>• Mini-project: Choose one application (e.g., face mask detection, fall detection, people counting). | |

**Evaluation Scheme:**

**Internal Assessment:**

Depending on the activities mentioned above a project should be developed of 50 marks. Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

| The Rubric will have the following Evaluation Parameters: | |
|---|---|
| **Evaluation Parameters** | **Marks** |
| Functionality (Core Features) | 20 |
| Code Quality (Structure, Readability) | 10 |
| GUI Design & User Experience | 10 |
| Input Handling & Error Management | 5 |
| Report writing or Documentation | 5 |

The project Internal assessment will be done by conducting project presentation at the department level, where External Examiner appointed by the department will be evaluating the project.

Two sample projects are given below and it is expected from students to choose a good and advance project topic that aligns with their interests and abilities to work on.

Project 1: Develop a system that captures video from a webcam and detects faces in real-time. The system should also recognize basic facial emotions like happy, sad, neutral, and surprised.

Project 2: Create a tool to detect and count vehicles from pre-recorded road surveillance videos using object detection techniques.

**External Assessment:**

End Semester examination of 50 marks for 2 hours duration will be conducted by the University.

**References:**

- Adrian Rosebrock, *Practical Python and OpenCV*, PyImageSearch
- Gonzalez & Woods, *Digital Image Processing* – only selected basics
- A. Kaehler & G. Bradski, *Learning OpenCV 4*, O'Reilly
- François Chollet, *Deep Learning with Python*, Manning

| | |
|---|---|
| **Course Code** | **3.5.4** |
| **Course Title** | **Mobile App Development Technologies**<br>**(Elective-III)** |
| **Course Credits** | **4** |
| **Course Outcomes** | **After Completion of this Course, students will be able to**<br>• Understand mobile app architecture, platforms (Android, iOS), and development environments.<br>• Develop responsive, user-friendly mobile applications using cross-platform frameworks.<br>• Integrate device features like camera, location, and sensors into mobile applications.<br>• Deploy mobile apps and understand app publishing, testing, and |

| | |
|---|---|
| | security practices. |
| **Module 1: Mobile App Fundamentals and Development Tools (Credit 1)** | |
| **Learning Outcomes** | **After learning this module, learners will be able to**<br>• Explain mobile platform differences and app types (native, hybrid, web).<br>• Set up development environments (Android Studio, Flutter, etc.)<br>• Understand app architecture, UI guidelines, and lifecycle. |
| **Content Outline** | • Mobile ecosystem: Android vs iOS, hybrid vs native vs web apps<br>• App architecture (MVC, MVVM)<br>• IDEs: Android Studio, Visual Studio Code<br>• Flutter, React Native, Kotlin, Swift overview<br>• App structure: activities, fragments, views, layouts<br>• Emulator setup and debugging basics |
| **Module 2: UI Design, Layouts & Navigation (Credit 1)** | |
| **Learning Outcomes** | **After learning this module, learners will be able to**<br>• Design and implement interactive and responsive UI.<br>• Handle user interaction with forms, buttons, and menus.<br>• Navigate between screens using routing/navigation techniques. |
| **Content Outline** | • Flutter widgets: text, images, buttons, input fields<br>• Layouts: Column, Row, Stack, GridView<br>• Navigation and routing between screens<br>• State management (Provider, setState, Bloc – overview)<br>• App theme and localization |
| **Module 3: Device Integration & Data Handling (Credit1)** | |
| **Learning Outcomes** | **After learning the module, learners will be able to**<br>• Access mobile hardware features like GPS, camera, and accelerometer.<br>• Read/write to internal and external storage.<br>• Connect mobile apps to databases and APIs. |
| **Content Outline** | • Accessing sensors: camera, GPS, accelerometer<br>• Handling permissions<br>• Local storage: SharedPreferences, SQLite, Hive<br>• REST APIs: fetching and sending data using HTTP<br>• JSON parsing<br>• Firebase overview (Auth, Realtime DB) |
| **Module 4: App Testing, Deployment & Security (Credit1)** | |

| **Learning Outcomes** | **After learning the module, learners will be able to**<br>• Understand app testing strategies and performance optimization.<br>• Prepare and publish apps to Play Store/App Store.<br>• Apply security and privacy measures in mobile apps. |
|---|---|
| **Content Outline** | • Debugging and logging<br>• Unit testing and UI testing (Flutter Test, Espresso – overview)<br>• App performance tips: lazy loading, image optimization<br>• App signing, APK generation<br>• Publishing to Google Play Store (steps, guidelines)<br>• Best practices: secure API handling, data encryption |

**Activities towards Comprehensive Continuous Evaluation (CCE)**

**Module 1:**

- Build a basic "Hello World" app in Flutter and explore its UI structure.

**Module 2:**

- Create a multi-screen user registration/login app with form validation.

**Module 3:**

- Build a location-aware app that captures photos and stores them with GPS data.

**Module 4:**

- Final mini-project: Fully functional app (e.g., To-do app, Weather app, Expense Tracker) with storage, navigation, and API integration.

**Evaluation Scheme:**

**Internal Assessment:**

Depending on the activities mentioned above a project should be developed of 50 marks. Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

The Rubric will have the following Evaluation Parameters:

| **Evaluation Parameters** | **Marks** |
|---|---|
| Functionality (Core Features) | 20 |
| Code Quality (Structure, Readability) | 10 |
| GUI Design & User Experience | 10 |
| Input Handling & Error Management | 5 |
| Report writing or Documentation | 5 |

The project Internal assessment will be done by conducting project presentation at the department level, where External Examiner appointed by the department will be evaluating the project.

Two sample projects are given below and it is expected from students to choose a good and advance project topic that aligns with their interests and abilities to work on.

Project 1: Design a mobile application to help users track their daily health and fitness activities like steps, calories burned, water intake, and sleep hours.

Project 2: Create a mobile app to manage daily expenses, categorize spending, and generate simple reports for budgeting.

**External Assessment:**

End Semester examination of 50 marks for 2 hours duration will be conducted by the University.

References:-

1. Flutter in Action – Eric Windmill
2. 
3. Android Programming: The Big Nerd Ranch Guide – Bill Phillips
4. 
5. Beginning iOS Programming with Swift – Simon Ng
6. 
7. Pro Android with Kotlin – Peter Späth

| **Course Code** | **3.6** |
|---|---|
| **Course Title** | **Research Proposal** |
| **Course Credits** | **4** |
| **Course Outcomes** | **After Completion of this Course, students will be able to**<br>• Understand the research process and define a viable research problem.<br>• Conduct a thorough literature review and identify research gaps.<br>• Design research proposals using appropriate scientific methods.<br>• Plan data collection and analysis using data science tools. |
| **Module 1: Problem Formulation and Research Process (Credit 1)** | |
| **Learning Outcomes** | **After learning this module, learners will be able to**<br>• Understand components of research and formulate research problems.<br>• Recognize types of research and associated methodologies. |
| **Content Outline** | • Definition and objectives of research<br>• Types of research: exploratory, descriptive, analytical, predictive<br>• Steps in the research process<br>• Problem identification and formulation<br>• Feasibility study of research problem |
| **Module 2: Literature Review and Hypothesis Development (Credit 1)** | |
| **Learning Outcomes** | **After learning this module, learners will be able to**<br>• Conduct systematic literature reviews.<br>• Formulate hypotheses and define research objectives. |

| | |
|---|---|
| **Content Outline** | • Importance of literature review<br>• Sources of information: journals, conferences, databases<br>• Reference management tools (Zotero, Mendeley)<br>• Hypothesis formulation and types<br>• Identifying research gaps |
| **Module 3: Research Proposal and Design (Credit1)** | |
| **Learning Outcomes** | **After learning the module, learners will be able to**<br>• Develop effective research proposals.<br>• Design research using quantitative and qualitative methods. |
| **Content Outline** | • Components of a research proposal<br>• Experimental, quasi-experimental, survey, and case study methods<br>• Ethical considerations in research<br>• Approval processes (IRB, consent forms)<br>• Proposal writing and review |
| **Module 4: Data Collection and Analysis Planning (Credit1)** | |
| **Learning Outcomes** | **After learning the module, learners will be able to**<br>• Select and apply suitable data collection methods.<br>• Plan data analysis using statistical and ML techniques. |
| **Content Outline** | • Data collection tools: surveys, interviews, APIs, sensors<br>• Sampling techniques<br>• Data validation and preprocessing<br>• Analytical tools: Python, R, Excel, SPSS<br>• Planning analysis pipelines for quantitative research |
| **Activities towards Comprehensive Continuous Evaluation (CCE)** | |
| **Module 1:** Formulating a research problem and defining objectives<br>**Module 2:** Conducting and presenting a literature review<br>**Module 3:** Writing a research proposal<br>**Module 4:** Designing a data collection plan and analysis approach | |

**Evaluation Scheme:**

**Internal Assessment (50 Marks)**

| Evaluation Parameter | Marks |
|---|---|
| Research Proposal Development | 20 |
| Literature Review Presentation | 10 |
| Tool/Technique Design | 10 |
| Documentation and Reporting | 10 |

Assessment will be based on presentations, written submissions, and faculty feedback.

**External Assessment (50 Marks)**

University-level end-semester examination (2 hours) comprising theory and case-based questions.

**Suggested Textbooks and References**

1. C.R. Kothari, Research Methodology: Methods and Techniques
2. Ranjit Kumar, Research Methodology: A Step-by-Step Guide
3. Cooper & Schindler, Business Research Methods
4. Selected journal articles and conference papers from Scopus/Web of Science databases

# Semester-IV

| | |
|---|---|
| **Course Code** | **4.1.1** |
| **Course Title** | **Applied Data Analysis (XAI)** |
| **Course Credits** | **4** |
| **Course Outcomes** | **After completion of this Course, the students will be able to**<br>• Understand the data analysis pipeline from data acquisition to actionable insights.<br>• Apply statistical and machine learning techniques to structured and unstructured data.<br>• Design and implement end-to-end data analysis solutions using Python libraries.<br>• Solve domain-specific problems using applied case studies from healthcare, finance, and IoT. |
| **Module 1: Foundations of Data Analysis (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Understand the data lifecycle and problem definition process.<br>• Perform data loading, preprocessing, and cleaning using Python (Pandas, NumPy).<br>• Conduct exploratory data analysis (EDA) and basic visualization. |
| **Module Contents** | • Introduction to applied data analysis and real-world problems<br>• Data types, sources (structured, unstructured)<br>• Python libraries: Pandas, NumPy, Matplotlib, Seaborn<br>• Handling missing values, outliers, duplicates<br>• EDA: statistical summaries, histograms, boxplots, pair plots<br>• Data wrangling and feature engineering |
| **Module 2: Statistical & Predictive Modeling Techniques (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Apply statistical techniques (correlation, hypothesis testing) in data analysis.<br>• Develop and evaluate machine learning models (classification, regression).<br>• Understand model overfitting, bias-variance tradeoff, and cross-validation. |
| **Module Contents** | • Statistical analysis: t-tests, chi-square, correlation, ANOVA<br>• Introduction to supervised learning: Linear Regression, Logistic Regression<br>• Decision Trees, Random Forests, KNN |

| | |
|---|---|
| | • Model selection, evaluation (Confusion Matrix, ROC, Precision, Recall, F1-Score)<br>• Feature importance and dimensionality reduction (PCA) |
| **Module 3: Time Series, IoT & Streaming Data Analysis (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Perform time series decomposition, trend/seasonality detection.<br>• Analyze sensor data and real-time streaming data from IoT environments.<br>• Apply forecasting techniques for actionable decision-making. |
| **Module Contents** | • Time Series concepts: trend, seasonality, cyclic behavior<br>• ARIMA, Holt-Winters, Prophet<br>• IoT data structure: streaming data pipelines and sensor integration<br>• Real-time data analysis with PySpark or Kafka (conceptual overview)<br>• Feature extraction from temporal data |
| **Module 4: Domain-Specific Case Studies and Communication of Results (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Integrate end-to-end data analysis for healthcare, finance, or IoT case study.<br>• Develop dashboards and visual storytelling using Plotly/Power BI/Tableau.<br>• Communicate actionable insights to non-technical stakeholders. |
| **Module Contents** | • Healthcare: patient risk stratification, hospital readmission prediction<br>• Finance: fraud detection, customer segmentation, investment analysis<br>• IoT: anomaly detection, predictive maintenance<br>• Creating dashboards and interactive visuals<br>• Ethics, privacy, and data governance |
| **Assignments/ Activities towards Comprehensive Continuous Evaluation** | |
| These assignments aim to apply theoretical concepts to practical application and critical thinking.<br><br>**Module 1:**<br>• EDA on a healthcare dataset (e.g., diabetes, patient records): derive insights from missing data patterns and variable distributions.<br>**Module 2:**<br>• Finance case study: Predict stock movement or credit default using classification models and interpret model performance.<br>**Module 3:**<br>• IoT case study: Analyze and forecast smart home energy consumption data or wearable device health signals.<br>**Module 4:**<br>• Choose one domain (Healthcare/Finance/IoT) and conduct a complete case study project (data ingestion → modeling → visualization → reporting). | |

**Evaluation Scheme:**

**Internal Assessment:**

Depending on the activities mentioned above a project should be developed of 50 marks. Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

| The Rubric will have the following Evaluation Parameters: | |
|---|---|
| **Evaluation Parameters** | **Marks** |
| Functionality (Core Features) | 20 |
| Code Quality (Structure, Readability) | 10 |
| GUI Design & User Experience | 10 |
| Input Handling & Error Management | 5 |
| Report writing or Documentation | 5 |

The project Internal assessment will be done by conducting project presentation at the department level, where External Examiner appointed by the department will be evaluating the project.

Two sample projects are given below and it is expected from students to choose a good and advance project topic that aligns with their interests and abilities to work on.

Project 1: Build a predictive model using a public healthcare dataset (e.g., Pima Indians Diabetes Dataset) to identify whether a patient is at risk of diabetes.

Project 2: Analyze financial transactions to detect anomalies and classify potential fraud using machine learning algorithms on an imbalanced dataset.

**External Assessment:**

End Semester examination of 50 marks for 2 hours duration will be conducted by the University.

**References:**

1. Python for Data Analysis by Wes McKinney (O'Reilly)
2. Practical Statistics for Data Scientists by Peter Bruce, Andrew Bruce
3. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aurélien Géron
4. Storytelling with Data by Cole Nussbaumer Knaflic

| | |
|---|---|
| **Course Code** | **4.1.2** |
| **Course Title** | **Social Network Analysis (Elective-IV)** |
| **Course Credits** | **4** |
| **Course Outcomes** | **After completion of this Course, the students will be able to**<br>• Understand and model social structures using graph theory.<br>• Analyze centrality, influence, and community structures in networks.<br>• Apply algorithms for link prediction, clustering, and network diffusion.<br>• Use tools like NetworkX and Gephi to analyze real-world social media and communication networks. |
| **Module 1: Introduction to Social Networks and Graph Theory (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Understand key concepts of graph theory used in SNA.<br>• Represent social networks as graphs using Python.<br>• Visualize networks and compute basic metrics. |
| **Module Contents** | • What is a social network? Real-world examples<br>• Types of networks: Undirected, directed, weighted, bipartite<br>• Basic graph terminology: nodes, edges, paths, degrees<br>• Representing graphs in Python (adjacency matrix, list)<br>• Using NetworkX for graph creation and visualization<br>• Hands-on: Create and visualize sample social networks |
| **Module 2: Centrality Measures and Influence Analysis (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Quantify node importance using various centrality metrics.<br>• Interpret roles of individuals in networks (influencers, bridges, hubs).<br>• Use real data (e.g., Twitter, email) for centrality analysis. |
| **Module Contents** | • Degree centrality, betweenness centrality, closeness centrality<br>• Eigenvector centrality and PageRank<br>• Comparing centrality metrics and network influence<br>• Use cases: influencer detection in Twitter, citation networks<br>• Network visualization and ranking based on centrality |
| **Module 3: Community Detection and Network Dynamics (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Detect communities and analyze modular structure in networks.<br>• Understand how information spreads over networks.<br>• Model and simulate simple diffusion processes. |
| **Module Contents** | • Communities and modularity in networks<br>• Community detection algorithms: Girvan-Newman, Louvain<br>• Network density, clustering coefficient<br>• Cascade models: Linear Threshold Model, Independent Cascade Model<br>• Contagion and diffusion in social networks (viral content, epidemics) |
| **Module 4: Applications and Ethical Aspects of SNA (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to: |

| | |
|---|---|
| | - Apply SNA techniques in real-world domains.<br>- Understand privacy and ethical implications of analyzing social networks.<br>- Present and interpret analytical findings with tools. |
| **Module Contents** | - Applications: political analysis, marketing, healthcare, fraud detection<br>- Visualization tools: Gephi, Python-Plotly integration<br>- Link prediction and recommendation systems<br>- Dynamic networks and temporal changes<br>- Ethical concerns: user consent, bias, anonymization |

**Assignments/ Activities towards Comprehensive Continuous Evaluation**

These assignments aim to apply theoretical concepts to practical application and critical thinking.

**Module 1:**

- Task students to build a simple neural network from scratch using Python or a framework like TensorFlow/Keras. They should train it on a dataset and analyze its performance.

**Module 2:**

- Provide pre-trained neural network models and have students visualize the learned features and activations at different layers to understand how information is processed.

**Module 3:**

- Assign students to create a CNN model for image classification using a dataset like CIFAR-10 or MNIST. They should experiment with different architectures and hyper parameters.

**Module 4:**

- Challenge students to create a GAN model capable of generating realistic images from a given dataset (e.g., faces, digits). They should evaluate the quality of generated images

**Evaluation Scheme:**

**Internal Assessment:**

Depending on the activities mentioned above a project should be developed of 50 marks. Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

The Rubric will have the following Evaluation Parameters:

| **Evaluation Parameters** | **Marks** |
|---|---|
| Functionality (Core Features) | 20 |
| Code Quality (Structure, Readability) | 10 |
| GUI Design & User Experience | 10 |
| Input Handling & Error Management | 5 |
| Report writing or Documentation | 5 |

The project Internal assessment will be done by conducting project presentation at the department level, where External Examiner appointed by the department will be evaluating the project.

Two sample projects are given below and it is expected from students to choose a good and advance project topic that aligns with their interests and abilities to work on.

Project 1: Twitter Hashtag Influence Network
Analyze a real-world Twitter dataset to understand the relationships between hashtags and users. Build a network graph showing influential hashtags and their co-occurrence with others.

Project 2: LinkedIn Connection Network (Simulated Data)
Simulate a LinkedIn-like social graph representing professional connections. Analyze how information or job opportunities can flow across such networks and identify key connectors and community clusters.

**External Assessment:**

End Semester examination of 50 marks for 2 hours duration will be conducted by the University.

**References:**

1. Wasserman & Faust, Social Network Analysis: Methods and Applications, Cambridge University Press
2. David Easley & Jon Kleinberg, Networks, Crowds, and Markets: Reasoning about a Highly Connected World, Cambridge
3. Sune Lehmann et al., Social Network Analysis: A Practical Introduction, Wiley
4. Matthew A. Russell, Mining the Social Web, O'Reilly

| | |
|---|---|
| **Course Code** | **4.1.3** |
| **Course Title** | **Explainable Artificial Intelligence (XAI)** |
| **Course Credits** | **4** |
| **Course Outcomes** | **After completion of this Course, the students will be able to**<br>• Understand the motivations, principles, and scope of Explainable AI.<br>• Analyze the limitations of black-box models and implement explainability techniques..<br>• Apply model-specific and model-agnostic interpretability techniques in real-world problems.<br>• Evaluate interpretability methods across domains such as healthcare, finance, and autonomous systems. |
| **Module 1: Introduction to Explainable AI (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Understand why interpretability in AI models is important.<br>• Identify limitations and risks of black-box AI models.<br>• Explore regulatory, ethical, and human-centric needs for explainability. |
| **Module Contents** | • Overview of AI model interpretability<br>• Black-box vs white-box models<br>• Motivation and significance of XAI<br>• Ethical and regulatory frameworks: GDPR, transparency laws<br>• Trade-offs: Accuracy vs Interpretability<br>• Introduction to post-hoc explainability |
| **Module 2: Model-Agnostic Explainability Techniques (1 Credit)** | |

| | |
|---|---|
| **Learning Outcomes** | Learners will be able to:<br>• Implement interpretable visualizations using SHAP, LIME, and Partial Dependence Plots.<br>• Explore permutation-based feature importance and surrogate models.<br>• Apply local and global explainability techniques. |
| **Module Contents** | • LIME (Local Interpretable Model-Agnostic Explanations)<br>• SHAP (SHapley Additive exPlanations): Global vs Local<br>• Permutation importance<br>• PDP (Partial Dependence Plot) and ALE (Accumulated Local Effects)<br>• Surrogate models: Decision Trees for explanation<br>• Visualizing decision boundaries |
| **Module 3: Model-Specific Explainability (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Understand how interpretability is built into transparent models<br>• mplement XAI methods for neural networks and decision trees.<br>• Apply gradient-based and attention-based interpretability. |
| **Module Contents** | • Interpretability in Decision Trees, Logistic Regression, Rule-based models<br>• Feature importance in Random Forests<br>• Neural Network interpretability: Saliency maps, Grad-CAM for CNNs, Integrated Gradients, Attention visualization in Transformers<br>• Explainability in time series and text models |
| **Module 4: Applications, Tools, and Ethics in XAI (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Evaluate explainability techniques in healthcare, finance, and autonomous systems.<br>• Use tools and libraries for XAI (e.g., What-If Tool, AI Explainability 360).<br>• Understand limitations and ethical dilemmas in interpreting AI. |
| **Module Contents** | • XAI in healthcare: Model explanations for diagnosis (e.g., X-rays, ECGs)<br>• XAI in finance: Credit scoring, fraud detection<br>• Fairness and bias in interpretability<br>• Tools: Google What-If Tool, IBM AI Explainability 360, Alibi Explain, ELI5<br>• Limitations of explanations (incompleteness, instability)<br>• Future of human-centric AI |
| **Assignments/ Activities towards Comprehensive Continuous Evaluation** | |

These assignments aim to apply theoretical concepts to practical application and critical thinking.

**Module 1:**

- Analyze a news article/case study where lack of model transparency caused harm. Present findings and suggest how XAI could help.

**Module 2:**

- Apply SHAP and LIME to a black-box classifier trained on the UCI Adult Income dataset. Visualize local and global feature contributions.

**Module 3:**

- Apply Grad-CAM or attention visualization on a sentiment classification model using LSTM or BERT.

**Module 4:**

- Build a dashboard using Streamlit/Gradio to present a model with explainable output to non-technical users.

**Evaluation Scheme:**

**Internal Assessment:**

Depending on the activities mentioned above a project should be developed of 50 marks. Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

The Rubric will have the following Evaluation Parameters:

| **Evaluation Parameters** | **Marks** |
|---|---|
| Functionality (Core Features) | 20 |
| Code Quality (Structure, Readability) | 10 |
| GUI Design & User Experience | 10 |
| Input Handling & Error Management | 5 |
| Report writing or Documentation | 5 |

The project Internal assessment will be done by conducting project presentation at the department level, where External Examiner appointed by the department will be evaluating the project.

Three sample projects are given below and it is expected from students to choose a good and advance project topic that aligns with their interests and abilities to work on.

Project 1: Explainable Credit Scoring Model using SHAP + Streamlit Interface

Project 2: Visualizing Interpretability of CNN in Medical Imaging (Pneumonia Detection from X-rays)

Project 3: Comparing Global Interpretability Techniques for Loan Default Prediction

**External Assessment:**

End Semester examination of 50 marks for 2 hours duration will be conducted by the University.

**References:**

1. Rudin, Cynthia – Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature Machine Intelligence, 2019
2. Samek, Wojciech et al. – Explainable AI: Interpreting, Explaining and Visualizing Deep Learning.

| | |
|---|---|
| **Course Code** | **4.1.4** |
| **Course Title** | **Internet of Things (IOT) (Elective-IV)** |
| **Course Credits** | **4** |
| **Course Outcomes** | **After Completion of this Course, students will be able to**<br>• Understand the architecture, protocols, and applications of IoT.<br>• Design IoT systems using sensors, microcontrollers, and network interfaces.<br>• Develop real-time data acquisition and transmission systems.<br>• Implement IoT-based solutions using platforms such as Arduino and Raspberry Pi.<br>• Explore IoT applications in smart cities, healthcare, agriculture, and industry. |
| **Module 1: Introduction to IoT (Credit 1)** | |
| **Learning Outcomes** | **After learning this module, learners will be able to**<br>• Understand the IoT ecosystem and its enabling technologies.<br>• Identify use-cases across various domains. |
| **Content Outline** | • Definition, Evolution and Characteristics of IoT<br>• IoT Ecosystem and Architecture<br>• Sensors, Actuators, Embedded Systems<br>• IoT Applications: Smart Cities, Homes, Agriculture, Healthcare, Industry 4.0 |
| **Module 2: IoT Hardware and Networking (Credit 1)** | |
| **Learning Outcomes** | **After learning this module, learners will be able to**<br>• Work with microcontrollers and networking protocols used in IoT. |
| **Content Outline** | • Arduino and Raspberry Pi Boards<br>• GPIO Interfacing and Peripheral Sensors<br>• Wireless Technologies: Bluetooth, ZigBee, Wi-Fi, LoRa, RFID<br>• Communication Protocols: MQTT, CoAP, HTTP, TCP/IP |
| **Module 3: Data Acquisition, Processing, and Cloud Integration (Credit1)** | |
| **Learning Outcomes** | **After learning the module, learners will be able to**<br>• Design real-time systems with data collection, processing, and cloud storage. |
| **Content Outline** | • Real-Time Data Acquisition and Sensor Interfacing<br>• Data Processing and Visualization<br>• Edge and Fog Computing<br>• Cloud Platforms: ThingSpeak, AWS IoT, Google Cloud IoT<br>• IoT Dashboards and Alerts |

| **Module 4: Security, Standards, and Future Trends (Credit1)** | |
|---|---|
| **Learning Outcomes** | **After learning the module, learners will be able to**<br>- Explore challenges and advancements in securing and scaling IoT systems. |
| **Content Outline** | - IoT Security Challenges and Best Practices<br>- Encryption, Authentication, Secure Boot<br>- IoT Standards and Interoperability<br>- Future Trends: AIoT, 5G, Digital Twins<br>- Sustainable IoT and Green Computing |

| **Activities towards Comprehensive Continuous Evaluation (CCE)** |
|---|
| **Module 1:** Presentation on IoT applications across domains<br>**Module 2:** Sensor interfacing with Arduino or Raspberry Pi<br>**Module 3:** Build a cloud-connected weather monitoring system<br>**Module 4:** Research report on security vulnerabilities in IoT |

**Evaluation Scheme:**

**Internal Assessment:**

Depending on the activities mentioned above a project should be developed of 50 marks. Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

| The Rubric will have the following Evaluation Parameters: | |
|---|---|
| **Evaluation Parameters** | **Marks** |
| Functionality (Core Features) | 20 |
| Code Quality (Structure, Readability) | 10 |
| GUI Design & User Experience | 10 |
| Input Handling & Error Management | 5 |
| Report writing or Documentation | 5 |

The project Internal assessment will be done by conducting project presentation at the department level, where External Examiner appointed by the department will be evaluating the project.

Two sample projects are given below and it is expected from students to choose a good and advance project topic that aligns with their interests and abilities to work on.

Project 1: Real-Time Air Quality Monitoring and Alert System

Project 2: GPS-based Vehicle Tracking using GSM Module

**External Assessment:**

End Semester examination of 50 marks for 2 hours duration will be conducted by the University.

**References:-**

1. Arshdeep Bahga, Vijay Madisetti, Internet of Things: A Hands-On Approach
2. Adrian McEwen, Hakim Cassimally, Designing the Internet of Things
3. Raj Kamal, Internet of Things: Architecture and Design
4. Donald Norris, The Internet of Things: Do-It-Yourself Projects
5. Official Arduino, Raspberry Pi, and AWS IoT Documentation

| **Course Code** | **4.2** |
|---|---|
| **Course Title** | **Interactive and Advanced Data Visualization (with Python)** |
| **Course Credits** | **4** |
| **Course Outcomes** | **After completion of this Course, the students will be able to**<br>• Understand core principles of visual perception and graphical integrity.<br>• Create static, dynamic, and interactive data visualizations using Python libraries.<br>• Explore advanced visualizations for multidimensional and time-series data.<br>• Build interactive dashboards and communicate insights effectively |
| **Module 1: Foundations of Data Visualization (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Understand principles of good visual design.<br>• Create basic plots using Matplotlib and Seaborn.<br>• Customize visualizations for effective communication. |
| **Module Contents** | • What makes a good visualization? Edward Tufte's principles<br>• Types of data: categorical, numerical, temporal, spatial<br>• Introduction to Matplotlib: line, bar, scatter, pie<br>• Seaborn: pairplot, heatmap, boxplot, violin plot<br>• Plot customization: labels, legends, titles, styles, subplots<br>• Data wrangling with Pandas for visualization |
| **Module 2: Interactive Visualizations with Plotly and Altair (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Create interactive and animated visualizations.<br>• Use Plotly and Altair to build user-responsive plots.<br>• Apply filters, sliders, and tooltips for dynamic data exploration. |
| **Module Contents** | • Plotly Express and Graph Objects: scatter, bar, 3D, maps<br>• Interactive elements: hover, zoom, sliders, dropdowns |

| | |
|---|---|
| | • Introduction to Altair: grammar of graphics approach<br>• Building linked charts and interactive selections<br>• Exporting interactive plots to HTML<br>• Comparison of interactivity features in Plotly and Altair |
| **Module 3: Advanced Visualization Techniques (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Visualize complex data structures like time series, networks, and geospatial data.<br>• Apply animation and storytelling techniques to data.<br>• Integrate data visualization with statistics and machine learning. |
| **Module Contents** | • Time-series visualization with Pandas, Plotly<br>• Network graphs with NetworkX and Plotly<br>• Geospatial data visualization with Folium and GeoPandas<br>• Dimensionality reduction plots (PCA, t-SNE, UMAP)<br>• Animated visualizations with Plotly and Matplotlib<br>• Statistical plots: confidence intervals, regression lines |
| **Module 4: Dashboards, Storytelling & Real-World Applications (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Build and deploy dashboards using Streamlit or Dash.<br>• Design storytelling dashboards with effective narrative and layout.<br>• Address ethical concerns in data communication. |
| **Module Contents** | • Streamlit basics: layout, widgets, charts, state<br>• Dash basics: app structure, callbacks, layout, components<br>• Designing dashboards for business and research use cases<br>• Case studies: finance, healthcare, social media<br>• Storytelling techniques: narrative flow, annotation, color and design<br>• Ethics in visualization: misleading charts, bias, accessibility |
| **Assignments/ Activities towards Comprehensive Continuous Evaluation** | |

These assignments aim to apply theoretical concepts to practical application and critical thinking.

**Module 1:**

- Load a dataset and visualize basic distribution and correlation using Seaborn.
- Create subplots comparing multiple variables using Matplotlib.

**Module 2:**

- Use Plotly to create an interactive time-series chart with slider and tooltip.

- Design a linked Altair chart to show filtered views of multivariate data.

**Module 3:**

- Create a folium map showing heatmap of crime or COVID data.
- Visualize clusters using t-SNE or PCA for a multi-class dataset.

**Module 4:**

- Build an interactive Streamlit dashboard for a dataset of your choice.
- Create a story-driven dashboard with visuals explaining a real-world problem.

**Evaluation Scheme:**

**Internal Assessment:**

Depending on the activities mentioned above a project should be developed of 50 marks. Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

The Rubric will have the following Evaluation Parameters:

| Evaluation Parameters | Marks |
|---|---|
| Functionality (Core Features) | 20 |
| Code Quality (Structure, Readability) | 10 |
| GUI Design & User Experience | 10 |
| Input Handling & Error Management | 5 |
| Report writing or Documentation | 5 |

The project Internal assessment will be done by conducting project presentation at the department level, where External Examiner appointed by the department will be evaluating the project.

Two sample projects are given below and it is expected from students to choose a good and advance project topic that aligns with their interests and abilities to work on.

Project 1: Develop a web-based dashboard that visually analyzes COVID-19 trends globally. Users can interactively explore cases, deaths, and vaccination status by continent, country, and over time.
Project 2: Create an interactive dashboard to analyze historical stock data (e.g., Apple, Tesla, Reliance). Users can visualize trends, compare companies, and interact with various metrics such as moving averages and RSI.

**External Assessment:**

End Semester examination of 50 marks for 2 hours duration will be conducted by the University.

**References:**

1. Alberto Cairo, The Truthful Art: Data, Charts, and Maps for Communication, New Riders

2. Ben Fry, Visualizing Data, O'Reilly
3. Nathan Yau, Data Points: Visualization That Means Something, Wiley
4. Jake VanderPlas, Python Data Science Handbook, O'Reilly

| | |
|---|---|
| **Course Code** | **4.3** |
| **Course Title** | **Advanced Natural Language Processing** |
| **Course Credits** | **4** |
| **Course Outcomes** | **After completion of this Course, the students will be able to**<br>• Explain and compare modern representation learning techniques (word embeddings, contextual embeddings, subword tokenization).<br>• Implement and fine-tune deep neural architectures (RNN, LSTM, Transformer, encoder–decoder) for core NLP tasks.<br>• Apply state-of-the-art pretrained language models (BERT, GPT-style, T5, LLaMA family) to downstream NLU and NLG problems using transfer learning.<br>• Evaluate and error-analyze NLP systems across tasks (classification, QA, generation, sequence tagging) using appropriate metrics and benchmarks.<br>• Address multilingual, domain adaptation, fairness, bias, and safety challenges in deploying NLP applications in real settings. |
| **Module 1: Representation Learning for Language (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Contrast symbolic vs. distributed representations used in NLP.<br>• Train and interpret dense word embeddings (Word2Vec, GloVe, FastText).<br>• Understand subword tokenization (BPE, WordPiece, SentencePiece) and why it matters for large vocabularies & multilingual text.<br>• Use contextual embeddings from pretrained models (ELMo, BERT embeddings) in downstream tasks. |
| **Module Contents** | • Review: From bag-of-words & sparse features (from prior course) to dense embeddings.<br>• Distributional hypothesis; co-occurrence statistics; embedding spaces.<br>• Word2Vec (CBOW, Skip-gram), negative sampling.<br>• GloVe: global matrix factorization intuition.<br>• FastText: subword character n-grams for morphology & OOV words.<br>• Subword tokenization: BPE, WordPiece, SentencePiece—handling rare words, multilingual corpora.<br>• Contextual vs. static embeddings: ELMo intuition; extracting BERT embeddings.<br>• Practical: using gensim, fasttext, Hugging Face tokenizers. |
| **Module 2: Deep Sequence Modeling & Attention (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to: |

| | |
|---|---|
| | • Explain sequence models: RNN, LSTM, GRU and their limitations.<br>• Understand attention mechanisms and why they replaced recurrence in state-of-the-art systems.<br>• Describe and illustrate the Transformer architecture (encoder, decoder, self-attention, multi-head, positional encoding).<br>•  Implement a small seq2seq model with attention (e.g., translation toy corpus) and inspect alignments. |
| **Module Contents** | • From n-grams to RNNs: vanishing gradients & long-range dependencies.<br>• LSTM/GRU architectures; teacher forcing; seq2seq encoder–decoder.<br>• Attention basics (Bahdanau, Luong); alignment visualization.<br>• Transformer deep dive: self-attention math intuition; scaled dot-product attention; multi-head attention; feed-forward blocks; layer norm; residuals; positional encodings.<br>• Efficiency tricks: masking, padding, batching variable length sequences.<br>• Mini translation / summarization example using seq2seq w/ attention in PyTorch or TensorFlow. |
| **Module 3: Pretrained LMs, Adaptation & Generation (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Load, fine-tune, and prompt pretrained LMs (BERT, RoBERTa, DistilBERT, GPT-2, T5).<br>• Distinguish task heads: classification, token classification, QA, seq2seq generation.<br>• Apply parameter-efficient fine-tuning (PEFT: LoRA, adapters, prefix tuning) for resource-constrained settings.<br>• Use prompting, in-context learning, and instruction tuning principles for LLM interaction.<br>• Build small NLU/NLG pipelines (sentiment analysis, QA, summarization, dialogue snippet). |
| **Module Contents** | • Pretraining objectives: MLM (BERT), CLM (GPT), encoder-decoder denoising (T5).<br>• Masked token prediction vs. next token prediction—downstream implications.<br>• Fine-tuning workflows with Hugging Face transformers Trainer API.<br>• Parameter-efficient tuning: adapters, LoRA, quantization-aware low-resource adaptation.<br>• Prompt engineering basics: zero-shot, few-shot, chain-of-thought hints.<br>• Retrieval-Augmented Generation (RAG) overview: embedding stores, vector search, grounded generation.<br>• Practical tasks: sentiment classification, NER w/ token classification, abstractive summarization. |

| **Module 4: Evaluation, Responsible & Applied NLP Systems (1 Credit)** | |
|---|---|
| **Learning Outcomes** | Learners will be able to:<br>• Select and compute appropriate task metrics: accuracy, F1, BLEU, ROUGE, METEOR, BERTScore, perplexity.<br>• Perform error analysis: confusion matrices, calibration, qualitative failure modes.<br>• Address bias, fairness, toxicity, hallucination, and privacy in NLP/LLM systems.<br>• Architect and benchmark an end-to-end applied NLP system (domain choice: education, healthcare, governance, financial tech, multilingual India-centric use cases).<br>• Prepare a professional report & demo for external evaluation.. |
| **Module Contents** | • Evaluation frameworks by task type: classification, tagging, generation, QA, dialogue.<br>• Human vs. automated evaluation; when metrics mislead.<br>• Robustness: domain shift, adversarial inputs, data leakage.<br>• Fairness & bias audits: demographic parity checks; stereotype probes; toxicity filtering.<br>• Safety & alignment considerations in LLM apps.<br>• Productionization: dataset versioning, model cards, data statements, continuous evaluation.<br>• Case studies: multilingual chatbot; assistive tech for low-resource languages; policy doc QA; clinical note de-identification (privacy). |

**Assignments/ Activities towards Comprehensive Continuous Evaluation**

These assignments aim to apply theoretical concepts to practical application and critical thinking.

**Module 1:**

- Train (or load) Word2Vec embeddings on a medium corpus; evaluate with word similarity/analogy probes.
- Compare static vs. contextual embeddings for a small semantic similarity or clustering task.
- Analyze tokenization differences across languages/scripts (e.g., English vs. Hindi).

**Module 2:**

- Visualize attention weights for a handful of sentences.
- Compare inference quality between LSTM-attention and Transformer on a tiny dataset.

**Module 3:**

- Fine-tune DistilBERT for sentiment analysis on a labeled domain dataset (e.g., product reviews).
- Prompt GPT-style model to generate domain-specific responses; compare few-shot vs. fine-

tuned.

**Module 4:**

- Evaluate two models (classical vs. Transformer) on the same classification task; compute macro vs. weighted F1 and discuss class imbalance.
- Conduct a mini bias probe (gendered pronouns in occupation predictions).
- Capstone proposal + prototype: applied ANLP system with evaluation + ethical reflection.

**Evaluation Scheme:**

**Internal Assessment:**

Depending on the activities mentioned above a project should be developed of 50 marks. Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

| The Rubric will have the following Evaluation Parameters: | |
|---|---|
| **Evaluation Parameters** | **Marks** |
| Functionality (Core Features) | 20 |
| Code Quality (Structure, Readability) | 10 |
| GUI Design & User Experience | 10 |
| Input Handling & Error Management | 5 |
| Report writing or Documentation | 5 |

The project Internal assessment will be done by conducting project presentation at the department level, where External Examiner appointed by the department will be evaluating the project.

Two sample projects are given below and it is expected from students to choose a good and advance project topic that aligns with their interests and abilities to work on.

Project 1: Design and build a smart chatbot that can answer medical-related queries using contextual understanding powered by **BERT or DistilBERT**. It can assist users in asking questions related to symptoms, diseases, or first-aid tips..

Project 2: Create a fake news detection system using advanced NLP pipelines, applying **TF-IDF + LSTM** or **BERT embeddings** to classify whether financial news headlines are real or fake.

**External Assessment:**

End Semester examination of 50 marks for 2 hours duration will be conducted by the University.

**References:**

1. Jurafsky, D., & Martin, J. H. *Speech and Language Processing* (3rd ed. draft, online).
2. Goldberg, Y. *Neural Network Methods in Natural Language Processing.* Morgan &

Claypool.

3. Eisenstein, J. *Introduction to Natural Language Processing.* MIT Press.
4. Bird, S., Klein, E., & Loper, E. *Natural Language Processing with Python* (NLTK Book) – for bridging foundational material.

| | |
|---|---|
| **Course Code** | **4.4** |
| **Course Title** | **Deep Learning** |
| **Course Credits** | **4** |
| **Course Outcomes** | **After completion of this Course, the students will be able to**<br>• Implement perceptron with understanding of input features, weights, bias, and activation function.<br>• Explore deep learning concepts like activation functions and forward propagation.<br>• Implement convolution and pooling layers in TensorFlow, understanding convolution operations.<br>• Train different RNN architectures including one-to-one, one-to-many, many-to-one, and many-to-many. |
| **Module 1: Introduction to Deep Learning (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Define input features, weights, bias, and activation function for building a perceptron.<br>• Recognize limitations of single-layer perceptron, particularly in learning non-linear relationships.<br>• Describe artificial neural network structure, comprising input, hidden, and output layers.<br>• Outline activation functions, essential for introducing non linearity in neural networks to learn complex patterns. |
| **Module Contents** | • Perceptron: What is a Perceptron? Implementing perceptron, Introducing & Implementing Weights & Bias, Multilayer Perceptron, Limitations of perceptron.<br>• Introduction to Deep Learning: What is deep learning? Biological and artificial neurons, ANN and its layers, Input layer, Hidden layer, Output layer, exploring activation functions, the sigmoid function, the tanh function, The Rectified Linear Unit function, The leaky ReLU function, The Swish function, The softmax function, Forward propagation in ANN, How does ANN learn? |
| **Module 2: Convolutional Neural Networks (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Examine TensorFlow's representation of computations through directed acyclic graphs (DAGs).<br>• Understand sessions in TensorFlow for executing operations within a computational graph.<br>• Analyze the general architecture of Convolutional Neural Networks (CNNs), involving convolutional, pooling, and fully connected layers.<br>• Implement Convolutional and Pooling layers within the CNN architecture. |

| | |
|---|---|
| **Module Contents** | • Getting to Know TensorFlow<br>• What is TensorFlow? Understanding computational graphs and sessions, Sessions, Variables, constants, and placeholders, Introducing Tensor Board, Creating a name scope.<br>• Back propagation Algorithm, Neural Network Training,<br>• Convolutional Neural Networks: Overall Architecture, The Convolution Layer, Issues with the Fully Connected Layer, Convolution Operations, Padding, Stride, Batch Processing, The Pooling Layer, Implementing a Convolution Layer, Implementing a Pooling Layer, Implementing a CNN, Visualizing a CNN. |
| **Module 3: Optimizers in DL (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Comprehend gradient descent as an optimization method for minimizing loss during training.<br>• Introduce adaptive learning rates based on historical parameter gradients.<br>• Understand the challenges of training Recurrent Neural Networks (RNNs) and managing sequential dependencies.<br>• Explain backpropagation through time, the algorithm used for training RNNs by unfolding them into computational graphs over time, and explore various RNN architectures. |
| **Module Contents** | • Optimizers in DL: Gradient Descent, Stochastic Gradient Descent, Mini-Batch Gradient Descent, SGD with Momentum, AdaGrad (Adaptive Gradient Descent), RMS-Prop (Root Mean Square Propagation), AdaDelta, Adam (Adaptive Moment Estimation).<br>• Introducing RNNs: RNN implementation and training, Backpropagation through time, Vanishing & exploding gradients, long short-term memory LSTM,<br>• Different types of RNN architectures:<br>• One-to-one architecture<br>• One-to-many architecture<br>• Many-to-one architecture<br>• Many-to-many architecture. |
| **Module 4: Deep Unsupervised Learning (1 Credit)** | |
| **Learning Outcomes** | Learners will be able to:<br>• Grasp autoencoders as neural network architectures for unsupervised learning by encoding and decoding input data.<br>• Explore Generative Adversarial Networks (GANs) as a framework for training generative models via adversarial training.<br>• Understand various scenarios for different model implementations. |
| **Module Contents** | • Deep Unsupervised Learning: Auto encoders (standard, sparse, denoising, contractive, etc), Variational Auto encoders.<br>• Deep Generative Models GANS. |
| **Assignments/ Activities towards Comprehensive Continuous Evaluation** | |
| These assignments aim to apply theoretical concepts to practical application and critical thinking.<br>**Module 1:**<br>• Task students to build a simple neural network from scratch using Python or a framework | |

like TensorFlow/Keras. They should train it on a dataset and analyze its performance.

**Module 2:**

- Provide pre-trained neural network models and have students visualize the learned features and activations at different layers to understand how information is processed.

**Module 3:**

- Assign students to create a CNN model for image classification using a dataset like CIFAR-10 or MNIST. They should experiment with different architectures and hyper parameters.

**Module 4:**

- Challenge students to create a GAN model capable of generating realistic images from a given dataset (e.g., faces, digits). They should evaluate the quality of generated images

**Evaluation Scheme:**

**Internal Assessment:**

Depending on the activities mentioned above a project should be developed of 50 marks. Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

| The Rubric will have the following Evaluation Parameters: | |
|---|---|
| **Evaluation Parameters** | **Marks** |
| Functionality (Core Features) | 20 |
| Code Quality (Structure, Readability) | 10 |
| GUI Design & User Experience | 10 |
| Input Handling & Error Management | 5 |
| Report writing or Documentation | 5 |

The project Internal assessment will be done by conducting project presentation at the department level, where External Examiner appointed by the department will be evaluating the project.

Two sample projects are given below and it is expected from students to choose a good and advance project topic that aligns with their interests and abilities to work on.

Project 1: Pneumonia Detection from Chest X-Ray Images
Use Convolutional Neural Networks (CNNs) to classify chest X-ray images as Pneumonia or Normal.

Project 2: Image Caption Generator using CNN + LSTM
Create a deep learning model that can automatically generate textual captions for a given image.

**External Assessment:**

End Semester examination of 50 marks for 2 hours duration will be conducted by the University.

**References:**

1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning (Adaptive Computation and Machine Learning series). The MIT Press.

2. Chollet, F. (2018). Deep Learning with Python. Manning.
3. Buduma, N., & Locascio, N. (2017). Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms. O'Reilly Media.

| | |
|---|---|
| **Course Code** | **4.4** |
| **Course Title** | **Project Dissertation** |
| **Course Credits** | **6** |
| **Course Outcomes** | **After Completion of this Course, students will be able to**<br>• Apply machine learning and deep learning techniques to real-world problems using structured and unstructured datasets.<br>• Evaluate model performance using appropriate metrics and statistical tests.<br>• Integrate research methodology with data modeling, experimentation, and visualization.<br>• Prepare a comprehensive dissertation and a research article suitable for publication.<br>• Defend their research findings effectively in an academic viva voce. |
| **Module 1: Data Collection and Modeling (Credit 1)** | |
| **Learning Outcomes** | **After learning this module, learners will be able to**<br>• Acquire and preprocess structured/unstructured data for ML/DL models.<br>• Implement data pipelines and develop baseline models. |
| **Content Outline** | • Dataset acquisition (Kaggle, APIs, web scraping, sensors)<br>• Data cleaning, transformation, and feature engineering<br>• ML models: regression, classification, clustering<br>• Deep learning models: CNN, RNN, Transformers<br>• Use of tools: Scikit-learn, TensorFlow, PyTorch |
| **Module 2: Experimental Design and Methodology (Credit 1)** | |
| **Learning Outcomes** | **After learning this module, learners will be able to**<br>• Define research design, hypothesis, and metrics.<br>• Document methodology and justify model selection. |
| **Content Outline** | • Research objectives and hypothesis formulation<br>• Training, validation, and testing protocols<br>• Cross-validation and hyperparameter tuning<br>• Model selection rationale<br>• Ethical considerations in ML research |
| **Module 3: Evaluation, Results and Discussion (Credit1)** | |
| **Learning Outcomes** | **After learning the module, learners will be able to**<br>• Evaluate model performance using statistical and visual tools.<br>• Discuss results, interpret outputs, and identify limitations. |

| | |
|---|---|
| **Content Outline** | • Evaluation metrics: accuracy, precision, recall, F1-score, AUC, MAE, RMSE<br>• Explainable AI techniques (e.g., SHAP, LIME)<br>• Visualization of results using plots and dashboards<br>• Comparison with baseline models or state-of-the-art<br>• Discussion and reflection on findings |
| **Module 4: Dissertation Documentation, Article Submission and Viva (Credit1)** | |
| **Learning Outcomes** | **After learning the module, learners will be able to**<br>• Prepare and submit a well-structured dissertation and research article.<br>• Defend research work during viva voce. |
| **Content Outline** | • Final dissertation report with all chapters<br>• Drafting a publishable research article (IEEE/Elsevier/Scopus format)<br>• Formatting, citations, plagiarism check<br>• Viva preparation: presenting and defending work<br>• Responding to examiner queries and critiques |
| **Activities towards Comprehensive Continuous Evaluation (CCE)** | |
| **Module 1:** Dataset finalization and preprocessing logs<br>**Module 2:** Weekly updates on model training and tuning.<br>Draft submissions of chapters (Methodology, Results)<br>**Module 3:** Mini review presentation of intermediate results<br>**Module 4:** Submission of research article draft | |

**Evaluation Scheme:**

**Internal Assessment (50 Marks)**

| Evaluation Parameter | Marks |
|---|---|
| Research Proposal Development | 20 |
| Literature Review Presentation | 10 |
| Tool/Technique Design | 10 |
| Documentation and Reporting | 10 |

Assessment will be based on presentations, written submissions, and faculty feedback.

**External Assessment (100 Marks)**

University-level end-semester viva-voice examination will be conducted.

**Suggested Tools and References**

1. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow
2. Ian Goodfellow, Deep Learning
3. TensorFlow and PyTorch Documentation
4. IEEE Xplore, Springer, Elsevier, and Scopus Journals
5. SHAP, LIME, Captum for XAI methods