Course Scheme and Detailed Syllabus of Minor in CSE for 2024 Batch Onwards

For a minor degree in CSE, students from other departments are required to complete at least 18 credits by opting the courses as per the following.

Semester	Professional Core	Professional Elective
Semester 4	3 Credits	3 Credits
Semester 5	4 Credits	3 Credits
Semester 6		3 Credits
Semester 7	-----	3 Credits

FOURTH SEMESTER

S. No.	Course			Contact Hours				Credits
	Category	Course Code	Course Title	L	T	P	Total	
1	PCC	25B11CI414	Software Engineering	3	0	0	3	3
2	PEC	XXXXXXX	Discipline Elective – 1	2	0	0	2	2
3	PEC	XXXXXXX	Discipline Elective – 1 Lab	0	0	2	2	1

FIFTH SEMESTER

S. No.	Course			Contact Hours				Credits
	Category	Course Code	Course Title	L	T	P	Total	
1	PCC	XXXXXXX	Operating Systems	3	0	0	3	3
2	PCC	XXXXXXX	Operating Systems Lab	0	0	2	2	1
PROFESSIONAL ELECTIVE								
3	PEC	XXXXXXX	Discipline Elective – 2	2	0	0	2	2
4	PEC	XXXXXXX	Discipline Elective – 2 Lab	0	0	2	2	1
OR								
5	PEC	XXXXXXX	Discipline Elective – 3	2	0	0	2	2
6	PEC	XXXXXXX	Discipline Elective – 3 Lab	0	0	2	2	1

SIXTH SEMESTER

S. No.	Course			Contact Hours				Credits
	Category	Course Code	Course Title	L	T	P	Total	
1	PEC	XXXXXXX	Discipline Elective – 4	3	0	0	3	3
OR								
2	PEC	XXXXXXX	Discipline Elective – 5	3	0	0	3	3

SEVENTH SEMESTER

S. No.	Course			Contact Hours				Credits
	Category	Course Code	Course Title	L	T	P	Total	
1	PEC	XXXXXXX	Discipline Elective – 6	3	0	0	3	3

Bucket Wise Tentative List of Discipline Electives (To be updated time to time)

Discipline Elective – 1 and Discipline Elective – 1 Lab (offered in 4th Semester)

S. No.	Course Code	Course Title	L	T	P	Total	Credits
1		Data Analytics using R and Python	2	0	0	2	2
2		Data Analytics using R and Python Lab	0	0	2	2	1
3		Mobile Application Development	2	0	0	2	2
4		Mobile Application Development Lab	0	0	2	2	1
5		Fundamentals of Smart Systems and IoT	2	0	0	2	2
6		Fundamentals of Smart Systems and IoT Lab	0	0	2	2	1

Discipline Elective – 2 and Discipline Elective – 2 Lab (offered in 5th Semester)

S. No.	Course Code	Course Title	L	T	P	Total	Credits
1		Fundamentals of Soft Computing	2	0	0	2	2
2		Fundamentals of Soft Computing Lab	0	0	2	2	1
3		Fundamentals of Computer and Cyber Security	2	0	0	2	2
4		Fundamentals of Computer and Cyber Security Lab	0	0	2	2	1
5		Data Mining and Data Warehousing	2	0	0	2	2
6		Data Mining and Data Warehousing Lab	0	0	2	2	1
7		Agile Software Development Process	2	0	0	2	2
8		Agile Software Development Process Lab	0	0	2	2	1
9		IoT Analytics	2	0	0	2	2
10		IoT Analytics Lab	0	0	2	2	1

Discipline Elective – 3 and Discipline Elective – 3 Lab (offered in 5th Semester)

S. No.	Course Code	Course Title	L	T	P	Total	Credits
1		Image Processing and Computer Vision	2	0	0	2	2
2		Image Processing and Computer Vision Lab	0	0	2	2	1
3		Introduction to Blockchain Technology	2	0	0	2	2
4		Introduction to Blockchain Technology Lab	0	0	2	2	1
5		Computing for Data Science	2	0	0	2	2
6		Computing for Data Science Lab	0	0	2	2	1
7		Sensor Technology & Android Programming	2	0	0	2	2

8		Sensor Technology & Android Programming Lab	0	0	2	2	1

Discipline Elective – 4 (offered in 6th Semester)

S. No.	Course Code	Course Title	L	T	P	Total	Credits
1		Introduction to Deep Learning	3	0	0	3	3
2		Cryptocurrency Technologies	3	0	0	3	3
3		Information Retrieval and Semantic Web	3	0	0	3	3
4		Cloud Computing Essentials: Azure and AWS	3	0	0	3	3

Discipline Elective – 5 (offered in 6th Semester)

S. No.	Course Code	Course Title	L	T	P	Total	Credits
1		Machine Learning and Big Data	3	0	0	3	3
2		Secure Design of Software Systems	3	0	0	3	3
3		Fog and Edge Computing	3	0	0	3	3

Discipline Elective – 6 (offered in 7th Semester)

S. No.	Course Code	Course Title	L	T	P	Total	Credits
1		Machine Learning and Natural Language Processing	3	0	0	3	3
2		Ethical Hacking & Prevention	3	0	0	3	3
3		Introduction to Large Scale Database Systems	3	0	0	3	3
4		Introduction to DevOps	3	0	0	3	3
5		Industrial Automation and IOT	3	0	0	3	3

Software Engineering

COURSE CODE: 25B11CI414

COURSE CREDITS: 3

CORE/ELECTIVE: Core

L-T-P: 3-0-0

Pre-requisite: None

Course Objectives:

1. Provide students with a systematic understanding of software engineering principles and software process models used in the development of large-scale software systems.
2. Enable students to identify, analyze, and document functional and non-functional requirements and prepare a well-structured Software Requirement Specification (SRS).
3. Develop the ability to apply software design principles such as modularity, cohesion, and coupling, and to model software systems using UML diagrams.
4. Equip students with knowledge of software construction practices, coding standards, metrics, and software testing techniques to ensure software quality.
5. Encourage students to evaluate and improve software systems based on quality attributes, performance metrics, and testing outcomes, preparing them for professional software development roles.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understand software engineering principles and various process models for project development.	Remembering
CO-2	Outline the functional and non-functional requirements of a software project for creating Software Requirement Specifications (SRS).	Understand
CO-3	Apply software design principles (modularity, cohesion, coupling) and create UML models.	Apply
CO-4	Analyze the performance of software system using different testing methodologies.	Analyze
CO-5	Evaluate the software in terms of general software quality attributes, code structure and try to optimize the overall software system.	Evaluate

Course Contents:

Unit	Contents	Lectures required
1	***Introduction to Software Engineering:*** Introduction to Software Engineering Principles, Software Process Models (Build and Fix Model, Waterfall Model, Incremental Process Model, Evolutionary- Prototype and Spiral Models, Agile Models (tools	7

	study). Project Planning, Project Scheduling: Network Diagram, Gant Chart, CPM and PERT.	
2	***Requirement Engineering:*** Types of Requirement, Requirement Elicitation, Analysis, Specification, SRS, Requirement Verification and Validation.	**4**
3	***Software Design:*** Use Case Diagram, State Diagram, Activity Diagram, Class Diagram, Sequence Diagram, Collaboration Diagram, Deployment Diagram, Component Diagram and Package Diagram, Design Modularity: Coupling Cohesion.	**7**
4	***Software Construction:*** Coding Standards and Guidelines, Code Checklist, Code Reviews, Code Refactoring, Code Optimization. Design Pattern, Modern Programming Environments (Code Search, Programming using Library Components and their APIs), Program Comprehension; Program Correctness, Defensive Programming.	**8**
5	***Software Metrics:*** Size-Oriented Metric, Function-Oriented Metric, Halstead's Software Metric, Information Flow Metric, Object-Oriented Metric, Class-Oriented Metric, COCOMO Model.	**7**
6	***Software Testing:*** White-Box Testing, Basis Path Testing, Control Structure Testing: Condition Testing, Data Flow Testing, Loop Testing, Black-Box Testing: Equivalence Class Partitioning, Boundary Value Analysis, Decision Table Testing, Cause Effect Graphing, Mutation Testing and Regression Testing, Formal Methods.	**9**
Total lectures		**42**

Project-Based Learning: Each student works on a different case study in Assignments. They utilize the concepts taught in lecture and develop project in a group of 3-4.

The course emphasized on the skill development for employability in software industry by engaging students on Software Development methodologies. Various activities are carried out to enhance the student's software development skills. Some of them are study of various software process models and their applicability, progress tracking, size estimation techniques, software testing strategies, etc.

Suggested Text Book(s):

1. Roger S. Pressman, "Software Engineering: A Practitioner's Approach", Fifth Edition-TMH International.
2. Sommerville, "Software Engineering", Seventh Edition - Addison Wesley.

Suggested Reference Book(s):

1. Grady Booch, James Rumbaugh, Ivar Jacobson, The Unified Modeling Language User Guide, Addison Wesley, Reading, Massachusetts, May 2005
2. Richard Thayer, "Software Engineering Project Management", Second Edition -Wiley-IEEE Computer Society Press.

3. B. Bezier, “Software Testing Techniques”, Second Edition- International Thomson Computer Press.
4. Pankaj Jalote, “An Integrated Approach to Software Engineering” Third addition, Springer Press

Other useful resource(s):

1.Link to NPTEL course contents:https://nptel.ac.in/courses/106105182

EvaluationScheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	2	2	1	1	1	2	2	2	**1.73**
CO-2	2	3	3	3	3	1	1	1	2	2	1	**2.00**
CO-3	2	2	2	2	3	1	1	1	2	2	1	**1.73**
CO-4	2	3	3	3	2	1	1	1	2	3	2	**2.09**
CO-5	1	2	2	1	3	2	1	1	1	1	1	**1.45**
Average	**1.8**	**2.4**	**2.4**	**2.2**	**2.6**	**1.2**	**1.0**	**1.0**	**1.8**	**2.0**	**1.4**	

Data Analytics Using R and Python

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: Elective

L-T-P: 2-0-0

Pre-requisite: Basic understanding of mathematics, including elementary statistics (mean, median, probability)

Course Objectives:

1. Gain a clear understanding of the basic concepts of data analytics and how data is used to solve real-world problems.
2. Learn to use R and Python for writing programs to clean, organize, and analyze data.
3. Develop skills to explore datasets and apply basic statistical techniques to draw meaningful conclusions.
4. Create clear and informative visualizations to represent data using popular R and Python libraries.
5. Work confidently with essential data analytics libraries such as NumPy, pandas, and tidyverse.
6. Apply analytical and predictive techniques to solve practical problems using real datasets.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the fundamental concepts of data analytics.	Understand
CO-2	Demonstrate the concepts of R & Python for data analytics.	Understand
CO-3	Apply advanced methods and their quantitative analysis for real-world problems.	Assessment
CO-4	Apply statistical methods for hypothesis testing and inference problems.	Assessment
CO-5	Analyze, visualize and interpret the results for useful insights.	Analyze

Course Contents:

Unit	Contents	Lectures required
1	Data Definitions and Analytical Programming Techniques: Introduction to Data Analytics, Elements, Variables, and Data categorization, Levels of Measurement, Introduction to analytical programming languages R & Python, and Installing Software & Setting up, Lists & Dictionaries, Functions & Packages, Data frame, Import and Export data, Data Preprocessing.	**10**
2	Parametric &Non Parametric Tests: Hypothesis Testing, Assumption Testing, T-Test, Power Analysis, ANOVA, Fitting ANOVA Model in Python & R, Wilcoxon Tests, Mann-Whitney U Test	**6**

3	Correlation & Association Analysis: Pearson Correlation, Spearman Correlation, Kendall Tau Correlation, Affinity Analysis & Market Basket Analysis APriori Algorithm, Association Rules, Frequent Pattern Analysis	**7**
4	Data Analysis Techniques: Introduction to Machine Learning, Applications of ML Library in R & Python for Supervised & Unsupervised Learning, Basic Neural Network, Transfer Function Models, Multivariate Time Series Analysis.	**5**
Total lectures		**28**

Suggested Text Book(s):

3. Navlani, A., Fandango, A. and Idris, I. (2021). Python Data Analysis – Third Ed. Packt Publishing Ltd.
4. Jake vanderPlas, Python Data Science Handbook – Essential Tools for Working with Data, O'Really
5. Media, 2022
6. David J. Pine, Introduction to Python for Science and Engineering, CRC Press, 2019.
7. Manoj Kumar Srivastava and Namita Srivastava, Statistical Inference – Testing of Hypotheses, PHI, 2014.
8. Kabacoff, Robert I. R in action: data analysis and graphics with R. Simon and Schuster, 2015.
9. Haider, M. (2015). Getting Started with Data Science: Making Sense of Data with Analytics. IBM Press.

Suggested Reference Book(s):

5. Doing Data Science, Straight Talk from The Frontline, Cathy O'Neil and Rachel Schutt, O'Reilly (2014).
6. Robert Johansson, Numerical Python – Scientific Computing and Data Science Applications with NumPy, SciPy and Matplotlib, Apress, 2019
7. Robert Sedgewick, Kevin Wayne, Robert Dondero, Introduction to Programming in Python: An Inter-disciplinary Approach, Pearson India Education Services Pvt. Ltd., 2016
8. Nelli, F., Python Data Analytics: with Pandas, NumPy and Matplotlib, Apress, 2018.
9. Wickham, H., &Grolemund, G. (2016). R for data science: import, tidy, transform, visualize, and model data. " O'Reilly Media, Inc.".

Other useful resource(s):

1. Link to NPTEL course contents:https://nptel.ac.in/courses/106107220

EvaluationScheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2

3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	2	2	1	1	1	2	2	2	**1.73**
CO-2	2	3	3	3	3	1	1	1	2	2	1	**2**
CO-3	2	2	2	2	3	1	1	1	2	2	1	**1.73**
CO-4	2	3	3	3	2	1	1	1	2	3	2	**2.09**
CO-5	1	2	2	1	3	2	1	1	1	1	1	**1.45**
Average	**1.8**	**2.4**	**2.4**	**2.2**	**2.6**	**1.2**	**1**	**1**	**1.8**	**2**	**1.4**	

Data Analytics using R and Python Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Programming Language Fundamentals

Course Objectives:

1. An understanding of fundamental concepts of data analytics and its importance in real-world applications.
2. Proficiency with R and Python programming languages for data analysis.
3. Skills with data collection, cleaning, preprocessing, and transformation techniques.
4. Experience with exploratory data analysis and effective data visualization methods.
5. Knowledge with basic statistical methods and machine learning algorithms used in analytics.
6. Ability with interpreting analytical results for data-driven decision making.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Apply fundamental concepts of data analytics to understand and analyze real-world datasets.	Apply
CO2	Use R and Python programming languages to perform data manipulation, cleaning, and preprocessing tasks.	Apply
CO3	Perform exploratory data analysis and data visualization to extract meaningful insights from data.	Apply
CO4	Implement basic statistical techniques and machine learning algorithms for data analysis and prediction.	Assessment

List of Experiments

S. No	Description	Hours
1	Program to install, import, and use basic R and Python libraries (NumPy, Pandas, Matplotlib, ggplot2). Program to create and manipulate vectors, lists, matrices, and data frames in R.	**2**
2	Program to perform basic data types and control structures in Python. Program to read data from CSV/Excel files and display summary statistics.	**2**
3	Program to handle missing values (NA/NULL) using mean, median, or deletion methods.	**2**
4	Program to compute descriptive statistics (mean, median, mode, variance, SD). Program to perform data normalization and standardization.	**2**

5	Program to generate data visualizations: • Bar chart • Histogram • Box plot • Scatter plot	**2**
6	Program to detect and remove outliers using IQR/Z-score. Program to perform correlation and covariance analysis.	**2**
7	Program to filter, sort, and group data using Pandas and dplyr. Program to perform data aggregation and pivot operations.	**2**
8	Program to merge and join multiple datasets.	**2**
9	Program to apply string manipulation and date-time operations.	**2**
10	Program to perform feature selection and feature scaling.	**2**
11	Program to perform probability distribution analysis.	**2**
12	Program to implement hypothesis testing (t-test, z-test, chi-square test).	**2**
13	Project	**4**
Total Lab hours		**28**

Suggested/Resources:

1. Wes McKinney, Python for Data Analysis, O'Reilly Media.
2. Hadley Wickham & Garrett Grolemund, R for Data Science, O'Reilly Media.
3. Anil Maheshwari, Data Analytics Made Accessible, McGraw Hill.
4. Gareth James et al., An Introduction to Statistical Learning, Springer.
5. Link to topics related to course:
   i. https://www.bigbookofr.com/
   ii. https://rstudio-education.github.io/hopr/

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects, etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	3	1	1	1	1	1	2	**1.73**
CO-2	3	3	3	3	3	2	1	2	2	1	2	**2.27**
CO-3	3	3	2	3	3	2	1	2	2	1	2	**2.18**
CO-4	3	3	3	3	3	2	1	2	2	1	2	**2.27**
Average	**3**	**2.75**	**2.5**	**2.75**	**3**	**1.75**	**1**	**1.75**	**1.75**	**1**	**2**	

Mobile Application Development

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Basic Programming (C/C++/Java), OOP Concepts

Course Objectives:

1. To understand mobile application development concepts, platforms, and architecture.
2. To develop basic Android applications using activities, intents, and UI components.
3. To design user-friendly interfaces using layouts and widgets.
4. To implement event-driven programming in mobile applications.
5. To integrate database and device features in mobile applications.
6. Understand basic mobile application security practices, including permissions and secure data handling.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the mobile application development lifecycle, Android architecture, and core components	Understand
CO-2	Apply concepts of activities, intents, and event handling to develop simple apps	Apply
CO-3	Design user interfaces using layouts, widgets, and navigation components	Create
CO-4	Analyze and implement advanced UI components, data handling techniques, and basic application security practices	Analyze
CO-5	Develop mobile applications integrating databases, device services, and secure data handling mechanisms	Create

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Mobile App Development: Mobile platforms overview (Android), app development lifecycle, Android architecture, setting up development environment (Android Studio, SDK), project structure, Activities, Activity lifecycle, Intents (explicit & implicit), manifest file, passing data between activities	**5**
2	Event Handling and Interaction: Event-driven programming, handling button clicks, listeners (onClick, onLongClick), input	**4**

	handling, simple app development using event listeners	
3	User Interface Design – I: Views and ViewGroups, Layouts (Linear, Relative, Constraint), TextView, EditText, ImageView, ListView, GridView, Spinner, XML-based UI design, Adapter classes	**6**
4	User Interface Design – II: Advanced UI components: RadioButton, CheckBox, ToggleButton, ProgressBar, Menus, Navigation (Tabs, Navigation Drawer basics), Date & Time Picker, handling gestures (swipe, pinch, zoom)	**5**
5	Advanced UI and Device Features: Customizing ListView, GridView, RecyclerView (intro), filtering data, sending SMS/email (conceptual), introduction to location-based services and Google Maps API	**4**
6	Mobile Database (SQLite): Introduction to SQLite, database creation, CRUD operations, content values, simple app with database integration, data persistence. Application security fundamentals: Android permissions model, runtime permissions, secure storage of user data, data privacy principles, secure coding practices.	**4**
Total lectures		**28**

Suggested Text Book(s):

1. Wei-Meng Lee, Beginning Android Programming, Wiley
2. Bill Phillips et al., Android Programming: The Big Nerd Ranch Guide, Big Nerd Ranch

Suggested Reference Book(s):

1. Reto Meier, Professional Android Development, Wiley
2. Jerome (J.F.) DiMarzio, Android: A Programmer's Guide, McGraw-Hill

Other useful resource(s):

1. Android Developer Documentation: https://developer.android.com
2. NPTEL Courses on Mobile Computing / Android Development: https://nptel.ac.in/courses/106106147

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered up to T-1
2	T-2	25	1.5 Hours	Syllabus covered up to T-2
3	T-3	35	2 Hours	Entire Syllabus

4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	2	1	1	1	2	1	2	**1.5**
CO-2	3	3	2	2	3	1	1	1	2	1	2	**1.9**
CO-3	3	2	3	2	3	1	1	2	3	1	2	**2.0**
CO-4	2	3	2	3	3	1	2	1	2	2	2	**2.1**
CO-5	3	2	3	2	3	1	2	2	3	2	3	**2.4**
Average	**2.8**	**2.4**	**2.2**	**2**	**2.8**	**1**	**1.4**	**1.4**	**2.4**	**1.4**	**2.2**	

Mobile Application Development Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Basic Programming, OOP Concepts

Course Objectives:

1. To provide hands-on experience in developing Android applications.
2. To implement activities, intents, and event-driven programming.
3. To design interactive user interfaces using layouts and widgets.
4. To develop applications using mobile databases (SQLite).
5. To integrate device features and services into mobile applications.
6. Implement basic mobile application security practices, including permission handling and secure data storage.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Develop basic Android applications using activities and intents	Apply
CO-2	Implement event handling and user interaction in mobile apps	Apply
CO-3	Design user interfaces using layouts, views, and adapters	Create
CO-4	Develop applications with advanced UI components, device features, and secure handling of permissions	Apply
CO-5	Implement database-driven applications using SQLite and basic secure data handling practices	Apply

List of Experiments

S. No	Description	Hours
1	Installation and setup of Android Studio, SDK, emulator; creating the first Android application (Hello World)	**2**
2	Study of Android project structure, Activity lifecycle demonstration using a sample app	**2**
3	Developing an application using Intents (explicit and implicit) and passing data between activities	**2**
4	Event handling: Button clicks, listeners (onClick), simple calculator application	**2**
5	Designing UI using Layouts (Linear, Relative, Constraint) with XML	**2**
6	Implementing UI elements: TextView, EditText, and ImageView with user input handling	**2**

7	Implementing ListView/GridView using Adapter classes	**2**
8	Creating an application using a Spinner and handling item selection events	**2**
9	Implementing advanced UI controls: RadioButton, CheckBox, ToggleButton, ProgressBar	**2**
10	Working with Menus and Navigation (Options menu / Context menu)	**2**
11	Designing an app with DatePicker and TimePicker	**2**
12	Customizing ListView/introduction to RecyclerView	**2**
13	SQLite database: Create database, insert and retrieve data (simple CRUD app).Implement basic secure storage practices (e.g., avoiding hardcoded data, using internal storage/SharedPreferences securely)	**2**
14	Mini Application: Develop a small Android app integrating UI + database (e.g., student record/notes app). Include runtime permission handling (e.g., storage/location) and basic secure data handling	**2**
Total Lab hours		**28**

Suggested/Resources:

1. Wei-Meng Lee, Beginning Android Programming, Wiley
2. Bill Phillips et al., Android Programming: The Big Nerd Ranch Guide

Other useful resource(s):

1. Android Developer Documentation: https://developer.android.com

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects, etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	3	1	1	1	2	1	2	**1.7**
CO-2	3	2	2	2	3	1	1	1	2	1	2	**1.8**
CO-3	3	2	3	2	3	1	1	2	3	1	2	**2.1**
CO-4	3	2	3	2	3	1	2	2	2	1	2	**2.1**
CO-5	3	3	3	2	3	1	2	2	2	2	3	**2.4**
Average	**2.5**	**2.2**	**2.6**	**1.8**	**3**	**1**	**1.4**	**1.6**	**2.2**	**1.2**	**2.2**	

Fundamentals of Smart System and IoT

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: Elective

L-T-P: 2-0-0

Pre-requisite: Basic Programming Knowledge, Basic Digital Electronics, Basic Computer Networking

Course Objectives:

1. Develop a foundational understanding of the Internet of Things (IoT), including smart sensors, communication paradigms, and their role in smart applications such as smart homes, smart cities, healthcare, and industrial automation.
2. Explain the architecture, communication protocols, and core components of IoT systems, along with the operational principles of Arduino microcontroller-based platforms.
3. Apply sensor interfacing techniques and Arduino programming skills to design, implement, and control basic IoT-enabled smart devices.
4. Analyze the performance characteristics, limitations, and suitability of different sensors and communication modules for various IoT-based smart system applications.
5. Demonstrate the design and working of real-world IoT applications in domains such as smart homes, healthcare, agriculture, and intelligent transportation systems.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Define the basic concepts of the Internet of Things (IoT), smart sensors, and their applications in various smart systems.	Understand
CO-2	Explain the architecture, protocols, and components of IoT systems and describe the working of Arduino microcontroller-based setups.	Understand
CO-3	Apply the knowledge of different sensors and Arduino programming to interface and control smart devices.	Apply
CO-4	Analyze the characteristics and performance of various sensors and communication modules used in IoT-based smart systems.	Apply
CO-5	Illustrate the design and operation of IoT applications in fields such as smart homes, healthcare, agriculture, and transportation.	Apply

Course Contents:

Unit	Contents	Lectures required

1	Introduction to Smart Sensors and IoT: Introduction to IoT: Definition, evolution, enabling technologies. Smart Sensors: concept, role in IoT. Measuring and monitoring environmental conditions. Types of smart systems and application domains (home, healthcare, agriculture, transport)	**3**
2	IoT Architecture, Protocols, and Arduino Microcontroller IoT system architecture (device, gateway, cloud, application layers). Bread board and its internal connections. IoT protocol stack: Physical, MAC, Network, Transport, Application layers. Overview of common IoT protocols. Arduino overview: board components, microcontroller basics. Arduino development environment and basic operation.	**5**
3	Sensors and Their Characteristics: Sensor classification and working principles. Capacitive, resistive, SAW sensors. Sensors for temperature, pressure, humidity, gas. Static characteristics: sensitivity, accuracy, resolution. Dynamic characteristics: response time, bandwidth. Smart sensor architecture and interfacing challenges.	**5**
4	Design of Smart Systems: Embedded system design using Arduino. LED, tricolor LED, LCD, seven-segment display concepts. Push button interfacing. Pulse Width Modulation (PWM). Environmental sensors (DHT11).	**5**
5	Wireless Smart Systems: ESP8266-01 Wi-Fi module. IoT system components for cloud connectivity. ThingSpeak platform overview. MQTT protocol fundamentals. Introduction to low-level programming (AVR-GCC concepts).	**5**
6	IoT Applications: Smart Home systems: components, benefits, challenges. Smart Cities and Intelligent Transportation Systems. Smart Healthcare and wearable systems. Smart agriculture and precision farming. Smart factories and Industry 4.0. Societal, ethical, and privacy challenges	**5**
Total lectures		**28**

Suggested Text Book(s):

1. Advances in Modern Sensors; Physics, design, simulation and applications by Sinha, G, R, IOP (Institute of Physics Publishing), 2020
2. Internet of Things: Architecture and Design Principles, Raj Kamal, McGrawHill. 2017
3. JanHo"ller,VlasiosTsiatsis,CatherineMulligan,Stamatis,Karnouskos,StefanAvesand,David Boyle, "From Machine-to-Machine to the Internet of Things -Introduction to a New Age of Intelligence",Elsevier,2014.

Suggested Reference Book(s):

1. ArshdeepBahga, Vijay Madisetti, "Internet of Things – A hands-on approach", Universities Press, 2015
2. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton, Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things", CISCO Press, 2017.

Other useful resource(s):

1. https://spoken-tutorial.org/tutorial-search/?search_foss=Arduino&search_language=English
2. https://www.ibm.com/smarterplanet/us/en/?ca=v_smarterplanet

EvaluationScheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	2	2	1	1	1	2	2	2	**1.73**
CO-2	2	3	3	3	3	1	1	1	2	2	1	**2**
CO-3	2	2	2	2	3	1	1	1	2	2	1	**1.73**
CO-4	2	3	3	3	2	1	1	1	2	3	2	**2.09**
CO-5	1	2	2	1	3	2	1	1	1	1	1	**1.45**
Average	**1.8**	**2.4**	**2.4**	**2.2**	**2.6**	**1.2**	**1**	**1**	**1.8**	**2**	**1.4**	

Fundamentals of Smart System and IoT Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Basic Programming Knowledge, Basic Digital Electronics, Basic Computer Networking

Course Objectives:

1. Introduce students to the fundamental concepts of the Internet of Things (IoT) through hands-on interaction with embedded hardware, sensors, and actuators.
2. Enable students to acquire practical skills in interfacing sensors, displays, and actuators using microcontroller platforms such as Arduino and ESP32.
3. Familiarize students with basic networking concepts and cloud connectivity by implementing Wi-Fi–enabled data transmission and cloud-based data visualization.
4. Develop an understanding of edge computing concepts through Linux-based single-board computers by configuring Raspberry Pi for data acquisition and local storage.
5. Prepare students to design and analyze simple end-to-end IoT systems by integrating sensing, communication, data processing, and visualization components.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	To identify and explain basic components of IoT systems	Understand
CO2	To interface sensors, displays, and actuators using microcontroller platform	Apply
CO3	To implement wired and wireless IoT systems for sensor data transmission	Apply
CO4	To configure edge-based IoT systems using Raspberry Pi	Apply

List of Experiments

S. No	Description	Hours
1	Arduino IDE Setup and Digital I/O Test: Blink LED and Serial Monitor Output	**2**
2	Push Button and LED Control on Arduino: Toggle/Counter with Debouncing	**2**
3	Analog Input using Potentiometer/LDR: Sensor Value Reading and Mapping in Arduino	**2**
4	LCD Interfacing with Arduino: Display Sensor Readings (Potentiometer/LDR/DHT11)	**2**
5	PWM with Arduino: LED Brightness Control using Potentiometer (AnalogRead to AnalogWrite)	**2**

6	DHT11 Sensor Interfacing with Arduino: Temperature/Humidity Readings and Serial Logging	**2**
7	ESP8266 Wi-Fi Interfacing: Connect to Wi-Fi, Obtain IP, and Test Connectivity (Ping/Basic Web Request)	**2**
8	IoT Data Upload using ESP8266 and DHT11: Send Sensor Data to ThingSpeak using HTTP GET/POST	**2**
9	Cloud Dashboard on ThingSpeak: Visualize DHT11 Data, Field Configuration, and Update-Rate Analysis	**2**
10	Raspberry Pi Setup and GPIO with Python: Blink LED and Read Push Button (RPi.GPIO / gpiozero)	**2**
11	Raspberry Pi as IoT Gateway: Read DHT11/DS18B20 and Store Data in CSV/SQLite with Timestamp	**2**
12	Mini IoT Project (ESP8266/Raspberry Pi): Sensor Node and Cloud/Local Dashboard (Smart Home/Agriculture/Health)	**6**
Total Lab hours		**28**

Suggested/Resources:

1. Advances in Modern Sensors; Physics, design, simulation and applications by Sinha, G, R, IOP (Institute of Physics Publishing), 2020
2. Internet of Things: Architecture and Design Principles, Raj Kamal, McGrawHill. 2017
3. Jan Ho"ller, VlasiosTsiatsis, Catherine Mulligan, Stamatis, Karnouskos, Stefan Avesand, David Boyle, "From Machine-to-Machine to the Internet of Things -Introduction to a New Age of Intelligence", Elsevier, 2014.
4. ArshdeepBahga, Vijay Madisetti, "Internet of Things – A hands-on approach", Universities Press, 2015
5. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton, Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things", CISCO Press, 2017.
6. Link to topics related to course:

iii. https://spoken-tutorial.org/tutorial-search/?search_foss=Arduino&search_language=English
iv. https://www.ibm.com/smarterplanet/us/en/?ca=v_smarterplanet
v. https://www.arduino.cc/en/Tutorial/HomePage
vi. https://www.raspberrypi.org/documentation/
vii. "InternetofThings:AHands-onApproach",byArshdeepBahgaandVijayMadisetti

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	3	1	1	1	1	1	2	**1.73**
CO-2	3	3	3	3	3	2	1	2	2	1	2	**2.27**
CO-3	3	3	2	3	3	2	1	2	2	1	2	**2.18**
CO-4	3	3	3	3	3	2	1	2	2	1	2	**2.27**
Average	**3**	**2.75**	**2.5**	**2.75**	**3**	**1.75**	**1**	**1.75**	**1.75**	**1**	**2**	

Operating Systems

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

L-T-P: 3-0-0

Pre-requisite: Programming inC/C++

Course Objectives:

1. To learn the mechanisms of OS to handle processes and threads and their communication.
2. To learn the mechanisms involved in memory management in contemporary OS
3. To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols.
4. To know the components and management aspects of concurrency management

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To appreciate the role of an operating system.	Remember, Understand
CO-2	Create processes and threads.	Create, Evaluate
CO-3	Develop algorithms for process scheduling for a given specification of CPU utilization, Throughput, Turnaround Time, Waiting Time, and Response Time.	Create, Evaluate
CO-4	For a given specification of memory organization develop the techniques for optimally allocating memory to processes by increasing memory utilization and for improving the access time	Evaluate
CO-5	Design and implement file management system.	Create, Evaluate
CO-6	For a given I/O devices and OS (specify) develop the I/O management functions in OS as part of a uniform device abstraction by performing operations for synchronization between CPU and I/O controllers.	Apply

Course Contents:

Unit	Contents	Lectures required
1	Introduction and Historical context of Operating Systems, Operating Structure and Architecture	**3**
2	Process: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching. Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads.	**3**

3	Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling Algorithms: Pre-emptive and Non preemptive, FCFS, SJF, Priority, Round Robin, Multiple processor scheduling	**7**
4	Inter-process Communication: Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson's Solution, Lamport's Bakery Algorithm, The Producer\ Consumer Problem, Semaphores, Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem etc. Monitors	**10**
5	Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.	**4**
6	Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation – Fixed and variable partition–Internal and External fragmentation and Compaction; Paging: Principle of operation – Page allocation – Hardware support for paging, Protection and sharing, Disadvantages of paging.	**6**
7	Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault, Working Set, Dirty page/Dirty bit – Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Not recently used (NRU) and Least Recently used (LRU).	**4**
8	Case studies of OS: Windows, Linux, IBM, Tizen Operating System, Distributed O.S: Introduction to distributed operating systems, synchronization and deadlock in distributed systems	**2**
Total lectures		**42**

Suggested Text Book(s):

1. "Operating System Concepts" 9th Edition by Silberschatz, Peter Galvin, Greg Gagne, Wiley Asia Student Edition.
2. "Operating Systems: Internals and Design Principles" 9th Edition, William Stallings, Pearson.

Suggested Reference Book(s):

1. "Operating System: A Design-oriented Approach, 1st Edition by Charles Crowley, Irwin Publishing
2. "Operating Systems: A Modern Perspective" 2nd Edition by Gary J. Nutt, Addison-Wesley
3. Design of the Unix Operating Systems, 8th Edition by Maurice Bach, Prentice-Hall of India
4. Understanding the Linux Kernel, 3rd Edition, Daniel P. Bovet, Marco Cesati, O'Reilly and Associates

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/106108101/
2. Link to topics related to course:
   a. https://nptel.ac.in/courses/106106144/
   b. https://www.class-central.com/course/udacity-introduction-to-operatingsystems-3419

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) -10 Attendance - 5

Course Outcomes (COs) contribution to the ProgrammeOutcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	3	3	2	2	3	2	2	3	2	3	**2.5**
CO-2	3	3	3	2	3	2	2	2	3	2	3	**2.5**
CO-3	3	3	3	2	2	3	2	3	3	2	3	**2.6**
CO-4	3	3	3	2	3	3	2	3	3	2	3	**2.7**
CO-5	3	3	3	2	3	3	2	3	3	2	3	**2.7**
CO-6	3	3	3	2	3	3	2	2	3	2	3	**2.6**
Average	**3**	**3**	**3**	**2**	**2.7**	**2.8**	**2**	**2.5**	**3**	**2**	**3**	

Operating Systems Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: Programming inC/C++

Course Objectives:

1. To write programs in Linux environment using system calls.
2. To implement the scheduling algorithms.
3. To implement page replacement algorithms
4. To implement file allocation methods.
5. To understand and implement ipc mechanism using named and unnamed pipes.
6. To develop solutions for synchronization problems using semaphores.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Understand Various Unix Commands	Understand
CO2	Develop programs to create different types of processes using pthread library under Linux environment.	Apply
CO3	Develop programs to implement resource management task like CPU scheduling algorithms, deadlock handling.	Apply
CO4	Develop programs to implement and test various synchronization techniques like semaphores, binary semaphore and monitors via different classical test suites.	Apply
CO5	Design and analyse various disk-scheduling algorithms, memory management schemes, file management systems.	Analyze

List of Experiments

S. No	Description	Hours
1	Learning Unix Commands (file commands, directory commands, symbolic links, terminal commands, help commands, information commands, useful cshell symbols, permissions and file storage (Unix), permissions and file storage (andrew), processes, printing Environment, customizing networking, x-applications unix filters)	**6**
2	Develop programs to create different types of processes under Linux environment. Develop programs to create multitasking threads using pthread library under Linux environment. Develop programs to implement inter-process communication	**4**

3	Develop programs to implement resource management task like CPU scheduling algorithms (First Come First Serve, Shortest Job First, Round Robin, Priority Scheduling, Multi level Queue), deadlock handling (Prevention, Avoidance and Detection)	**8**
4	Develop programs to implement and test various synchronization techniques like semaphores, binary semaphore and monitors via different classical test suites.	**4**
5	Design and analyze various disk-scheduling algorithms. Develop programs to implement memory management schemes. Design, implement and assess file management systems (file organization and file directories).	**6**
Total Lab hours		**28**

Suggested/Resources:

1. Charles Crowley "Operating System A Design Approach"TMH.
2. Andrew S. Tanenbaum "Operating Systems Design and Implementation", Third Edition, Prentice Hall Publications2006
3. A.S. Tanenbaum, "Modern Operating Systems", 2nd edition, Prentice Hall India.
4. A.Silberschatz, P.Galvin, G. Gagne, "Operating systems concepts" Willey international company (sixth edition)
5. Gary Nutt, "Operating Systems – A modern perspective", Pearson Education
6. David Solomon and Mark Russinovich ," Inside Microsoft Windows 2000", Third Edition, Micorosoft Press
7. D. M. Dhamdhere, " Systems Programming and Operating systems" TMH, 2nd revised edition.2006
8. ACM/IEEE transactions on operating systems concepts.
9. www.vmware.com

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	3	1	1	1	1	1	2	**1.7**
CO-2	3	3	3	3	3	2	1	2	2	1	2	**2.2**
CO-3	3	3	2	3	3	2	1	2	2	1	2	**2.1**
CO-4	3	3	3	3	3	2	1	2	2	1	2	**2.2**
CO-5	3	2	3	3	3	3	2	1	2	2	2	**2.3**
Average	**3**	**2.6**	**2.6**	**2.8**	**3**	**2**	**1.2**	**1.6**	**1.8**	**1.2**	**2**	

Fundamentals of Soft Computing

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Basic Mathematics, Programming Fundamentals

Course Objectives:

1. To understand the fundamental concepts of soft computing and its components.
2. To study fuzzy logic systems and their applications in decision-making.
3. To learn artificial neural networks and their learning mechanisms.
4. To explore evolutionary algorithms such as genetic algorithms for optimization.
5. To apply hybrid soft computing techniques to real-world problems.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the concepts of soft computing, fuzzy logic, and neural networks	Understand
CO-2	Apply fuzzy logic principles to solve decision-making problems	Apply
CO-3	Implement artificial neural network models for classification and prediction	Apply
CO-4	Analyze optimization problems using genetic algorithms and evolutionary techniques	Analyze
CO-5	Design hybrid soft computing models for real-world applications	Create

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to Soft Computing:** Concept of Soft Computing, Difference between Hard and Soft Computing, Components of Soft Computing, Applications in engineering and AI	2
2	**Fuzzy Logic:** Fuzzy Sets, Membership Functions, Operations on Fuzzy Sets, Fuzzy Relations, Fuzzy Rules and Inference Systems, Defuzzification Methods	5
3	**Artificial Neural Networks (ANN):** Biological Neuron Model, ANN Architecture, Types of Neural Networks, Learning Rules, Perceptron, Multilayer Perceptron, Backpropagation Algorithm	6

4	**Genetic Algorithms (GA):** Introduction to Evolutionary Computation, Genetic Algorithm Components, Selection, Crossover, Mutation, Fitness Function, Applications of GA	**4**
5	**Hybrid Systems:** Neuro-Fuzzy Systems, Genetic-Fuzzy Systems, Applications in pattern recognition, optimization, and control systems	**4**
6	**Swarm Intelligence:** Introduction to Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Applications in optimization problems	**4**
7	**Applications of Soft Computing:** Case studies in image processing, medical diagnosis, robotics, and data mining	**3**
Total lectures		**28**

Suggested Text Book(s):

1. S. N. Sivanandam& S. N. Deepa, Principles of Soft Computing, Wiley India
2. K. T. Atanassov, Intuitionistic Fuzzy Sets, Springer
3. J. S. R. Jang, C. T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing, Prentice Hall
4. Simon Haykin, Neural Networks and Learning Machines, Pearson

Suggested Reference Book(s):

1. David E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Pearson
2. Timothy J. Ross, Fuzzy Logic with Engineering Applications, Wiley
3. Laurene Fausett, Fundamentals of Neural Networks, Pearson

Other useful resource(s):

1. NPTEL: Soft Computing – https://nptel.ac.in
2. Coursera: Neural Networks and Deep Learning
3. TutorialsPoint: Soft Computing Tutorials

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.4**	**2.4**	**2.0**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.2**	

Fundamentals of Soft Computing Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: Fundamentals of Soft Computing (Theory), Basic Programming (Python/MATLAB preferred)

Course Objectives:

1. To implement fuzzy logic systems for decision-making problems.
2. To design and simulate artificial neural networks.
3. To apply genetic algorithms for optimization problems.
4. To explore swarm intelligence techniques for real-world applications.
5. To develop hybrid soft computing models using modern tools.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Implement fuzzy logic systems including fuzzification and defuzzification	Apply
CO2	Design and train artificial neural networks using suitable tools	Apply
CO3	Apply genetic algorithms to solve optimization problems	Apply
CO4	Implement swarm intelligence techniques such as PSO and ACO	Apply
CO5	Develop hybrid soft computing models for real-world applications	Apply

List of Experiments

S. No	Description	Hours
1	Introduction to Python/MATLAB environment for Soft Computing (NumPy, SciPy, MATLAB toolbox basics)	2
2	Implementation of basic fuzzy set operations (union, intersection, complement) and membership functions	3
3	Design of fuzzy inference system (Mamdani model) for decision-making problems (e.g., temperature control)	3
4	Implementation of defuzzification techniques (centroid, bisector, mean of maxima)	2
5	Implementation of single-layer perceptron model	3

6	Implementation of Multilayer Perceptron (MLP) using backpropagation algorithm	3
7	Training ANN for classification problems (e.g., Iris dataset / plant leaf dataset)	3
8	Implementation of Genetic Algorithm for function optimization (selection, crossover, mutation)	3
9	Implementation of Particle Swarm Optimization (PSO) for optimization problems	3
10	Mini Project: Hybrid model (Neuro-Fuzzy / GA-ANN) for real-world problem (e.g., plant classification, prediction, optimization)	3
Total Lab hours		**28**

Suggested/Resources:

1. S. N. Sivanandam& S. N. Deepa, Principles of Soft Computing, Wiley
2. J. S. R. Jang, Neuro-Fuzzy and Soft Computing, Prentice Hall
3. Simon Haykin, Neural Networks, Pearson
4. Timothy J. Ross, Fuzzy Logic with Engineering Applications, Wiley
5. Python Libraries: NumPy, TensorFlow/Keras, Scikit-learn
6. MATLAB Fuzzy Logic Toolbox & Neural Network Toolbox

Other useful resource(s):

1. NPTEL Soft Computing Course – https://nptel.ac.in
2. Coursera – Neural Networks and Deep Learning
3. Scikit-learn Documentation – https://scikit-learn.org

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	3	1	3	1	1	2	2	1	2	**1.9**
CO-2	3	3	3	1	3	1	1	2	2	1	2	**2.0**
CO-3	3	2	2	1	3	1	1	1	1	1	2	**1.6**
CO-4	3	2	2	2	2	1	1	1	1	1	2	**1.6**
CO-5	3	2	3	2	3	1	1	1	2	2	2	**2.0**
Average	**3.0**	**2.2**	**2.6**	**1.4**	**2.8**	**1.0**	**1.0**	**1.4**	**1.6**	**1.2**	**2.0**	

Fundamentals of Computer and Cyber Security

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Computer Fundamentals

Course Objectives:

1. To introduce fundamental concepts of computer and cyber security, including threats, vulnerabilities, and the CIA triad.
2. To understand and demonstrate cryptographic techniques for securing data and communication.
3. To apply security mechanisms to networks and operating systems, and analyze common attack scenarios.
4. To evaluate security policies, ethical issues, and governance frameworks in organizational and legal contexts.
5. To design basic security solutions and incident response strategies for emerging technologies such as cloud, IoT, and mobile systems.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Recall and explain key cybersecurity concepts, terminology, and models (CIA triad, threats, vulnerabilities).	Remember, Understand
CO-2	Demonstrate the use of cryptographic algorithms (AES, RSA, hashing, digital signatures) in securing communication.	Apply
CO-3	Analyze network/system vulnerabilities and propose defense mechanisms (firewalls, IDS/IPS, secure coding).	Analyze
CO-4	Evaluate cybersecurity policies, ethical dilemmas, and compliance requirements (IT Act, GDPR, ISO frameworks).	Evaluate
CO-5	Design and implement basic security solutions for emerging domains (cloud, IoT, mobile) and incident response plans.	Create

Course Contents:

Unit	Contents	Lectures required
1	**Introduction:** Definition and scope of cyber security, CIA triad: Confidentiality, Integrity, Availability, Threats, vulnerabilities, exploits, Risk management and security models, Case studies of real-world breaches	**4**
2	**Cryptography Essentials:** Classical ciphers (Caesar, Vigenère), Symmetric encryption (DES, AES), Asymmetric encryption (RSA, ECC), Hash functions (SHA family, MD5), Digital signatures and certificates, Public Key Infrastructure (PKI), Applications in secure communication	**6**

3	**Network Security Fundamentals:** OSI and TCP/IP security layers, Firewalls: packet filtering, stateful inspection, Intrusion Detection/Prevention Systems (IDS/IPS), Virtual Private Networks (VPNs), Secure protocols: TLS, SSH, HTTPS, Common attacks: DoS, DDoS, MITM, sniffing, Network monitoring and traffic analysis	**6**
4	**System Security:** Operating system hardening (Windows/Linux), User authentication and access control, Secure coding practices, Application vulnerabilities: buffer overflow, SQL injection, XSS, Patch management and updates, Malware types and defense strategies	**4**
5	**Security Policies and Management:** Security policies and frameworks (ISO 27001, NIST), Cyber laws: Indian IT Act, GDPR basics, Ethical hacking and penetration testing, Governance, compliance, and audit processes, Incident reporting and accountability	**4**
6	**Emerging Issues in Cyber Security:** Cloud security fundamentals (shared responsibility model), IoT security challenges (device vulnerabilities, botnets), Mobile security (app permissions, sandboxing), AI and machine learning in threat detection, Incident response planning and disaster recovery, Future trends: quantum cryptography, zero-trust architecture	**4**
Total lectures		**28**

Suggested Text Book(s):

1. James Graham, Richard Howard, Ryan Olson – Cyber Security Essentials, CRC Press, Taylor & Francis Group, 2011.
2. William Stallings & Lawrie Brown – Computer Security: Principles and Practice, Pearson (5th Edition), 2024.

Suggested Reference Book(s):

1. Charles P. Pfleeger& Shari Lawrence Pfleeger, Security in Computing, Pearson, 2015, 5th Edition.
2. David Kim & Michael G. Solomon, Fundamentals of Information Systems Security, Jones & Bartlett Learning, 2018, 3rd Edition
3. Mark Stamp, Information Security: Principles and Practice, Wiley, 2019, 3rd Edition

Other useful resource(s):

1. NIST Cyber security Framework (2024 update): https://www.nist.gov/cyberframework
2. OWASP (Open Web Application Security Project): https://owasp.org
3. SANS Institute Reading Room (whitepapers & case studies): https://www.sans.org/white-papers
4. CERT-In (Indian Computer Emergency Response Team): https://www.cert-in.org.in
5. MIT OpenCourseWare – Computer Systems Security: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-858-computer-systems-security-fall-2014/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination

1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	1	1	–	–	–	2	2	**1.5**
CO-2	3	2	2	2	3	–	–	–	–	2	2	**2.3**
CO-3	3	3	3	3	3	2	2	–	2	2	2	**2.5**
CO-4	2	2	1	1	1	3	2	3	2	3	2	**2.0**
CO-5	3	3	3	3	3	3	3	3	3	3	3	**3.0**
Average	**2.8**	**2.4**	**2.0**	**2.0**	**2.2**	**2.3**	**2.3**	**3.0**	**2.3**	**2.4**	**2.3**	

Fundamentals of Computer and Cyber Security Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Computer Fundamentals

Course Objectives:

1. To familiarize with cyber security tools, environments, and basic commands for system and network monitoring.
2. To implement cryptographic algorithms and security mechanisms through hands-on coding and demonstrations.
3. To examine network traffic, system vulnerabilities, and application flaws using practical experiments.
4. To assess the effectiveness of firewalls, secure protocols, and hardening techniques in controlled lab scenarios.
5. To develop and document incident response strategies, security policies, and prototype solutions for emerging technologies.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Identify and recall cyber security tools, commands, and environments used in lab practice.	Remember, Understand
CO2	Apply cryptographic techniques (AES, RSA, hashing) to secure files and communications.	Apply
CO3	Analyze captured network traffic and system/application vulnerabilities using Wireshark, firewalls, and test web apps.	Analyze
CO4	Evaluate the effectiveness of OS hardening, secure protocols, and penetration testing methods.	Evaluate
CO5	Create incident response plans, draft security policies, and design prototype solutions for cloud, IoT, and mobile security.	Create

List of Experiments

S. No	Description	Hours
1	**System Reconnaissance & Vulnerability Scanning** • Install Kali Linux / security tools, • Perform basic system reconnaissance using Nmap, • Identify open ports and potential vulnerabilities.	2
2	**Classical Cryptography Implementation** • Implement Caesar cipher and Vigenère cipher in Python. • Encrypt and decrypt sample text messages.	2

3	**Symmetric Encryption (AES)** • Use Python's cryptography library to implement AES. • Demonstrate secure file encryption and decryption. • Compare performance with different key sizes.	**4**
4	**Asymmetric Encryption (RSA) & Digital Signatures** • Generate RSA key pairs. • Encrypt and decrypt messages using public/private keys. • Demonstrate digital signatures and certificate validation.	**4**
5	**Hashing & Integrity Verification** • Implement SHA-256 hashing. • Verify file integrity using hash comparison. • Explore password hashing with bcrypt.	**2**
6	**Packet Sniffing & Traffic Analysis** • Capture and analyze network traffic using Wireshark. • Identify HTTP, DNS, and TCP packets. • Detect anomalies in traffic patterns.	**4**
7	**Firewall Configuration & Secure Protocols** • Configure firewall rules using iptables or Windows Firewall. • Demonstrate secure login using SSH. • Enable HTTPS/TLS for a sample web server.	**2**
8	**Web Application Vulnerabilities (SQL Injection & XSS)** • Demonstrate SQL injection on a test web app. • Show XSS attack and mitigation techniques.	**2**
9	**Operating System Hardening (2 hours)** • Configure user accounts, permissions, and password policies. • Apply patches and updates. • Demonstrate malware detection basics.	**2**
10	**Incident Response & Emerging Trends** • Configure multi-factor authentication. • Simulate a breach scenario and draft an incident response plan. • Explore IoT device vulnerability scanning.	**4**
Total Lab hours		**28**

Suggested/Resources:

1. James Graham, Richard Howard, Ryan Olson – Cyber Security Essentials, CRC Press, Taylor & Francis Group, 2011.
2. William Stallings & Lawrie Brown – Computer Security: Principles and Practice, Pearson (5th Edition), 2024.
3. Charles P. Pfleeger& Shari Lawrence Pfleeger, Security in Computing, Pearson, 2015, 5th Edition.
4. David Kim & Michael G. Solomon, Fundamentals of Information Systems Security, Jones & Bartlett Learning, 2018, 3rd Edition
5. Mark Stamp, Information Security: Principles and Practice, Wiley, 2019, 3rd Edition

Other useful resource(s):

1. Kali Linux Documentation – Official guide for penetration testing tools: https://www.kali.org/docs
2. Wireshark Labs – Packet analysis tutorials: https://www.wireshark.org/docs
3. OWASP Web Security Testing Guide – Hands-on web app vulnerability testing: https://owasp.org/www-project-web-security-testing-guide
4. Metasploit Unleashed (Offensive Security) – Free penetration testing labs: https://www.offensive-

security.com/metasploit-unleashed

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	2	–	–	–	–	2	2	**1.8**
CO-2	3	2	2	2	3	–	–	–	–	2	2	**2.3**
CO-3	3	3	3	3	3	2	2	–	2	2	2	**2.5**
CO-4	2	2	2	2	2	3	2	3	2	3	2	**2.3**
CO-5	3	3	3	3	3	3	3	3	3	3	3	**3.0**
Average	**2.8**	**2.4**	**2.2**	**2.2**	**2.6**	**2.3**	**2.3**	**3.0**	**2.3**	**2.4**	**2.4**	

Data Mining and Data Warehousing

COURSE CODE:XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Data Structures, Compilers, Operating Systems, Computer Networks, Machine Learning and Genetic Algorithms.

Course Objectives:

1. To describe the concept of Data warehouse & its attributes
2. To study different data warehouse models, architectures and implementation
3. To understand the basic concept of data mining and its functionality
4. To understand the concept of classification techniques and its implementation
5. To understand the concept of association rules, different techniques and implementation details
6. To understand the concept of cluster analysis, anomaly detection and its usage and implementation details

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understand data mining concepts, KDD process, data types, and preprocessing techniques.	Understand
CO-2	Understand data warehouse architecture, multi-dimensional models, schemas, and OLAP systems.	Understand
CO-3	Apply and evaluate classification techniques for solving predictive data mining problems.	Apply and Evaluate
CO-4	Analyze and implement association rule mining techniques to discover meaningful patterns in data.	Analyze
CO-5	Apply and assess clustering techniques for pattern discovery and data analysis.	Apply

Course Contents:

Unit	Contents	Lectures required
1	**Introduction:** Data Mining, Motivation, Challenges, Origins of Data Mining, Applications, Issues, Data Mining Tasks, stages of Knowledge discovery in database, Data: Types of Data, Data Quality, Measures of Similarity and Dissimilarity, Data Pre-processing, Measures of Similarity andDissimilarity, Exploring Different Datasets, Summary Statistics, Data Visualization.	**8**
2	**Architecture:** Basic Concept of Data Warehouse, DBMS vs. Data Warehouse, Data marts, Metadata, Multidimensional data model, Data	**5**

	Cubes, Schemas for Multidimensional Database: Stars, Snowflakes and Fact Constellations, Data Warehouse Architecture, ETL Process (Extraction, Transformation and Loading), Distributed and Virtual Data Warehouse, Data Warehouse Manager, OLTP, OLAP, MOLAP, HOLAP, types of OLAP servers.	
3	**Classification:** Basic Concepts and Preliminaries, Approach to Solving a Classification Problem, Decision Tree Induction: Iterative Dichotomiser 3, C4.5 Decision Tree Algorithms, Classification and Regression Trees, Model Overfitting, Evaluating Performance of binary and multi-class classifiers. Rule-Based Classifier, Ensemble Learning Methods: Bagging, Boosting, Staking and Voting-based classifiers, Class Imbalance Problem,	**5**
4	**Association Analysis:** Basic Concepts and Problem Definition, Frequent Itemset Generation, Rule Generation, Representation of Frequent Itemsets, FP-Growth Algorithm, Evaluation of Association Patterns, Sequential Patterns, Subgraph Patterns.	**5**
5	**Cluster Analysis:** Basic Concepts, Characteristics of Data, Clusters,Cluster Evaluation, Partitional Clustering: K-Means, K-Medoids, Hierarchical Clustering: Agglomerative and Divisive Clustering, Dendrogram Representation, Density-Based Clustering: DBSCAN and outlier detection, Graph-Based Clustering: Introduction and Applications	**5**
Total lectures		**28**

Suggested Text Book(s):

1. Data Mining Concepts and Techniques J. Han and M. Kamber Morgan Kaufmann, 2006, ISBN 1-55860 901-6
2. Introduction to Data Mining Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Pearson Education (Addison Wesley), 0-321-32136-7

Suggested Reference Book(s):

1. Mining Massive data sets Anand Rajaram, Jure Leskovec and Jeff Ullman Cambridge University Press 2

Other useful resource(s):

1. https://onlinecourses.nptel.ac.in/noc18_cs14/preview
2. https://www.coursera.org/specializations/data-mining.
3. https://www.futurelearn.com/courses/data-mining-with-weka

EvaluationScheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus

4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	2	2	3	1	1	1	2	2	2	**2.0**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-5	3	3	3	2	3	1	1	2	3	2	2	**2.3**
Average	**3**	**2.6**	**2.4**	**1.8**	**2.8**	**1**	**1**	**1.2**	**2.2**	**1.8**	**2**	

Data Mining and Data Warehousing Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Relational Database Management Systems (RDBMS), Python Programming

Course Objectives:

1. Practical exposure on implementation of data mining tasks using WEKA and Python Programming by applying well-known standard datasets
2. To learn, understand and implement various data preprocessing techniques
3. Learning performance evaluation on data mining predictive tasks
4. Learning performance evaluation on data mining descriptive tasks
5. Handling a small data mining project for a given practical domain

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Getting familiar with WEKA, ARFF file format, and Python Libraries	Understand
CO2	Apply data preprocessing techniques such as data cleaning, data reduction and data transformation, etc	Apply
CO3	Implementing and evaluating various data mining predictive tasks such as classification and regression	Evaluate
CO4	Implementing and evaluating various data mining descriptive tasks such as clustering and association rule mining	Evaluate
CO5	Create a group project using any of the data mining algorithms and evaluate the performance.	Create

List of Experiments

S. No	Description	Hours
1	To understand WEKA environment and to create and upload ARFF file in WEKA tool.	**2**
2	To implement data cleaning and data reduction techniques by applying various filters in WEKA.	**2**
3	To implement data transformation techniques such as normalization and discretization by applying various filters in WEKA.	**2**
4	• To compute measures of central tendency and measures of data dispersion using the Python NumPy library by loading a dataset in CSV format with the Pandas library. • To implement data visualization using Python Matplotlip library.	**2**
5	To implement various data preprocessing techniques by using Python libraries.	**2**

6	To implement decision trees classification algorithms using WEKA and Python Programming	**2**
7	To implement and compare different ensemble learning algorithms such as Random Forest, AdaBoost, Gradient Boosting, XGBoost, Voting Classifier, and Stacking using WEKA and Python Programming.	**4**
8	To implement and compare partition-based clustering algorithms such as K-Means and K-Medoids using WEKA and Python Programming.	**2**
9	To implement and analyze density-based clustering algorithms such as DBSCAN for cluster formation and outlier detection using WEKA and Python Programming.	**4**
10	To Apriori algorithm for Association Rule Mining using WEKA and Python Programming	**2**
11	Group project based on any of the data mining algorithms.	**4**
Total Lab hours		**28**

Suggested/Resources:

1. Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar. Introduction to data mining. Pearson Education India, 1st Edition, 2016.
2. J. Han and M. Kamber, Data Mining Concepts and Techniques. Morgan Kaufmann, 3rd Edition, ISBN 978-0-12-381479-1,
3. Witten, Ian H., and Eibe Frank. "Data mining: practical machine learning tools and techniques." Morgan Kaufmann, 2nd Edition, 2012.

Other useful resource(s):

4. Link to NPTEL course contents:https://nptel.ac.in/courses/106105174
5. https://www.youtube.com/watch?v=eYZI7vj5tvE&list=PLH5_eZVldmtV9qLMy4pBYFynNoXl_5YR

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	3	1	3	1	1	2	2	1	2	**1.9**
CO-2	3	3	3	1	3	1	1	2	2	1	2	**2.0**
CO-3	3	2	2	1	3	1	1	1	1	1	2	**1.6**
CO-4	3	2	2	2	2	1	1	1	1	1	2	**1.6**
CO-5	3	2	3	2	3	1	1	1	2	2	2	**2.0**
Average	**3.0**	**2.2**	**2.6**	**1.4**	**2.8**	**1.0**	**1.0**	**1.4**	**1.6**	**1.2**	**2.0**	

Agile Software Development Process

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Software Engineering

Course Objectives:

1. Understand the principles and practices of agile software development.
2. To study and apply various Agile methodologies and practices such as Scrum, Extreme Programming.
3. To learn and implement Agile design principles, including SOLID principles.
4. To understand Agile testing methodologies, including test planning, test automation, Test-Driven development, user acceptance testing, and continuous integration practices.
5. To develop skills in Agile project management.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understand the principles and practices of agile software development.	Understand
CO-2	Apply the principles of Agile Software Development, including the Agile Manifesto, to differentiate Agile methods from traditional software development approaches.	Apply
CO-3	Apply agile testing techniques, project management tasks like scheduling, estimation, monitoring and quality assurance activities.	Apply
CO-4	Analyze and implement various Agile methodologies such as Scrum, Extreme Programming (XP), Lean Development, and Feature driven development in software projects.	Analyze
CO-5	Design software systems using Agile design practices and apply SOLID principles to improve code quality, maintainability, and flexibility.	Create

Course Contents:

Unit	Contents	Lectures required
1	**Basics:** The Genesis of Agile, Introduction and background, Agile Manifesto and Principles, Differences between Agile and traditional plans, Stakeholders, Challenges.	4
2	**Agile Approaches-**Extreme Programming, Agile Process Models: Scrum, Project Phases, Dynamic Systems Development Method, Agile Modeling, Agile Unified Process. A Tool Set for the Agile Process, Feature Driven development, Lean Software Development, Agile Project management, Test Driven Development, Continuous Integration, Refactoring, Pair Programming.	8

3	**Agile Design** practices, The Single-Responsibility Principle, The Open-Closed Principle, The Liskov-Substitution Principle, The Dependency-Inversion Principle, The Interface-Segregation Principle.	**6**
4	**Agile Testing**-Planning and Managing Testing Cycle, Agile Lifecycle and its impact on testing, Principles of Agile Testing, Agile Testing Techniques, xUnit Framework, Test Driven Development, User Acceptance Tests, Test Automation.	**8**
5	**Agile Project Management-** Scheduling in an agile project, scheduling challenges, estimating costs, monitoring.	**2**
Total lectures		**28**

Suggested Text Book(s):

1. Robert C. Martin, Agile Software Development, Principles, Patterns, and Practices , Pearson.
2. James Schiel, Enterprise-Scale Agile Software Development, CRC Press
3. Lisa Crispin, Janet Gregory, "Agile Testing: A Practical Guide for Testers and Agile Teams",
4. Addison Wesley.
5. Roger S. Pressman,Bruce R. Maxim, "Software Engineering", McGraw Hill, 8th edition

Suggested Reference Book(s):

1. Craig Larman, Agile & Iterative development, Pearson.
2. John Hunt, Agile Software construction, Springer.

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/106105182
2. Link to NPTEL course contents: https://www.youtube.com/watch?v=jRs-aFETAXY
3. Link to topics related to course: https://www.coursera.org/learn/agile-development-and-scrum

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1		2	2	2		3	1	3	1	1	3	**1.6**
CO-2		2	2	2		3			1	2	3	**1.3**
CO-3	2		2	2	1	3	2	1	2	1	3	**1.7**
CO-4	2		2	2	1	3	2	1	2	2	3	**1.8**
CO-5	2		2	2	2	3	2	1	2	3	3	**2**
Average	**2**	**2**	**2**	**2.0**	**0.8**	**3.0**	**1.4**	**1.2**	**1.6**	**1.8**	**3**	

Agile Software Development Process Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: Elective

L-T-P: 0-0-2

Pre-requisite: Object oriented Programming (java is preferred)

Course Objectives:

1. To understand and apply agile methodologies such as Scrum and Extreme Programming (XP) in real-world software development projects.
2. To develop proficiency in modern development **tools** including version control systems, automated build tools, and continuous integration frameworks.
3. To apply agile design and refactoring principles for improving software quality, maintainability, and adaptability.
4. To implement agile project management practices such as backlog creation, user story development, estimation, scheduling, and continuous monitoring.
5. To perform effective testing within agile environments using agile testing techniques to ensure reliable and high-quality software delivery.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Identify the roles and responsibilities in agile projects and their difference from traditional method projects.	Apply
CO2	Apply agile design principles to develop software	Apply
CO3	Employ agile testing techniques to develop software	Apply
CO4	Implement agile project management tasks like scheduling, estimation, monitoring and quality assurance activities	Apply
CO5	Develop software by applying agile design principles and approaches.	Apply

List of Experiments

S. No	Description	Hours
1	**Agile Background & Driving Forces:** To implement and analyze real-world scenarios with frequently changing requirements; compare Agile and Waterfall models; and identify key driving forces such as flexibility, customer collaboration, and faster delivery, using discussion, charts, and presentations.	**4**
2	**Backlog Creation & User Stories:** To implement the learning of Agile practices, select a sample project such as an e-commerce application, create user stories using the standard format, define clear acceptance criteria for each story, and organize them into a prioritized product backlog using tools like Taiga or Jira (Scrum-based).	**2**

3	**Automated Build Tool:** Build tools for compile, package, and manage dependencies in software projects via Maven, Gradle (open source) tools.	**2**
4	**Version Control System:** To implement repository management, commits, branching, and merging via Git, GitHub.	**2**
5	**Continuous Integration (CI):** To implementautomated build and testing, configure a CI pipeline, set it to trigger builds on each code commit, and integrate both build and testing steps using Jenkins.	**2**
6	**Design Principles & Refactoring:** Implement SOLID principles for code smell, refactoring and maintainability and flexibility. Useful tools are Visual Studio Code, Any programming language (Java/Python/C++).	**4**
7	**Agile Testing Practices:** Implement unit testing and basic TDD (test-driven development) with the tools **Junit (Java), Pytest (Python), C++ (**Google Test / Catch2).	**4**
8	**Agile Project Management - Scheduling & Monitoring:** Implement scheduling, estimation, progress monitoring, backlog tracking (burn-down), reporting, and project control in an Agile environment.	**4**
9	**Mini Project (Agile Implementation):** Implement small project environment via scrum, XP by considering all necessary tools and each unit methods.	**4**
Total Lab hours		**28**

Suggested/Resources:

Suggested Text Book(s):

1. Robert C. Martin, Agile Software Development, Principles, Patterns, and Practices , Pearson.
2. James Schiel, Enterprise-Scale Agile Software Development, CRC Press
3. Lisa Crispin, Janet Gregory, “Agile Testing: A Practical Guide for Testers and Agile Teams”, Addison Wesley.
4. Roger S. Pressman,Bruce R. Maxim, ”Software Engineering”, McGraw Hill, 8th edition

Suggested Reference Book(s):

1. Craig Larman, Agile & Iterative development, Pearson.
2. John Hunt, Agile Software construction, Springer.

Other useful resource(s):

1. Link to NPTEL course contents:https://nptel.ac.in/courses/106105182
2. Link to NPTEL course contents: https://www.youtube.com/watch?v=jRs-aFETAXY
3. Link to topics related to course: https://www.coursera.org/learn/agile-development-and-scrum

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment	60

	execution, viva voce, quizzes, mini-projects etc.)	
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	3	2	1	2	3	3	1	2	3	**2.1**
CO-2	2	3	3	3	3	3	3	3	1	3	3	**2.7**
CO-3	2	3	3	3	3	3	3	3	1	3	3	**2.7**
CO-4	2	3	1	2	3	3	3	3	1	3	3	**2.4**
CO-5	2	3	3	3	3	3	3	3	2	3	3	**2.8**
Average	**2.0**	**2.8**	**2.6**	**2.6**	**2.6**	**2.8**	**3.0**	**3.0**	**1.2**	**2.8**	**3.0**	

IoT Analytics

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Programming in Python/C++, Basics of Electronics

Course Objectives:

1. To understand the architecture of IoT systems and the role of hardware platforms such as Arduino Uno, ESP32, and Raspberry Pi in data acquisition.
2. To interface and configure diverse sensors — temperature, humidity, pressure, sound, camera, and light — for real-world IoT data collection.
3. To apply data preprocessing, storage, and communication protocols for reliable IoT data pipelines.
4. To implement statistical and machine learning techniques for analysing, visualising, and deriving insights from IoT sensor data.
5. To design end-to-end IoT analytics solutions with edge processing, cloud integration, and real-time dashboard visualisation.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the IoT system architecture and differentiate the capabilities, use-cases, and programming models of Arduino Uno, ESP32, and Raspberry Pi platforms.	Understand
CO-2	Interface and configure multiple sensor types - temperature, humidity, pressure, sound, light, and camera - for IoT data acquisition.	Apply
CO-3	Apply data preprocessing, storage, and communication protocols (MQTT, HTTP) to build reliable IoT data acquisition and transmission pipelines.	Apply
CO-4	Implement statistical analysis, anomaly detection, and machine learning models to extract actionable insights from IoT sensor streams.	Analyse
CO-5	Design and demonstrate an end-to-end IoT analytics system integrating edge processing, cloud connectivity, and real-time data visualisation.	Create

Course Contents:

Unit	Contents	Lectures Required
1	Introduction to IoT Analytics: Definition and scope of IoT analytics — descriptive, diagnostic, predictive, and prescriptive analytics; IoT system layers — sensing, network, data processing, and application; IoT data characteristics — high frequency, heterogeneity, noise, and temporal correlation; Overview of hardware platforms — Arduino Uno, ESP32, and Raspberry Pi: architecture, processing power, memory, connectivity, and use-cases; Selection criteria for hardware platforms and sensors based on	3

	application requirements.	
2	Arduino Uno — ATmega328P architecture, GPIO, ADC/DAC, PWM, I2C, SPI, and UART; Programming with Arduino IDE — sketch structure, libraries, and serial monitor; ESP32 — dual-core architecture, integrated Wi-Fi and Bluetooth, deep-sleep modes, Arduino IDE programming on ESP32; Raspberry Pi — ARM Cortex-A architecture, GPIO, , and Linux-based OS; Python GPIO programming on Raspberry Pi using RPi.GPIO and gpiozero.	4
3	Sensors and Actuators for IoT Analytics: Temperature and humidity sensing —DHT22; ESP32-CAM module (OV2640), ultrasonic sensor (HC-SR04), and IR obstacle sensor; Actuator interfacing — LED, buzzer, relay module, and servo motor;	3
4	IoT Data Acquisition and Communication Protocols: Analog-to-digital conversion (ADC) sampling, resolution, and Nyquist considerations; Serial communication protocols — I2C bus arbitration, SPI full-duplex communication, UART framing; MQTT protocol — broker architecture, topics, QoS levels, and publish-subscribe model; Introduction to IoT platforms — ThingSpeak.	4
5	IoT Data Preprocessing and Storage: Sources of IoT data quality issues — sensor drift, missing readings, noise, and outliers; Techniques for handling missing data — forward-fill, interpolation, and median imputation; Noise filtering — moving average, exponential smoothing, and Butterworth low-pass filter; Feature extraction from time-series sensor data — statistical features (mean, variance, RMS, kurtosis) and sliding window segmentation; Local data storage on Raspberry Pi — CSV logging and SQLite.	5
6	Analytics and Machine Learning for IoT Data: Exploratory data analysis (EDA) — summary statistics, correlation heatmaps, and time-series plots using Pandas and Matplotlib/Seaborn; Statistical methods — Z-score and IQR-based anomaly detection; Threshold-based and rule-based alerting systems; Supervised learning — classification (k-NN, Decision Tree, Random Forest) and regression (Linear Regression, SVR) on sensor data using Scikit-learn; Model evaluation — cross-validation, confusion matrix, RMSE, and MAE; Model persistence and loading using joblib/pickle for edge deployment.	6
7	Edge Analytics and Real-Time Processing: Motivation for edge analytics — latency, bandwidth, and privacy benefits; On-device inference with TensorFlow Lite on Raspberry Pi; Real-time data processing with Python threading and asyncio.	3
Total lectures		**28**

Suggested Text Book(s):

1. Vijay Madisetti, ArshdeepBahga, Internet of Things: A Hands-On Approach, Universities Press, 2nd Edition, 2021.
2. Simon Monk, Programming Arduino: Getting Started with Sketches, McGraw-Hill, 3rd Edition, 2022.
3. Neil Cameron, ESP32 Arduino Projects, Apress, 2021.
4. Simon Monk, Raspberry Pi Cookbook: Software and Hardware Problems and Solutions, O'Reilly Media, 4th Edition, 2023.

Suggested Reference Book(s):

1. Wes McKinney, Python for Data Analysis, O'Reilly Media, 3rd Edition, 2022.

2. AurélienGéron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, 3rd Edition, 2022.
3. Pete Warden, Daniel Situnayake, TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers, O'Reilly Media, 2020.
4. Guy Hart-Davis, Raspberry Pi Projects for the Evil Genius, McGraw-Hill, 2nd Edition, 2017.

Other useful resource(s):

1. NPTEL course — Internet of Things: https://nptel.ac.in/courses/106105166
2. ESP32 official documentation: https://docs.espressif.com/projects/esp-idf/en/latest/
3. Arduino reference and tutorials: https://www.arduino.cc/reference/en/
4. Raspberry Pi documentation: https://www.raspberrypi.com/documentation/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	1.6
CO-2	3	3	3	2	3	1	1	2	2	2	2	2.2
CO-3	3	3	2	2	3	1	1	1	2	2	2	2.0
CO-4	3	3	3	2	3	1	1	2	3	2	2	2.3
CO-5	3	3	3	3	3	2	2	2	3	2	2	2.5
Average	**3.0**	**2.8**	**2.6**	**2.0**	**2.8**	**1.2**	**1.2**	**1.6**	**2.4**	**1.8**	**2.0**	

IoT Analytics Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Programming in Python/C++, Basics of Electronics

Course Objectives:

1. To set up and program IoT hardware platforms - Arduino Uno, ESP32, and Raspberry Pi — for sensor-based data acquisition experiments.
2. To interface and experimentally validate multiple sensors including temperature, humidity, pressure, sound, light, and camera modules.
3. To implement IoT communication protocols and transmit sensor data to cloud platforms and local dashboards.
4. To apply data preprocessing, statistical analysis, and machine learning techniques on collected IoT sensor data using Python.
5. To design and demonstrate a complete IoT analytics pipeline integrating hardware sensing, data communication, analysis, and visualisation.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Set up development environments and write programs for GPIO, sensor interfacing, and serial communication on Arduino Uno, ESP32, and Raspberry Pi.	Apply
CO-2	Interface and validate the operation of temperature, humidity, pressure, sound, light, and camera sensors on appropriate IoT platforms.	Apply
CO-3	Implement MQTT and HTTP-based IoT data pipelines to transmit sensor readings to cloud platforms and local brokers.	Apply
CO-4	Apply data preprocessing and analysis techniques — cleaning, feature extraction, anomaly detection, and ML-based classification on real sensor datasets.	Analyse
CO-5	Build and demonstrate an integrated, multi-sensor IoT analytics system with real-time data visualisation and automated response logic.	Create

List of Experiments:

S. No	Description	Hours
1	Platform Setup and Blink-to-Buzz: Installation of Arduino IDE, Thonny/MicroPython for ESP32, and Raspberry Pi OS setup. Write and upload programs to blink an LED and activate a buzzer using digital GPIO on Arduino Uno and ESP32. Explore serial monitor for debugging. Verify GPIO, PWM, and	**2**

	timer configurations.	
2	GPIO and Analog Interfacing on Arduino Uno: Interfacing a push button (digital input), LDR (analog input using ADC), and LED (digital output) with Arduino Uno. Read and display analog light levels on the serial monitor. Implement threshold-based LED control responding to ambient light.	**2**
3	Temperature and Humidity Sensing (DHT11/DHT22): Interface DHT11 or DHT22 sensor with Arduino Uno and ESP32. Read temperature and relative humidity values, display them on the serial console, and log timestamped readings to a CSV file via serial. Observe effects of breath and heat source on readings.	**2**
4	Pressure, Altitude, and Environmental Sensing (BMP280/BME680): Interface the BMP280 sensor over I2C with ESP32. Read barometric pressure, temperature, and estimated altitude. Plot the variation of readings with height change (lifting sensor vertically). Use the BME680 (if available) to additionally measure gas/VOC resistance.	**2**
5	Sound and Light Intensity Sensing: Interface a sound sensor module and a BH1750 digital lux sensor with ESP32. Measure ambient sound levels and lux intensity values. Implement event detection — trigger an LED when sound exceeds a threshold. Log both sensor readings simultaneously over time.	**2**
6	Camera Interfacing and Image Capture: Configure the ESP32-CAM module (OV2640) to capture still images and stream MJPEG video over Wi-Fi. Alternatively, use the Raspberry Pi Camera Module v2 to capture images using the picamera2 library in Python. Perform basic image processing operations (grayscale conversion, edge detection) using OpenCV.	**2**
7	MQTT Communication with ESP32: Set up a local MQTT broker (Mosquitto) on Raspberry Pi. Program the ESP32 to publish temperature and humidity data from DHT22 to MQTT topics at fixed intervals. Subscribe to topics from Raspberry Pi using Python (paho-mqtt) and display incoming data on the terminal.	**2**
8	IoT Cloud Integration — ThingSpeak and AWS IoT Core: Publish multi-sensor data (temperature, humidity, lux) from ESP32 to ThingSpeak channels using HTTP GET API. Visualise live data on ThingSpeak dashboards and set up a ThingSpeak React alert. Optionally, connect ESP32 to AWS IoT Core using MQTT with X.509 certificate authentication.	**2**
9	IoT Data Preprocessing and Storage on Raspberry Pi: Collect multi-sensor data (DHT22, BMP280, LDR) from ESP32 via MQTT into a Raspberry Pi Python script. Store data in a CSV file and query it using Pandas. Perform preprocessing — handle missing values, apply moving average smoothing, and visualise time-series trends using Matplotlib and Seaborn.	**2**
10	Anomaly Detection on Sensor Data: Load a pre-collected IoT sensor dataset (temperature, humidity, pressure). Apply Z-score and IQR-based statistical anomaly detection methods to flag abnormal readings. Visualise detected anomalies on time-series plots. Implement a real-time threshold-based alerting mechanism on ESP32 that triggers a buzzer on anomaly.	**2**

11	Machine Learning on IoT Sensor Data: Using Scikit-learn on Raspberry Pi, train a classification model (Decision Tree or Random Forest) on labelled sensor data (e.g., normal vs. fault conditions). Evaluate the model using confusion matrix and F1-score. Save the trained model using joblib and load it for inference on new sensor readings.	**2**
12	Design and demonstrate a complete IoT analytics pipeline: (i) Acquire data from at least three sensors (e.g., DHT22, BMP280, and sound/LDR) using ESP32 ; (ii) Transmit data to Raspberry Pi via MQTT; (iii) Preprocess and visualise data using Python; (iv) Apply a trained ML model for real-time classification; (v) Build a Node-RED dashboard showing live sensor values, alerts, and trends.	**2**
13	Mini Project: End-to-End Multi-Sensor IoT Analytics System.	**4**
Total Lab Hours		**28**

Suggested / Resources:

1. Vijay Madisetti, ArshdeepBahga, Internet of Things: A Hands-On Approach, Universities Press, 2nd Edition, 2021.
2. Simon Monk, Programming Arduino: Getting Started with Sketches, McGraw-Hill, 3rd Edition, 2022.
3. Neil Cameron, ESP32 Arduino Projects, Apress, 2021.
4. Simon Monk, Raspberry Pi Cookbook: Software and Hardware Problems and Solutions, O'Reilly Media, 4th Edition, 2023.
5. Pete Warden, Daniel Situnayake, TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers, O'Reilly Media, 2020.
6. Wes McKinney, Python for Data Analysis, O'Reilly Media, 3rd Edition, 2022.
7. AurélienGéron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, 3rd Edition, 2022.

Other useful resource(s):

1. Arduino reference and tutorials: https://www.arduino.cc/reference/en/
2. ESP32 official documentation: https://docs.espressif.com/projects/esp-idf/en/latest/
3. Raspberry Pi documentation: https://www.raspberrypi.com/documentation/
4. Node-RED documentation: https://nodered.org/docs/

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	3	1	3	1	1	2	2	1	2	1.9
CO-2	3	3	3	2	3	1	1	2	2	2	2	2.2
CO-3	3	3	2	2	3	1	1	1	2	2	2	2.0
CO-4	3	3	3	2	3	1	1	2	3	2	2	2.3
CO-5	3	3	3	3	3	2	2	2	3	2	2	2.5
Average	**3.0**	**2.8**	**2.8**	**2.0**	**3.0**	**1.2**	**1.2**	**1.8**	**2.4**	**1.8**	**2.0**	

Image Processing and Computer Vision

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Linear Algebra, Probability, Machine Learning and Deep learning

Course Objectives:

1. To understand the fundamentals of digital image processing and computer vision.
2. To learn image representation, enhancement, and transformation techniques.
3. To apply feature extraction and segmentation techniques for image analysis.
4. To understand machine learning and deep learning approaches in computer vision.
5. To design and implement real-world vision-based applications.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain fundamental concepts of image processing and computer vision	Understand
CO-2	Apply image enhancement, filtering, and transformation techniques.	Apply
CO-3	Analyze images using segmentation and feature extraction methods	Apply
CO-4	Apply machine learning and deep learning techniques for vision tasks.	Analyze
CO-5	Design and implement real-world computer vision applications.	Create

Course Contents:

Unit	Contents	Lectures required
1	**Introduction and Image Formation:** Introduction and goals of computer vision and image processing; applications in autonomous driving, medical imaging, and robotics; image formation concepts; radiometry basics; geometric transformations; geometric camera models: pinhole model; sampling and quantization; camera calibration; image formation in stereo vision.	**6**

2	**Image Processing Techniques:** Image representation: spatial and intensity resolution; color models: RGB, HSV; intensity transformations and histogram equalization; noise models: Gaussian, salt-and-pepper noise; spatial filtering: Gaussian, median, bilateral; edge detection: Sobel; multi-scale representation image pyramids Gaussian and Laplacian; basics of Image restoration noise removal, de-blurring, and image degradation model.	**7**
3	**Image Segmentation and Feature Extraction:** Image segmentation techniques: thresholding (Otsu's method), region-based methods, clustering-based segmentation, watershed algorithm; segmentation evaluation metrics: IoU, Dice coefficient; feature detection: edges, Harris corner detector; feature descriptors: SIFT, SURF, ORB, HOG; feature matching and outlier removal: feature matching, RANSAC.	**5**
4	**Machine Learning for Computer Vision:** Supervised and unsupervised learning; linear and logistic regression; K-nearest neighbors (KNN); support vector machines (SVM); dimensionality reduction (PCA, LDA)	**5**
5	**Deep Learning for Computer Vision:** Neural network fundamentals; convolutional neural networks (CNNs); architectures (LeNet, AlexNet, VGG, ResNet); image classification; introduction to object detection (R-CNN, Fast R-CNN, YOLO); semantic segmentation (FCN, U-Net basics); transfer learning and fine-tuning.	**5**
Total lectures		**28**

Suggested Text Book(s):

1. Computer Vision and Image Processing: Fundamentals and Applications, "Computer Vision and Image Processing: Fundamentals and Applications", CRC Press, USA.
2. Computer Vision: A Modern Approach, "Computer Vision - A Modern Approach", Pearson Education.
3. Computer Vision: Algorithms and Applications, "Computer Vision - Algorithms & Applications", Springer.

Suggested Reference Book(s):

1. Digital Image Processing, "Digital Image Processing", Rafael C. Gonzalez and Richard E. Woods, Pearson.
2. Deep Learning, "Deep Learning", Ian Goodfellow, YoshuaBengio, and Aaron Courville, MIT Press.
3. Learning OpenCV, "Learning OpenCV", Gary Bradski and Adrian Kaehler, O'Reilly Media.

Other useful resource(s):

4. Link to NPTEL course contents: https://nptel.ac.in/courses/108103174
5. Link to NPTEL course contents: https://nptel.ac.in/courses/117105135

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination

1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	2	1	1	1	1	1	2	**1.4**
CO-2	3	3	2	2	3	1	1	1	1	1	2	**1.8**
CO-3	3	3	3	2	3	1	1	1	1	1	2	**1.9**
CO-4	3	3	2	2	3	1	1	1	1	1	3	**1.9**
CO-5	3	2	3	2	3	1	1	2	2	2	3	**2.1**
Average	**3**	**2.6**	**2.2**	**1.8**	**2.8**	**1**	**1**	**1.2**	**1.2**	**1.2**	**2.4**	

Image Processing and Computer Vision Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Programming (Python/C++), and fundamentals of Linear Algebra and Probability.

Course Objectives:

1. To introduce the fundamental concepts of digital image processing and computer vision.
2. To provide knowledge of image representation, enhancement, and transformation techniques.
3. To develop understanding of image segmentation and feature extraction methods for visual data analysis.
4. To familiarize students with machine learning techniques applied to computer vision problems.
5. To expose students to deep learning approaches for advanced vision tasks such as image classification, object detection, and segmentation.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Understand the fundamentals of digital image processing including image representation, sampling, quantization, and geometric transformations.	Apply
CO2	Analyze and apply image enhancement and filtering techniques for improving image quality and noise removal.	Apply
CO3	Analyze image data using segmentation and feature extraction techniques for meaningful interpretation.	Apply
CO4	Apply machine learning algorithms for image classification and evaluate model performance using appropriate metrics.	Apply
CO5	Design and implement deep learning models for computer vision tasks such as image classification, object detection, and segmentation.	Apply

List of Experiments

S. No	Description	Hours
1	Write a program to study image representation, sampling, and quantization of images.	2
2	Write a program to perform geometric transformations such as translation, rotation, and scaling on images.	2
3	Write a program to implement image enhancement techniques including histogram equalization and intensity transformations.	2

4	Write a program to add noise (Gaussian and salt & pepper) to images and perform noise removal using spatial filters such as Gaussian, median, and bilateral filters.	2
5	Write a program to perform edge detection using Sobel, Laplacian, and Canny operators.	2
6	Write a program to implement frequency domain filtering using Fourier Transform, including low-pass and high-pass filters.	2
7	Write a program to implement image segmentation techniques such as Otsu thresholding, K-means clustering, and watershed algorithm.	2
8	Write a program to perform feature extraction using methods such as Harris corner detection, SIFT/ORB descriptors, and HOG features	2
9	Write a program to implement image classification using K-Nearest Neighbors (KNN) with dimensionality reduction using PCA, and evaluate the results using precision, recall, F1-score, and IoU metrics.	2
10	Write a program to implement image classification using Support Vector Machine (SVM) with dimensionality reduction using PCA, and evaluate the performance using precision, recall, F1-score, and IoU metrics.	2
11	Write a program to implement CNN-based image classification for computer vision tasks.	2
12	Write a program to perform image classification using transfer learning with pre-trained deep learning models.	2
13	Write a program to implement basic object detection using YOLO or semantic segmentation using U-Net for computer vision applications.	4
Total Lab hours		**28**

Suggested Text Book(s):

1. Computer Vision and Image Processing: Fundamentals and Applications, “Computer Vision and Image Processing: Fundamentals and Applications”, CRC Press, USA.
2. Computer Vision: A Modern Approach, “Computer Vision - A Modern Approach”, Pearson Education.
3. Computer Vision: Algorithms and Applications, “Computer Vision - Algorithms & Applications”, Springer.

Suggested Reference Book(s):

1. Digital Image Processing, “Digital Image Processing”, Rafael C. Gonzalez and Richard E. Woods, Pearson.
2. Deep Learning, “Deep Learning”, Ian Goodfellow, YoshuaBengio, and Aaron Courville, MIT Press.
3. Learning OpenCV, “Learning OpenCV”, Gary Bradski and Adrian Kaehler, O’Reilly Media.

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/108103174
2. Link to NPTEL course contents: https://nptel.ac.in/courses/117105135

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	3	1	1	1	2	1	2	**1.7**
CO-2	3	3	2	2	3	1	1	1	2	1	2	**1.9**
CO-3	3	3	3	2	3	1	1	1	1	1	2	**1.9**
CO-4	3	3	2	2	3	1	1	1	1	1	2	**1.8**
CO-5	3	2	3	2	3	1	1	2	2	2	3	**2.1**
Average	**3**	**2.6**	**2.4**	**1.8**	**3**	**1**	**1**	**1.2**	**1.6**	**1.2**	**2.2**	

Introduction to Blockchain Technology

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Basics of Computer Networks, Fundamentals of Programming

Course Objectives:

1. To understand the fundamental concepts of blockchain technology, including its history, architecture, and types.
2. To explain cryptographic primitives (hashing, digital signatures) and their role in ensuring blockchain security.
3. To analyze the working of consensus algorithms (PoW, PoS, etc.) and their impact on sustainability and decentralization.
4. To explore smart contracts, decentralized applications (dApps), and blockchain platforms like Ethereum.
5. To evaluate real-world applications of blockchain beyond cryptocurrencies (e.g., supply chain, healthcare, voting).

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the fundamental concepts, architecture, and types of blockchain (public, private, consortium).	Understand
CO-2	Apply cryptographic techniques (hash functions, digital signatures) to ensure data integrity and non-repudiation in blockchain.	Apply
CO-3	Analyze different consensus mechanisms (Proof of Work, Proof of Stake, etc.) for their strengths, weaknesses, and energy implications.	Analyze
CO-4	Design basic smart contracts and explain the working of decentralized applications (dApps) on platforms like Ethereum.	Create
CO-5	Evaluate the societal, legal, and environmental impact of blockchain adoption across various industry sectors.	Evaluate

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to Blockchain:** Evolution of digital trust, centralized vs. decentralized systems, history of blockchain (Bitcoin whitepaper), basic terminology (blocks, chain, nodes, miners), types of blockchain (public, private, consortium, hybrid).	4
2	**Cryptographic Foundations:** Hash functions (SHA-256, properties), Merkle trees, digital signatures (ECDSA), public-private key pairs, wallets and addresses, immutability and data integrity.	5

3	**Blockchain Architecture & Transactions:** Structure of a block (header, body), block hash, previous block hash, nonce, transaction lifecycle (creation, broadcast, validation), UTXO vs. account-based models.	**4**
4	**Consensus Mechanisms:** Need for consensus, Byzantine Generals' Problem, Proof of Work (PoW) – mining, difficulty, energy concerns, Proof of Stake (PoS), Delegated PoS, Practical Byzantine Fault Tolerance (PBFT), and other eco-friendly alternatives.	**5**
5	**Smart Contracts & Ethereum:** Introduction to Ethereum, Ether and gas, structure of a smart contract, Solidity basics (simple contract deployment), Decentralized Applications (dApps), oracles.	**5**
6	**Blockchain Use Cases & Challenges:** Beyond cryptocurrencies – supply chain management, digital identity, healthcare records, voting systems, intellectual property. Scalability, privacy (zero-knowledge proofs), interoperability, legal and regulatory aspects.	**5**
Total lectures		**28**

Suggested Text Book(s):

1. ArshdeepBahga, Vijay Madisetti, *Blockchain Applications: A Hands-On Approach*, VPT, 2017.
2. Imran Bashir, *Mastering Blockchain*, Packt Publishing, 3rd Edition, 2020.
3. Andreas M. Antonopoulos, *Mastering Bitcoin*, O'Reilly Media, 2nd Edition, 2017.

Suggested Reference Book(s):

1. Daniel Drescher, Blockchain Basics: A Non-Technical Introduction in 25 Steps, Apress, 2017.
2. William Mougayar, *The Business Blockchain*, Wiley, 2016.

Other useful resource(s):

1. NPTEL: Blockchain Architecture Design and Use Cases – https://nptel.ac.in/courses/106105235
2. MIT OpenCourseWare: Blockchain and Money – https://ocw.mit.edu (15.S12)
3. Solidity Documentation – https://docs.soliditylang.org

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	2	2	1	1	2	1	2	**1.6**
CO-2	3	3	2	2	3	1	2	1	1	1	2	**1.9**
CO-3	2	3	2	3	2	3	1	1	1	1	3	**2.0**
CO-4	3	2	3	2	3	2	1	2	2	2	3	**2.3**
CO-5	2	3	2	2	1	3	3	2	3	3	3	**2.5**
Average	**2.6**	**2.6**	**2.0**	**2.0**	**2.2**	**2.2**	**1.6**	**1.4**	**1.8**	**1.6**	**2.6**	

Introduction To Blockchain Technology Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Basics of Computer Networks, Fundamentals of Programming, Linux/Command Line Basics

Course Objectives:

1. To install and configure blockchain development environments (Ganache, Truffle, MetaMask).
2. To implement cryptographic primitives (hashing, digital signatures) using Python/JavaScript libraries.
3. To simulate and analyze different consensus mechanisms (PoW, PoS) through hands-on coding.
4. To develop, deploy, and test smart contracts using Solidity and Ethereum platforms.
5. To build a basic decentralized application (dApp) and evaluate blockchain use cases.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Install and configure blockchain development tools and create a local blockchain network.	Apply
CO2	Implement cryptographic hash functions and digital signatures for data integrity and authentication.	Apply
CO3	Simulate and compare consensus algorithms (Proof of Work, Proof of Stake) through programming.	Analyze
CO4	Write, deploy, and test smart contracts on Ethereum-compatible blockchain using Solidity and Remix IDE.	Create
CO5	Develop a simple decentralized application (dApp) integrating smart contract with a front-end interface.	Create

List of Experiments

S. No	Description	Hours
1	Installation and setup of blockchain development environment: Node.js, npm, Ganache (personal blockchain), Truffle framework, MetaMask browser extension. Creating a local blockchain network with multiple test accounts.	**2**
2	Implementation of cryptographic hash functions (SHA-256) using Python/JavaScript. Understanding properties: deterministic, pre-image resistance, avalanche effect. Hashing of text, files, and blocks.	**2**
3	Implementation of digital signatures using ECDSA (Elliptic Curve Digital Signature Algorithm): key pair generation, signing a message, and signature verification. Simulating transaction authentication.	**2**

4	Building a simple blockchain in Python/JavaScript: Block structure (index, timestamp, data, previous hash, nonce, current hash), proof-of-work, chain validation, and tamper detection.	**2**
5	Simulation of Proof of Work (PoW) consensus: Mining with difficulty adjustment, computing nonce, measuring mining time and energy cost. Comparison with Proof of Stake (PoS) simulation (validator selection based on stake).	**2**
6	Design public-private key pairs and create Ethereum addresses, send signed transactions using Web3.js library. Also check balances on Ganache.	**2**
7	Write a basic Solidity smart contract (storage, counter, simple bank). Compile and deploy to local Ganache blockchain, and interact using Remix interface.	**2**
8	Implement Smart contract development – Variables, data types, functions, modifiers, and events in Solidity. Deploy a contract for transferring Ether between accounts.	**2**
9	Implement a simple token contract (ERC-20 like) with minting and transferring functions.	**2**
10	Interact with smart contracts using Web3.js: Connect to Ganache, read contract data, send transactions from different accounts, and listen to events.	**2**
11	Developing a decentralized application (dApp) – Part 1: Setting up project structure (HTML/CSS/JS front-end), connecting MetaMask, displaying account and network details.	**4**
12	Developing a decentralized application (dApp) – Part 2: Integrating a deployed smart contract with front-end, calling read/write functions, handling transaction confirmations.	**4**
Total Lab hours		**28**

Suggested/Resources:

3. Andreas M. Antonopoulos, Mastering Ethereum, O'Reilly Media, 2018.
4. Imran Bashir, Mastering Blockchain, Packt Publishing, 3rd Edition, 2020.
5. Dapp University (YouTube & GitHub) – Smart contract and dApp tutorials.
6. Solidity Documentation: https://docs.soliditylang.org
7. Web3.js Documentation: https://web3js.readthedocs.io
8. Truffle Suite Documentation: https://trufflesuite.com/docs
9. Remix IDE: https://remix.ethereum.org
10. CryptoZombies (Interactive Solidity tutorial): https://cryptozombies.io

Other useful resource(s):

1. NPTEL: Blockchain Architecture Design and Use Cases – https://nptel.ac.in/courses/106105235
2. MetaMask Documentation: https://docs.metamask.io
3. Etherscan (Blockchain Explorer): https://etherscan.io

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20

3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	1	1	3	1	1	2	2	1	2	**1.6**
CO-2	3	2	2	2	3	1	2	1	1	1	2	**1.8**
CO-3	2	3	2	3	2	2	1	1	1	1	3	**1.9**
CO-4	2	2	3	2	3	1	1	2	2	2	3	**2.1**
CO-5	2	2	3	2	3	2	2	3	3	2	3	**2.5**
Average	**2.2**	**2.2**	**2.2**	**2.0**	**2.8**	**1.4**	**1.4**	**1.8**	**1.8**	**1.4**	**2.6**	

Computing for Data Science

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Python Programming

Course Objectives:

1. To understand the concept of Data Science Process, Types & Levels of data, Data Munging, Feature Engineering.
2. To learn Statistical Methods in Data Science.
3. To implement Technologies & Tools in Database Analytics.
4. To analyze and debug code with Computing techniques for Data Science.
5. To analyze and evaluate the Statistical Methods for Evaluation and Exploratory Data Analysis & Data Science Process-.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Make use of basic concepts, methods, and mathematics relevant to computational techniques for data science	Apply
CO-2	Develop own statistical analyses and implement them with advanced statistical programming tools	Apply
CO-3	Develop and apply advanced and associated computing techniques and technologies.	Apply
CO-4	Compare the performance of multiple methods and models, recognize the connections between how the data were collected and the scope of conclusions from the resulting analysis, and articulate the limitations and abuses of formal inference and modeling.	Analyze
CO-5	Evaluate strategies for constructing models and can use different measures of model fit and performance to assess models.	Evaluate

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Data Science- Characteristics & Evolution of data, Data Science Process, Types & Levels of data, Datafication, Steps of Data Science, Central Tendency, Measure of Dispersion, Data Munging, Feature Engineering	**4**
2	Statistical Methods in Data Science- Data Distribution (Bernoulli, Uniform, Binomial, Normal, Poisson, Exponential), Mathematical Statistics, Inferential Statistics, Descriptive Statistics, Random Variable, Probabilistic Statistics, Sampling of data, Correlation Analysis	**5**

3	Technologies & Tools in Database Analytics- SQL Essentials for data science, String Pattern, Ranges, Sorting & Grouping Result Set, working with multiple tables, accessing database using R/Python, Database Text Analysis, User defined Functions & Aggregates, MADlib, Tools & Techniques for unstructured data.	**6**
4	Computing techniques for Data Science- Regression, Mapping Problem to Machine Learning Task, Memorization Method, Generalized Additive Models, Time-Series Model, Predictive Modeling, Ensemble Techniques, Outlier Detection, Recommendation Engine	**9**
5	Statistical Methods for Evaluation- Hypothesis Testing, Difference of Means, Significance Level and P-Value, Test Statistics (Z-test, ANOVA, T-Test, Redundancy Test), Bias Variance Trade off, Cross Validation	**2**
6	Exploratory Data Analysis & Data Science Process- Visualization before analysis, Dirty Data, Visualizing single and multiple variables, summary statistics of EDA, Data Exploration versus Presentation, Real time case study, Tools & Techniques	**2**
Total lectures		**28**

Suggested Text Book(s):

1. Haider, M. (2015). Getting Started with Data Science: Making Sense of Data with Analytics. IBM Press.
2. Dietrich, D. (Ed.). (2015). Data science & big data analytics: discovering, analyzing, visualizing and presenting data. Wiley.
3. Trevor, H., Robert, T., & JH, F. (2009). The elements of statistical learning: data mining, inference, and prediction.

Suggested Reference Book(s):

1. Grus, J. (2015). Data science from scratch: First principles with Python. " O'Reilly Media, Inc.".
2. Taylor, J. K., &Cihon, C. (2004). Statistical techniques for data analysis. Chapman and Hall/CRC.
3. Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.
4. Zumel, N., & Mount, J. (2014). Practical data science with R. Manning Publications Co..
5. Saltz, J. S., & Stanton, J. M. (2017). An introduction to data science. SAGE Publications.

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus

4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	3	2	1	1	1	3	2	1	**1.8**
CO-2	3	3	3	3	3	1	1	2	2	2	1	**2.1**
CO-3	2	2	2	2	1	1	2	2	2	2	2	**1.8**
CO-4	3	3	3	3	2	1	1	3	2	3	2	**2.3**
CO-5	3	2	3	2	1	1	2	1	1	3	1	**1.8**
Average	**2.6**	**2.4**	**2.6**	**2.6**	**1.8**	**1.0**	**1.6**	**1.8**	**2.0**	**2.4**	**1.6**	

Computing for Data Science Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Python Programming

Course Objectives:

1. To implement data science workflows using Python/R.
2. To perform statistical analysis on real-world datasets.
3. To work with structured and unstructured data.
4. To apply machine learning models and evaluate them.
5. To visualize and interpret data effectively.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Apply data preprocessing and feature engineering techniques	Apply
CO2	Perform statistical analysis using programming tools.	Apply
CO3	Use SQL and database tools for data analytics	Analyze
CO4	Implement machine learning models	Apply
CO5	Evaluate models using statistical metrics and validation methods	Evaluate

List of Experiments

S. No	Description	Hours
1	Get acquainted with data science tools and perform statistical analysis	**2**
2	Hypothesis tests (e.g., t-tests, Z-tests) on sample datasets to compare population means	**2**
3	Simulate and apply the central limit theorem (CLT) to demonstrate how sample distributions converge to a normal distribution	**2**
4	Perform data wrangling and ETL processes on a dataset, followed by exploratory data analysis (EDA)	**2**
5	Utilize tools to create effective data visualizations (e.g., line charts, bar charts, heat maps, box plots) to derive key insights from the dataset	**4**
6	Implement feature extraction and selection techniques, including experimenting with encoding methods like one-hot encoding and creating new features based on domain expertise	**4**

7	Develop a simple linear regression model, extend it to multiple linear regression with several variables, and visualize both the regression line and residual plots	**4**
8	Develop a simple linear regression model, extend it to multiple linear regression with several variables, and visualize both the regression line and residual plots	**4**
9	Apply K-means clustering and assess cluster quality using evaluation metrics like the silhouette score	**4**
Total Lab hours		**28**

Suggested Text Book(s):

1. Haider, M. (2015). Getting Started with Data Science: Making Sense of Data with Analytics. IBM Press.
2. Dietrich, D. (Ed.). (2015). Data science & big data analytics: discovering, analyzing, visualizing and presenting data. Wiley.
3. Trevor, H., Robert, T., & JH, F. (2009). The elements of statistical learning: data mining, inference, and prediction.

Suggested Reference Book(s):

1. Grus, J. (2015). Data science from scratch: First principles with Python. " O'Reilly Media, Inc.".
2. Taylor, J. K., &Cihon, C. (2004). Statistical techniques for data analysis. Chapman and Hall/CRC.
3. Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.
4. Zumel, N., & Mount, J. (2014). Practical data science with R. Manning Publications Co..
5. Saltz, J. S., & Stanton, J. M. (2017). An introduction to data science. SAGE Publications.

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	3	2	1	1	1	3	2	1	**1.8**
CO-2	3	3	3	3	3	1	1	2	2	2	1	**2.1**
CO-3	2	2	2	2	1	1	2	2	2	2	2	**1.8**
CO-4	3	3	3	3	2	1	1	3	2	3	2	**2.3**
CO-5	3	2	3	2	1	1	2	1	1	3	1	**1.8**
Average	**2.6**	**2.4**	**2.6**	**2.6**	**1.8**	**1.0**	**1.6**	**1.8**	**2.0**	**2.4**	**1.6**	

Sensor Technology and Android Programming

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: Elective

L-T-P: 2-0-0

Pre-requisite: Basic Programming, Mobile App Development (preferred)

Course Objectives:

1. To understand different types of sensors and their working principles.
2. To analyze sensor data and integrate it into mobile applications.
3. To develop context-aware Android applications using sensor APIs.
4. To explore external sensing technologies like NFC, RFID, and camera-based systems.
5. To design intelligent mobile services using sensor data.

Course Outcomes:

S. No.	Course Outcomes	Level
CO-1	Explain different types of sensors and smart sensor systems	Understand
CO-2	Analyze sensor data and the working of sensor-based systems	Analyze
CO-3	Apply Android Sensor APIs to develop context-aware applications	Apply
CO-4	Evaluate external sensing technologies such as NFC, RFID, and camera-based systems	Evaluate
CO-5	Design intelligent mobile applications using multiple sensors	Create

Course Contents:

Unit	Contents	Lectures
1	Fundamentals of Sensors and Smart Systems: Sensing concepts, sensor characteristics, sensing modalities; Mechanical, MEMS, Optical, Semiconductor, Electrochemical, Biosensors (overview only); Smart sensors, sensor systems, microcontrollers (Arduino overview), sensor data acquisition basics	6

2	Android for Sensor-Based Applications (Advanced): Quick recap of Android components (Activity, Manifest – brief), Android architecture for sensor apps, permissions, background services, introduction to sensor framework (avoid UI basics), handling real-time data	4
3	Android Sensor API and Physical Sensing: Types of sensors (accelerometer, gyroscope, proximity, light, pressure), SensorManager, sensor events, motion detection, orientation sensing, environmental sensing, data filtering basics	6
4	External and Augmented Sensing Systems: RFID, NFC basics, NFC-based applications (payments, tagging), camera as sensor (barcode scanning, QR code), introduction to image processing concepts, Android Open Accessory (AOA) overview and limitations	6
5	Sensor-Based Intelligent Services: Context-aware applications, motion detection apps, air quality monitoring (concept), smart brightness control, activity recognition, location + sensor fusion basics, energy-efficient sensing	6
	Total Lectures	**28**

Suggested Text Books:

1. Krzysztof Iniewski, *Smart Sensors for Industrial Applications*
2. Bill Phillips et al., *Android Programming: The Big Nerd Ranch Guide*

Suggested Reference Books:

1. Mike Banzi, *Getting Started with Arduino*
2. Android Developer Documentation

Other Resources:

1. YouTube

a) Android Sensors Tutorial (Accelerometer, Gyroscope, SensorManager)
b) Android App Development Full Course

2. Coursera

a) Android App Development Specialization (Vanderbilt University)
b) Introduction to the Internet of Things and Embedded Systems (University of California, Irvine)

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered up to T-1
2	T-2	25	1.5 Hours	Syllabus covered up to T-2
3	T-3	35	2 Hours	Entire Syllabus

4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) -10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	2	2	1	1	1	1	2	1.4
CO-2	3	3	2	3	2	2	1	1	1	1	2	1.8
CO-3	3	3	3	2	3	2	1	1	2	1	2	2
CO-4	2	3	2	2	3	2	1	1	2	2	2	2
CO-5	3	2	3	2	3	3	1	2	2	2	3	2.3
Average	2.8	2.6	2.2	2	2.6	2.2	1	1.2	1.6	1.4	2.2	

Sensor Technology and Android Programming Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

L-T-P: 0-0-2

Pre-requisite: Basic Programming, Mobile App Development (preferred)

Course Objectives:

1. To provide hands-on experience with Android sensor APIs.
2. To develop applications using mobile and external sensors.
3. To analyze real-time sensor data.
4. To build context-aware and intelligent applications.
5. To integrate sensing technologies into mobile apps.

Course Outcomes:

S. No.	Course Outcomes	Level
CO-1	Implement applications using Android sensor APIs	Apply
CO-2	Analyze and process sensor data from mobile devices	Analyze
CO-3	Develop applications using motion and environmental sensors	Apply
CO-4	Implement applications using NFC, camera, and external sensing	Apply
CO-5	Design a sensor-based intelligent mobile application	Create

List of Experiments:

S. No	Description	Hours
1	Study and implement listing of available sensors in an Android device using SensorManager	2
2	Develop an application to read and display accelerometer sensor data in real time	2
3	Implement an application for motion detection using accelerometer sensor data	2
4	Develop an application using gyroscope sensor for device orientation detection	2
5	Implement an application using proximity and light sensors for smart control (auto brightness concept)	2

6	Develop an application using pressure/barometer sensor for environmental sensing (or simulation if unavailable)	2
7	Implement sensor data logging and visualization using graphical representation	2
8	Develop an application using GPS and sensor fusion for location and motion tracking	2
9	Implement a step counter / activity recognition application using motion sensors	2
10	Develop an NFC-based application for reading NFC tags (or simulation if hardware not available)	2
11	Implement a camera-based application for QR code/barcode scanning	2
12	Implement integration of external sensors (Arduino + temperature sensor) with Android (demo/simulation)	2
13	Implement a context-aware service	2
14	Design a sensor-based intelligent application	2
Total Lab Hours		**28**

Suggested Text Books:

1. Krzysztof Iniewski, *Smart Sensors for Industrial Applications*
2. Bill Phillips et al., *Android Programming: The Big Nerd Ranch Guide*

Suggested Reference Books:

1. Mike Banzi, *Getting Started with Arduino*
2. Android Developer Documentation

Other Resources:

1. NFC Forum – **https://nfc-forum.org**
2. ML Kit – **https://developers.google.com/ml-kit**
3. Arduino – **https://www.arduino.cc/en/Tutorial/HomePage**
4. MIT App Inventor – https://appinventor.mit.edu

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20

3	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects, etc.)	60
	Total	100

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	3	2	1	1	1	1	2	1.7
CO-2	3	3	2	3	3	2	1	1	1	1	2	1.9
CO-3	3	2	3	2	3	2	1	2	2	1	2	2
CO-4	3	2	3	2	3	2	1	2	2	1	2	2
CO-5	3	3	3	3	3	2	1	2	2	2	3	2.4
Average	3	2.4	2.6	2.4	3	2	1	1.6	1.6	1.2	2.2	

Introduction to Deep Learning

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Machine Learning, Linear Regression, Probability Theory

Course Objectives:

1. To build a foundational understanding of neural network architectures, activation functions, and feedforward computation.
2. To develop proficiency in training deep networks using backpropagation, gradient-based optimizers, and regularization strategies.
3. To study convolutional and recurrent architectures for image and sequence modelling tasks, along with pre-trained model ecosystems.
4. To introduce the attention mechanism and Transformer-based models including BERT and GPT for language understanding and generation.
5. To expose learners to advanced generative models, post-Transformer architectures, and the latest trends in Generative AI.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the fundamental concepts of artificial neural networks	Understand
CO-2	Apply gradient descent and backpropagation while analyzing convergence and overfitting.	Apply, Analyze
CO-3	Design CNNs with transfer learning for vision tasks.	Apply, Evaluate
CO-4	Develop sequence models using RNNs, LSTM, GRU, and transformers.	Apply, Analyze
CO-5	Evaluate advanced models such as GANs, autoencoders, diffusion models, and modern large-scale architectures.	Evaluate

Course Contents:

Unit	Contents	Lectures required
1	Introduction to biological and artificial neurons, linear perceptron model and its representation as a neuron, decision boundaries and limitations of linear models including XOR problem, feedforward neural networks, activation functions such as Sigmoid, Tanh and ReLU, and Softmax output layer. Measurement of errors using loss functions including Mean Squared Error and Cross-Entropy, gradient descent optimization methods including batch, stochastic and mini-batch approaches, delta rule and learning rate, backpropagation algorithm, bias-variance tradeoff, overfitting and its prevention using regularization techniques such as L1, L2, ElasticNet dropout and early stopping, and advanced	**5**

	optimization algorithms including Adagrad, RMSProp, Adadelta and Adam.	
2	Motivation from human visual perception and limitations of traditional feature engineering approaches, scaling issues in fully connected neural networks, convolution operation including filters, stride and padding, feature maps and hierarchical feature learning, pooling operations, convolutional neural network architectures (CNNs) and design principles, applications in image classification and recognition tasks.	**7**
3	Self-supervised pretraining approaches, overview of classical deep learning architectures including AlexNet, VGG, GoogleNet (Inception), ResNet and DenseNet, transfer learning techniques including feature extraction and fine-tuning, and introduction to few-shot learning for limited data scenarios.	**5**
4	Introduction to sequential data and embedding techniques, basic recurrent neural network architecture (RNN) and hidden state representation, challenges of vanishing and exploding gradients, advanced recurrent architectures including Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), bidirectional and deep recurrent neural networks	**7**
5	Limitations of recurrent models for long-range dependencies, attention mechanisms, self-attention and multi-head attention, positional encoding techniques, transformer architecture including encoder-decoder structure, and pretrained transformer models such as BERT.	**5**
6	Generative Adversarial Networks including generator and discriminator architecture, variants of GANs, autoencoders including sparse and denoising autoencoders.	**6**
7	Scaling laws and foundation models, large language models and their architectural overview, efficient transformer variants including sparse and linear attention mechanisms, retrieval-augmented generation techniques.	**7**
Total lectures		**42**

Suggested Text Book(s):

1. Ian Goodfellow, YoshuaBengio, and Aaron Courville, “Deep Learning”, The MIT Press, Cambridge, 2016.
2. Charu C. Aggarwal, “Neural Networks and Deep Learning: A Textbook”, Springer, 2018.

Suggested Reference Book(s):

1. Josh Patterson, Adam Gibson, "Deep Learning: A Practitioner's Approach", 2nd Edition, OReillyMedia.
2. Andrew Ng, “Machine Learning Yearning”, available online published by deeplearning.ai

Other useful resource(s):

Link to NPTEL course contents:

1. https://www.youtube.com/watch?v=aPfkYu_qiF4&list=PLyqSpQzTE6M9gCgajvQbc68Hk_JKGBAYT
2. https://www.youtube.com/watch?v=oJlb4jBbKWw&list=PLyqSpQzTE6M-_1jAqrFCsgCcuTYm_2urp
3. https://www.youtube.com/watch?v=qrsNX1Rwle0&list=PLZ2ps__7DhBbaMNZoyW2Hizl8DG6ik

kjo

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	1	1	1	1	2	2	1	**1.5**
CO-2	3	3	1	2	2	1	1	1	2	2	1	**1.7**
CO-3	2	2	3	2	3	1	1	1	2	2	1	**1.8**
CO-4	2	3	2	2	3	1	1	1	2	3	2	**2.0**
CO-5	2	2	1	2	2	2	1	1	1	1	3	**1.6**
Average	**2.4**	**2.4**	**1.6**	**1.8**	**2.2**	**1.2**	**1**	**1**	**1.8**	**2**	**1.6**	

Cryptocurrency Technologies

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: C++/Python, data structures,discrete mathematics

Course Objectives:

1. To understand fundamental cryptographic primitives used in cryptocurrencies.
2. To explain the working of blockchain and distributed consensus mechanisms.
3. To analyze the Bitcoin protocol and mining process.
4. To explore security, privacy, and anonymity issues in cryptocurrencies.
5. To study real-world applications such as smart contracts and decentralized finance.
6. To develop the ability to design and evaluate secure cryptocurrency systems.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain various cryptographic techniques like hashing and digital signatures**.**	Understand
CO-2	Explain the structure and functioning of blockchains and Merkle trees**.**	Understand
CO-3	Analyze and implement Bitcoin-like protocols.	Apply
CO-4	Evaluate security risks, attacks, and privacy mechanisms in cryptocurrencies.	Analyze
CO-5	Design basic decentralized applications and smart systems. Critically assess future trends and innovations in block-chain technology.	Create

Course Contents:

Unit	Contents	Lectures required
1	Evolution of Digital currency; limitations of traditional banking , Introduction to Bitcoin; overview of decentralized systems; role of trust and intermediaries, Centralized vs decentralized vs distributed systems; advantages and challenges , Applications of block-chain: finance, supply chain, healthcare, governance; legal and regulatory overview.	**4**
2	Fundamentals of cryptography: confidentiality, integrity, authentication, non-repudiation, symmetric vs asymmetric cryptography; RSA and key exchange basics, Public key infrastructure (PKI) and key management, Digital signatures: algorithms, verification, security properties, Cryptographic hash functions: SHA family, properties (collision resistance, avalanche effect), Hash pointers and tamper-evident structures, Merkle trees: construction, proof of inclusion, efficiency, Applications of Merkle trees in blockchain verification and scalability	**8**

3	Cryptographic Data Structures & Blockchain:Blockchain fundamentals: blocks, headers, transactions, timestamps, Append-only ledger and immutability; trustless systems, Distributed ledger technology (DLT) and peer-to-peer networks, Transaction lifecycle: creation, propagation, validation **,**Consensus basics: agreement in distributed systems; Byzantine fault tolerance, Forks (soft vs hard), longest chain rule, and network consistency.	**7**
4	Bitcoin architecture and working: Keys and addresses as identities , Transactions and scripting basics, UTXO model and transaction lifecycle, Proof of Work (PoW) and mining process, Incentives and game theory in Bitcoin, Mining hardware: ASICs and mining pools , Alternative consensus mechanisms (PoS, Peercoin)	**5**
5	Bitcoin block structure: block header, Merkle root, nonce, Wallet technologies: hot wallets, cold wallets, hardware wallets, Key management: multi-signature, secret sharing, threshold cryptography , Financial transparency tools: proof of reserves, proof of liabilities	**5**
6	Security and Attacks**:** Threat model in cryptocurrencies; attack surfaces, Double spending attack and prevention mechanisms attack, selfish mining, eclipse attacks **,** Network security: Sybil attacks, DDoS, countermeasures and best practices.	**4**
7	Privacy and Anonymity : Privacy concepts: anonymity, pseudonymity, unlinkability, Transaction graph analysis and tracing techniques, De-anonymization attacks at network and application layer, Privacy-enhancing technologies: mixers, CoinJoin, blind signatures, zero-knowledge proofs	**6**
8	Advanced Applications & Future Trends : Smart contracts and smart property (e.g., Ethereum basics), Micropayments, payment channels (Lightning Network), auctions, voting systems, prediction markets, Scalability challenges, regulatory issues, future research directions and innovations	**3**
Total lectures		**42**

Suggested Text Book(s):

1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder, Jeremy ClarkBitcoin and Cryptocurrency Technologies
2. Satoshi NakamotoBitcoin: A Peer-to-Peer Electronic Cash System
3. Michael NielsenHow the Bitcoin Protocol Actually Works

Suggested Reference Book(s):

1. Andreas M. Antonopoulos & Gavin WoodMastering Ethereum: Building Smart Contracts and DAppsImran BashirMastering Blockchain
2. Paul Vigna & Michael J. CaseyThe Age of Cryptocurrency
3. Narayanan Arvind et al. Additional lecture notes and Princeton course materials
4. Michael NielsenHow the Bitcoin Protocol Actually Works

Other useful resource(s):

6. Link to NPTEL course

contents:https://www.youtube.com/watch?v=mzPoUjQC4WU&list=PLHRLZtgrF2jl8yqucJsMFqh5XpRLTgCI4

7. Link to NPTEL course contents:https://www.youtube.com/watch?v=GstOwbLyeYE&pp=ygUnQ3J5cHRvY3VycmVuY3kgVGVjaG5vbG9naWVzIGNvdXJzZSBucHRs

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.4**	**2.4**	**2.0**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.2**	

Information Retrieval and Semantic Web

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Basic knowledge of Java/Python. Knowledge of data structures and algorithms.

Course Objectives:

1. To provide a comprehensive understanding of information retrieval principles and Semantic Web technologies.
2. To learn and analyze the information retrieval techniques.
3. To apply the information retrieval algorithms for real time applications.
4. To understand and evaluate the applications of information retrieval techniques

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Apply fundamental information retrieval models and techniques for efficient data retrieval.	Apply
CO-2	Apply ontology development methods and Semantic Web technologies for knowledge representation**.**	Understand
CO-3	Analyze query processing, clustering, and indexing techniques for large-scale data.	Apply
CO-4	Evaluate semantic web services and intelligent retrieval systems for real-world applications.	Analyze
CO-5	To understand and evaluate the applications of information retrieval techniques.	Create

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Information Retrieval (IR) and its applications: IR process and architecture, Classic IR models (Boolean, Vector Space, Probabilistic), Structured and unstructured text retrieval, Retrieval evaluation (precision, recall, F-measure), Evaluation metrics: precision, recall, F1-score, MAP, NDCG.	**5**
2	Query languages: keyword-based, structured queries (SQL-like, XML queries), Pattern matching and text search techniques, Query operations: intersection, union, ranking, filtering, Relevance feedback (explicit and implicit), Rocchio algorithm, Query expansion, synonym handling, multimedia query languages	**5**
3	Document preprocessing: tokenization, normalization, stop-word removal, Stemming and lemmatization techniques , Text compression: dictionary-based and statistical methods, Clustering techniques: hierarchical, k-means, evaluation of clusters, Inverted index:	**6**

	construction, storage, optimization, Ranking algorithms (TF-IDF, cosine similarity) and IR user interfaces	
4	Multimedia IR: concepts and challenges:Data models for multimedia (image, audio, video retrieval), Feature extraction: color, texture, shape descriptors, Web search engines: crawling, indexing, ranking (PageRank overview), Distributed IR: meta-search engines, parallel processing, scalability.	**5**
5	Evolution from Web 1.0 to Web 3.0 (Semantic Web): Limitations of syntactic web and need for semantics, Semantic Web architecture (layered model), Metadata and knowledge representation techniques, Ontologies: taxonomy, thesaurus, ontology classification	**5**
6	Resource Description Framework (RDF): triples, graphs, syntax, RDF Schema (RDFS): classes, properties, inheritance, Web Ontology Language (OWL): Lite, DL, Full, Ontology components: classes, properties, individuals, data types, Rule languages: RuleML, SWRL; inference and reasoning, Semantic reasoning: forward chaining, backward chaining, inference problems	**6**
7	Ontology development methodologies (Uschold& King, TOVE): Ontology editors: Protégé and tools for ontology creation, Ontology libraries and upper ontologies, Reasoners: Pellet, Hermit; consistency checking and classification, Semantic desktop, software agents, and intelligent information systems	**5**
8	Web services fundamentals: SOAP, REST, WSDL: Semantic Web services: OWL-S and service ontology, Service discovery, composition, and execution, Applications: geospatial systems, knowledge graphs, AI agents, Challenges: scalability, interoperability, big data integration, future trends	**5**
Total lectures		**42**

Suggested Text Book(s):

1. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze
   Introduction to Information Retrieval
2. Grigoris Antoniou & Frank van Harmelen
   A Semantic Web Primer

Suggested Reference Book(s):

1. Ricardo Baeza-Yates & Berthier Ribeiro-Neto
   Modern Information Retrieval
2. Ian H. Witten et al.
   Managing Gigabytes: Compressing and Indexing Documents
3. Andreas M. Antonopoulos *(for semantic-blockchain integration topics, optional)*
4. Research papers from IEEE, ACM, Springer, Elsevier

Other useful resource(s):

1. Link to NPTEL course contents:https://onlinecourses.nptel.ac.in/noc26_cs13/preview
2. Link to NPTEL course contents:https://nptel.ac.in/courses/106106852

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.4**	**2.4**	**2.0**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.2**	

Cloud Computing Essentials: Azure and AWS

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Basic knowledge of Java/Python. Basic Computer Fundamentals, Networking Fundamentals, Database Basics.

Course Objectives:

1. To build strong conceptual and practical understanding of cloud computing.
2. To enable hands-on expertise in AWS and Microsoft Azure core and advanced services.
3. To design scalable, secure, and fault-tolerant cloud architectures.
4. To implement networking, storage, compute, and database services.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To analyze cloud deployment and service models for real-world scenarios	Apply
CO-2	To deploy and manage AWS EC2, Azure VM, storage, and networking services	Understand
CO-3	Develop server less and event-driven applications. Use monitoring, logging, and auditing tools effectively.	Apply
CO-4	Evaluate automate infrastructure using Cloud Formation and ARM templates...	Analyze
CO-5	To understand and evaluate design multi-cloud and hybrid architectures.	Create

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Cloud Computing: Evolution from traditional IT to cloud, Characteristics: On-demand, elasticity, pay-as-you-go, Cloud service providers overview, Cloud Deployment Model, Public vs Private vs Hybrid vs Community cloud, Multi-cloud strategies , Use cases and decision factors, IaaS, PaaS, SaaS (deep comparison), Real-world examples, Responsibility sharing model, Virtualization & Containers, Hypervisors (Type 1 & Type 2), Virtual Machines vs Containers, Docker basics and container evolution.	**5**
2	AWS Global Infrastructure, Regions, Availability Zones, Edge locations, Fault isolation and resilience. AWS IAM, Users, Groups, Roles, Policies (JSON structure), IAM best practices (least privilege), MFA, API keys, credential management, EC2 (Compute Services), Instance types (General, Compute, Memory optimized), AMI, Launch templates , Key pairs and security groups, Pricing models (On-demand, Spot, Reserved, Dedicated), AWS Storage Basics, EBS volumes and types , EFS overview, EBS vs Instance Store.	**5**

3	AWS Advanced Services, Amazon S3, Buckets, Objects, metadata , Storage classes (Standard, IA, Glacier), Lifecycle policies , Versioning and replication, Static website hosting, Load Balancing & Auto Scaling, ELB types (CLB, ALB, NLB), Target groups, Health checks , Auto Scaling group, AWS Databases, RDS architecture and engines, Multi-AZ and Read replicas, Aurora advantages, DynamoDB (tables, partition key, indexing).	**6**
4	Caching (ElasticCache) , CDN & Edge Services, CloudFront working , Caching strategies, Signed URLs and cookies, Global Accelerato, Serverless Computing, AWS Lambda architecture, Event-driven computing, API Gateway integration, Cognito for authentication Virtual Private Cloud (VPC) Fundamentals, CIDR blocks, Subnetting (public/private), IP addressing, Internet Connectivity, Internet Gateway , NAT Gateway vs NAT Instance, Route tables, Security Mechanisms, Security Groups (stateful), NACL (stateless)	**5**
5	Best practices, Advanced Networking, VPC Peering, Endpoints (Interface & Gateway), Bastion host, Hybrid Connectivity, Site-to-Site VPN, Client VPN, Direct Connect, Route 53 routing policies, Health checks, VPC Flow Logs, Azure Fundamentals, Azure architecture, Subscriptions and resource groups, Regions and availability zones, Azure Active Directory, Users, Groups, Roles, RBAC vs IAM, MFA & Self-Service Password Reset	**6**
6	Privileged Identity Management, Azure Virtual Machines, VM sizes and series, Availability Sets & Zone, VM Scale Sets, VM images and extension, Azure Storage, Storage accounts , Blob storage (Hot, Cool, Archive), File, Queue, Table storage, Data redundancy (LRS, GRS, ZRS), Static website hosting	**5**
7	Azure Databases, Azure SQL Database, Azure MySQL, Cosmos DB (NoSQL models), Redis Cache, Azure Networking, Virtual Network (VNet), Subnets and IP addressing, NSG and ASG, Service endpoints . Load Balancing & DNS, Azure Load Balancer, Application Gateway (Layer 7), Azure DNS zones and records	**4**
8	ARM Templates, Template structure, Parameters, variables, outputs, Deployment strategies , Monitoring & Recovery, Azure Monitor and Logs, Azure Backup, Azure Site Recovery, DevOps & Automation, CI/CD pipelines, GitHub Actions, Azure DevOps, Terraform basics, GitOps principles Containers & Cloud Architecture, Docker fundamentals , Kubernetes (AKS basics), Microservices architecture, High availability & multi-region design	**6**
Total lectures		**42**

Suggested Text Book(s):

1. Thomas Erl – Cloud Computing: Concepts, Technology & Architecture
2. Ben Piper – AWS Certified Solutions Architect Study Guide
3. Microsoft – Azure Fundamentals (AZ-900)
4. Michael Kavis – Architecting the Cloud

Suggested Reference Book(s):

1. Thomas Erl, Ricardo Puttini&Zaigham Mahmood – *Cloud Computing: Concepts, Technology & Architecture*.
2. Rajkumar Buyya, Christian Vecchiola& S. ThamaraiSelvi – *Mastering Cloud Computing*

Other useful resource(s):

1. Link to NPTEL course contents:

https://www.youtube.com/watch?v=NzZXz3fJf6o&list=PLShJJCRzJWxhz7SfG4hpaBD5bKOloWx9J

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.4**	**2.4**	**2.0**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.2**	

Machine Learning and Big Data

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Basics of Machine Learning, Python Programming, Basic NLP

Course Objectives:

1. To understand machine learning and big data concepts
2. To study distributed architecture for big data processing.
3. To study and apply scalable machine learning algorithms for big data.
4. To implement supervised and unsupervised algorithms for scalable solutions.
5. To design a recommendation system for real life NLP dataset.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain machine learning and big data concepts	Understand
CO-2	Apply data science fundamentals to big data	Apply
CO-3	Implement machine learning algorithms using Apache Spark	Apply
CO-4	Analyze supervised and unsupervised learning algorithms	Analyze
CO-5	Design recommendation system using MLlib library for NLP dataset	Create
CO-6	Explain machine learning and big data concepts	Understand

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to Machine Learning:** Fundamentals and Definition of Machine Learning, Types of Machine Learning, Learning of Models Based on the First Criteria, Learning of Models Based on Second Criteria (Batch Mode Learning and Online Mode Learning), Applications of Machine Learning	6
2	**Introduction to Big Data:** Big Data: Why and Where, Characteristics of Big Data and Dimensions of Scalability, Data Science: Getting Value out of Big Data, Foundations for Big Data Systems and Programming	3
3	**Apache Spark:** Core Concepts, Spark Components, Architecture, working of Spark, Apache Spark Environment Setup and Configuration: Installation on Linux, Mac and Windows. Standalone setup and local	5

	cluster setup	
4	**Apache Spark MLlib:** Programming with RDDs, Statistics and Linear Algebra Preliminaries,Data Exploration: Univariate Analysis & Multivariate Analysis, Data Manipulation: Feature Extraction, Feature Transform, Feature Selection, Unbalanced data: Under sampling, Regression: Linear Regression, Generalized linear regression, Decision tree Regression, Random Forest Regression, Gradient-boosted tree regression,Regularization: Ordinary least squares regression, Ridge regression, Least Absolute Shrinkage and Selection Operator (LASSO), Elastic net	**9**
5	**Supervised and Un-supervised Learning with Spark:** Classification: Binomial logistic regression, Multinomial logistic regression, Decision tree Classification, Random Forest Classification, Gradient-boosted tree Classification, XGBoost: Gradient-boosted tree Classification, Naive Bayes Classification	**10**
6	**Natural Language Processing with Apache Spark:** Text Mining: Text Collection, Text Preprocessing, Text Classification, Sentiment analysis, N-grams and Correlations, Topic Model: Latent Dirichlet Allocation	**6**
7	**Recommendation Engine with Distributed Framework:** Recommender systems: Alternating Least Squares	**3**
Total lectures		**42**

Suggested Text Book(s):

1. **Machine Learning and Big Data: Concepts, Algorithms, Tools and Applications**

Editor(s):Uma N. Dulhare, Khaleel Ahmad, KhairolAmali Bin Ahmad
(Publisher: Scrivener Publishing LLC)
(Print ISBN:9781119654742 |Online ISBN:9781119654834)
https://onlinelibrary.wiley.com/doi/book/10.1002/9781119654834

2. **Practical Machine Learning with Spark**

Gourav Gupta, Dr. Manish Gupta, Dr. Inder Singh Gupta
(BPB Publisher) (ISBN: 9789391392086, eISBN: 9789391392130)
https://in.bpbonline.com/products/practical-machine-learning-with-spark

Suggested Reference Book(s):

1. Spark: The Definitive Guide, Big Data Processing Made Simple

Bill Chambers and MateiZaharia
(Publisher: O'Reilly)(ISBN: 9781491912201)
https://www.oreilly.com/library/view/spark-the-definitive/9781491912201/

2. Essential PySpark for Scalable Data Analytics: A beginner's guide to harnessing the power and ease of PySpark 3

SreeramNudurupati
(Publisher: packtpublications)(ISBN-13 :9781800568877)
https://www.packtpub.com/en-ch/product/essential-pyspark-for-scalable-data-analytics-9781800568877

Other useful resource(s):

1. Coursera: Big Data Specialization: UCSanDiego
https://www.coursera.org/specializations/big-data
2. Coursera: Machine Learning with PySpark: Edureka
https://www.coursera.org/learn/machine-learning-with-pyspark
3. Coursera:Machine Learning with Apache Spark: IBMhttps://www.coursera.org/learn/machine-learning-with-apache-spark
4. Coursera: Introduction to Big Data with Spark and Hadoop: IBM
https://www.coursera.org/learn/introduction-to-big-data-with-spark-hadoop

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (Pos)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	2	2	1	1	1	1	1	1	**1.5**
CO-2	3	3	2	2	3	1	1	1	2	2	1	**2.0**
CO-3	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-4	2	3	2	3	2	1	1	1	2	2	2	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.6**	**2.2**	**2.2**	**2.6**	**1.0**	**1.0**	**1.2**	**2.0**	**1.8**	**1.6**	

Secure Design of Software Systems

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Programming Fundamentals, Data Structures, Software Engineering

Course Objectives:

1. To understand fundamental principles of secure software design and security engineering practices.
2. To analyze common software vulnerabilities and attack vectors in modern software systems.
3. To apply secure design principles, threat modeling, and risk assessment techniques.
4. To design software architectures incorporating confidentiality, integrity, availability, and privacy requirements.
5. To implement secure coding practices and verification techniques throughout the Software Development Life Cycle (SDLC).

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain core concepts of software security, security goals, and threat landscape.	Understand
CO-2	Apply secure design principles and threat modeling techniques to software systems.	Apply
CO-3	Analyze software vulnerabilities and security risks using systematic approaches.	Analyze
CO-4	Evaluate secure architectures and defense mechanisms against common attacks.	Evaluate
CO-5	Design secure software systems integrating secure coding and testing practices.	Create

Course Contents:

Unit	Contents	Lectures Required
1	**Introduction to Software Security:** Security concepts, CIA triad, security goals, secure software engineering, security vs reliability, overview of cyber threats, security design lifecycle, principles of least privilege and defense in depth.	**5**
2	**Secure Software Development Life Cycle (SSDLC):** Security requirements engineering, risk assessment, security policies, secure design review, DevSecOps concepts, integration of security in Agile and DevOps models.	**6**
3	**Threat Modeling and Risk Analysis:** Risk Management Life Cycle, Risk Profiling, Risk Exposure Factors, STRIDE and DREAD models, attack surface analysis, misuse/abuse cases, adversarial thinking, threat identification and mitigation strategies.	**6**

4	**Secure Design Principles and Architecture:** Secure architectural patterns, trust boundaries, secure interfaces, authentication and authorization models, access control mechanisms, secure microservices and distributed systems design.	**7**
5	**Common Software Vulnerabilities:** OWASP Top 10, buffer overflow, injection attacks (SQL/Command), cross-site scripting (XSS), cross-site request forgery (CSRF), insecure deserialization, race conditions.	**7**
6	**Secure Coding Practices:** Input validation, error handling, secure API design, cryptographic best practices, secrets management, code review techniques, static and dynamic analysis tools.	**6**
7	**Security Testing and Verification:** Secure Software Development Life Cycle, Risk Based Security Testing, Prioritizing Security Testing With Threat Modeling, Penetration testing basics, fuzz testing, vulnerability assessment, security auditing, automated testing tools, secure deployment and monitoring practices.	**5**
Total lectures		**42**

Suggested Text Book(s)

1. Ross Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems, Wiley, 3rd Edition.
2. Gary McGraw, Software Security: Building Security In, Addison-Wesley.
3. Mark Dowd, John McDonald & Justin Schuh, The Art of Software Security Assessment, Addison-Wesley.
4. Julia H. Allen et al., Software Security Engineering, SEI Series, Addison-Wesley.

Suggested Reference Book(s)

1. Michael Howard & David LeBlanc, Writing Secure Code, Microsoft Press.
2. Adam Shostack, Threat Modeling: Designing for Security, Wiley. OWASP Foundation Documentation (Latest Edition). Viega& McGraw, Building Secure Software, Addison-Wesley.

Other useful resource(s):

1. Link to NPTEL course contents:Software Security Courses:https://www.youtube.com/watch?v=2xrsRlWmcRQ
2. Link to NPTEL course contents:https://onlinecourses-archive.nptel.ac.in/noc19_cs24/preview
3. Link to topics related to course:
   - OWASP Secure Coding Practices Guide
   - https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/
   - https://www.youtube.com/watch?v=teLmHBPGiV8

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2

3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	2	1	1	2	**1.8**
CO-2	3	3	3	2	3	1	1	2	2	2	2	**2.2**
CO-3	3	3	2	3	3	1	1	2	2	2	2	**2.2**
CO-4	2	3	3	3	3	1	1	2	2	2	3	**2.3**
CO-5	3	2	3	2	3	1	1	3	3	2	2	**2.3**
Average	**2.8**	**2.6**	**2.6**	**2.2**	**2.8**	**1.0**	**1.0**	**2.2**	**2.0**	**1.8**	**2.2**	

Fog and Edge Computing

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Computer Networks, Cloud Computing

Course Objectives:

1. To introduce the fundamental concepts of IoT, cloud computing, and fog and edge computing paradigms.
2. To understand architectures, integration models, and challenges in fog and edge computing environments.
3. To develop knowledge of middleware design, data management, and resource allocation techniques in fog systems.
4. To explore security, privacy, machine learning, and optimization techniques in IoT and fog computing.
5. To analyze and design real-world applications of fog and edge computing using simulation tools and case studies.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understand the fundamental concepts of IoT, cloud computing, and the need for fog and edge computing paradigms.	Understand
CO-2	Analyze architectures, hierarchies, and integration models of IoT, fog, and cloud systems.	Analyze
CO-3	Apply middleware concepts and data management techniques for efficient fog and edge computing environments.	Apply
CO-4	Evaluate security challenges, privacy issues, and machine learning approaches in IoT and fog systems.	Analyze
CO-5	Analyze optimization techniques, performance metrics, and resource management strategies in fog computing.	Analyze
CO-6	Design and assess real-world applications of fog and edge computing using simulation tools and case studies.	Create

Course Contents:

Unit	Contents	Lectures required
1	**Foundations of IoT, Fog & Edge Computing** Introduction to IoT and emerging computing paradigms Cloud-centric IoT (CIoT) and its limitations Need for Fog and Edge Computing Comparison of Cloud, Fog, Edge, and Mist computing Architecture and hierarchy of Fog and Edge Computing Advantages (SCALE: Security, Cognition, Agility, Latency, Efficiency)	10

	Core mechanisms (SCANC: Storage, Compute, Acceleration, Networking, Control) Business models (XaaS, application and support services) Opportunities and research challenges	
2	**Architecture, Integration & Challenges** Federated edge systems and architecturesNetworking challenges in edge environmentsResource discovery, management, and service migrationLoad balancing and mobility supportIntegration of IoT, Fog, and Cloud (C2F2T model)Modeling techniques such as Analytical models, Petri Nets, and ILPRole of 5G in edge computingNetwork slicing and orchestration	**8**
3	**Middleware & Data Management** Need and role of middleware in fog and edge computing, Middleware design goals and architecture, Device discovery and context-awareness Scheduling, resource allocation, and execution management, Container-based edge computing using Docker and lightweight clustersFog data lifecycle and management, Data preprocessing including cleaning, fusion, and edge analytics, Data storage, placement, and privacy	**8**
4	**Security, Machine Learning & Optimization** Security challenges in IoT and fog environments, Privacy issues and attack models such as DDoS and malware, Machine learning for IoT security, Machine learning techniques including classification, clustering, and ANN, Optimization problems in fog computing, Performance metrics including QoS, energy, cost, and resource utilization	**6**
5	**Applications & Case Studies** Fog computing for big data analytics, Smart home and smart city applications, Healthcare monitoring systems, Smart surveillance and video analytics, Intelligent transportation systems, Testing perspectives of fog-based IoT applications	**6**
6	**Simulation, Legal & Emerging Topics** Simulation tools such as iFogSim and modeling of fog environments Case studies and performance evaluation, Legal and regulatory issues including data protection and privacy laws, Future research directions and emerging trends	**4**
Total lectures		**42**

Suggested Text Book(s):

1. Fog and Edge Computing: Principles and Paradigms by Rajkumar Buyya and Satish Narayana Srirama, Wiley, 2019
2. Edge Computing: Vision and Challenges by Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu, IEEE Internet of Things Journal
3. Cloud Computing: Concepts, Technology & Architecture by Thomas Erl, Ricardo Puttini, and Zaigham Mahmood, Pearson

Suggested Reference Book(s):

1. Fog Computing in the Internet of Things: Intelligence at the Edge by Amir M. Rahmani, PasiLiljeberg, Jürgo-SörenPreden, and Axel Jantsch, Springer, 2018
2. Fog/Edge Computing for Security, Privacy, and Applications edited by Wei Chang and Jie Wu, Springer, 2021
3. Energy Conservation Solutions for Fog-Edge Computing Paradigms edited by Rajeev Tiwari, Mamta Mittal, and Lalit Mohan Goyal, Springer, 2022

4. Mastering Cloud Computing by Rajkumar Buyya, Christian Vecchiola, and S. ThamaraiSelvi, McGraw Hill

Other useful resource(s):

1. NPTEL Online Course on Cloud Computing and Distributed Systems (IIT lectures)
2. IEEE Xplore Digital Library for latest research papers on fog and edge computing
3. OpenFog Consortium architecture and white papers on fog computing
4. ETSI resources on Multi-access Edge Computing (MEC)
5. iFogSim toolkit for modeling and simulation of fog environments
6. GitHub repositories and open-source platforms for IoT and edge computing implementations

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.63**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.09**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.18**
CO-6	2	2	2	2	2	1	1	1	3	2	2	**1.81**
Average	**2.6**	**2.3**	**2.3**	**2**	**2.5**	**1**	**1**	**1.1**	**2.3**	**1.8**	**2.1**	

Machine Learning and Natural Language Processing

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Probability and Statistics, Linear Algebra, Python Programming, Data Structures and Algorithms.

Course Objectives:

1. To introduce the principles, algorithms, and mathematical foundations of supervised and unsupervised machine learning.
2. To develop understanding of ensemble methods, model selection, regularisation, and performance evaluation techniques.
3. To familiarise students with the linguistic foundations of NLP and classical text processing pipelines including language models, POS tagging, and parsing.
4. To build understanding of deep learning architectures — RNNs, LSTMs, and Transformers — and their application to NLP tasks.
5. To explore state-of-the-art pre-trained language models (BERT, GPT, T5) and analyse their application to downstream NLP tasks.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the foundational concepts of machine learning including supervised, unsupervised, and reinforcement learning paradigms, along with bias-variance trade-off and model evaluation techniques.	Understand
CO-2	Explain core NLP concepts including text preprocessing, language models, POS tagging, parsing, and sentiment analysis using both statistical and neural approaches.	Understand
CO-3	Apply classical ML algorithms — linear/logistic regression, SVMs, decision trees, and ensemble methods — to solve real-world structured and text-based problems.	Apply
CO-4	Apply deep learning architectures — RNNs, LSTMs, and Transformers — and pre-trained language models (BERT, GPT, T5) to NLP tasks such as classification, NER, and machine translation.	Apply
CO-5	Analyse challenges in ML and NLP pipelines including statistical heterogeneity, model selection, hyperparameter tuning, and performance interpretation on benchmark datasets.	Analyze

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Machine Learning: Types of ML — supervised, unsupervised, semi-supervised, reinforcement learning; the ML pipeline — data collection, preprocessing, feature engineering, model training, evaluation, deployment; bias-variance trade-off; overfitting and underfitting; cross-validation; performance metrics — accuracy, precision, recall, F1, ROC-AUC.	**4**

2	Supervised Learning Algorithms: Linear regression — OLS, gradient descent, regularisation (Ridge, Lasso); logistic regression — binary and multiclass; Support Vector Machines — hard and soft margin, kernel trick; k-Nearest Neighbours; Naive Bayes classifier; decision trees — ID3, C4.5, CART; ensemble methods — bagging, random forests, boosting (AdaBoost, Gradient Boosting, XGBoost).	**9**
3	Unsupervised Learning and Dimensionality Reduction: K-Means clustering — algorithm, elbow method; hierarchical clustering — agglomerative, dendrogram; DBSCAN; Gaussian Mixture Models and EM algorithm; dimensionality reduction — PCA, t-SNE; anomaly detection.	**5**
4	Introduction to Natural Language Processing: Text as data — challenges and pipeline; text preprocessing — tokenisation, stop-word removal, stemming, lemmatisation; text representation — Bag of Words, TF-IDF, n-grams; word embeddings — Word2Vec (CBOW, Skip-gram), GloVe, FastText; introduction to NLTK and spaCy.	**5**
5	Core NLP Tasks — Statistical and Classical Approaches: Language models — N-gram models, smoothing techniques; Part-of-Speech (POS) tagging — HMM-based tagging, Viterbi algorithm; Named Entity Recognition (NER) — rule-based and CRF-based; syntactic parsing — constituency and dependency parsing; sentiment analysis — lexicon-based and ML-based approaches; topic modelling — LDA.	**7**
6	Deep Learning for NLP: Recurrent Neural Networks (RNNs) — vanishing gradient problem; Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU); sequence-to-sequence models with attention mechanism; Convolutional Neural Networks for text classification; Transformer architecture — self-attention, multi-head attention, positional encoding; encoder-decoder structure.	**6**
7	Pre-trained Language Models and Transfer Learning: BERT — pre-training objectives (MLM, NSP), fine-tuning for downstream tasks; GPT — autoregressive language modelling; RoBERTa, DistilBERT; T5 and BART for seq2seq tasks; prompt engineering and in-context learning; introduction to Large Language Models (LLMs); Hugging Face Transformers library — pipelines and tokenisers.	**6**
Total lectures		**42**

Suggested Text Book(s):

1. Bishop, C. M., Pattern Recognition and Machine Learning, Springer, 2006.
2. Jurafsky, D. and Martin, J. H., Speech and Language Processing, 3rd Edition (Draft), Pearson, 2023. [Available free online]
3. Géron, A., Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition, O'Reilly Media, 2022.
4. Goldberg, Y., Neural Network Methods for Natural Language Processing, Morgan & Claypool Publishers, 2017.

Suggested Reference Book(s):

1. Mitchell, T. M., Machine Learning, McGraw-Hill Education, 1997.
2. Manning, C. D. and Schütze, H., Foundations of Statistical Natural Language Processing, MIT Press, 1999.
3. Goodfellow, I., Bengio, Y., and Courville, A., Deep Learning, MIT Press, 2016.
4. Tunstall, L., von Werra, L., and Wolf, T., Natural Language Processing with Transformers, O'Reilly Media, 2022.

Other useful resource(s):

1. Hugging Face Transformers Documentation: https://huggingface.co/docs/transformers
2. spaCy Documentation and Tutorials: https://spacy.io
3. NLTK Book (Free): https://www.nltk.org/book
4. fast.ai Practical Deep Learning for Coders: https://course.fast.ai
5. NPTEL – Introduction to Machine Learning (IIT Madras): https://nptel.ac.in/courses/106106139
6. NPTEL – Deep Learning for NLP: https://nptel.ac.in/courses/106106198
7. Papers With Code – NLP Benchmarks: https://paperswithcode.com/area/natural-language-processing
8. Stanford CS224N – NLP with Deep Learning (Lecture Notes & Videos): https://web.stanford.edu/class/cs224n

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	1.6
CO-2	3	2	2	2	2	1	1	1	2	1	2	1.7
CO-3	3	3	3	2	3	1	1	1	2	2	2	2.1
CO-4	3	3	3	2	3	1	1	1	2	2	2	2.1
CO-5	2	3	2	3	2	1	1	1	2	2	3	2.0
Average	**2.8**	**2.6**	**2.4**	**2.0**	**2.4**	**1.0**	**1.0**	**1.0**	**2.0**	**1.6**	**2.2**	

Ethical Hacking & Prevention

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Basic knowledge of Computer Networks and Operating Systems

Course Objectives:

1. To understand the fundamentals of ethical hacking, cybersecurity, and threat landscape.
2. To learn various attack techniques used by hackers and their prevention mechanisms.
3. To analyze vulnerabilities in systems, networks, and web applications.
4. To apply penetration testing tools and methodologies in real-world scenarios.
5. To develop secure systems by implementing best security practices and counter measures.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain fundamentals of ethical hacking, cybersecurity concepts, and types of threats.	Understand
CO-2	Apply various scanning, enumeration, and vulnerability assessment techniques.	Apply
CO-3	Analyze different attack vectors including network, system, and web-based attacks.	Analyze
CO-4	Evaluate security tools and perform penetration testing.	Evaluate
CO-5	Design secure systems and implement preventive measures against cyber threats.	Create

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Ethical Hacking: Basics of Cybersecurity, Types of Hackers, Cyber Laws, Footprinting	**4**
2	Footprinting& Reconnaissance: WHOIS, DNS, Social Engineering	**5**
3	Scanning & Enumeration: Port Scanning, Vulnerability Scanning, SNMP, LDAP	**6**
4	System Hacking: Password Cracking, Malware, Privilege Escalation	**6**
5	Network Attacks & Prevention: DoS/DDoS, MITM, Firewalls, IDS/IPS	**7**
6	Web Application Security: SQL Injection, XSS, CSRF	**7**
7	Wireless & Mobile Security: Wi-Fi Attacks, Mobile Security	**4**

8	Penetration Testing & Countermeasures: Tools and Phases	**3**
Total lectures		**42**

Suggested Text Book(s):

1. Kevin Mitnick, The Art of Invisibility, Little, Brown and Company, 2017
2. Michael T. Simpson, Ethical Hacking and Network Defense, Cengage Learning, 3rd Edition
3. Rafay Baloch, Ethical Hacking and Penetration Testing Guide, CRC Press, 2014
4. William Stallings, Network Security Essentials, Pearson, 6th Edition

Suggested Reference Book(s):

1. Jon Erickson, Hacking: The Art of Exploitation, No Starch Press
2. Georgia Weidman, Penetration Testing: A Hands-On Introduction to Hacking, No Starch Press

Other useful resource(s):

1. OWASP Foundation: https://owasp.org
2. Kali Linux Documentation: https://www.kali.org

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO 1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO1 0	PO1 1	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.4**	**2.4**	**2.0**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.2**	

Introduction to Large Scale Database Systems

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Database Management Systems, Data Structures, Operating Systems

Course Objectives:

1. To understand the design challenges of large-scale and distributed database systems.
2. To study modern data storage architectures used in cloud and big data platforms.
3. To analyze scalability, consistency, availability, and fault tolerance in distributed databases.
4. To explore NoSQL databases and large-scale data processing frameworks.
5. To design scalable data management solutions for real-world applications.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain architecture and principles of large-scale database systems.	Understand
CO-2	Apply distributed database concepts for scalable data storage and processing.	Apply
CO-3	Analyze consistency models, replication strategies, and fault tolerance mechanisms.	Analyze
CO-4	Evaluate NoSQL and big data technologies for different application scenarios.	Evaluate
CO-5	Design scalable database solutions for large-scale real-world systems.	Create

Course Contents:

Unit	Contents	Lectures Required
1	**Introduction to Large-Scale Data Systems:** Evolution from traditional DBMS to distributed databases, data-intensive applications, scalability challenges, and centralized vs distributed databases, overview of cloud data platforms.	5
2	**Distributed Database Architecture:** Data partitioning (sharding), replication strategies, distributed query processing, distributed transactions, two-phase commit protocol, distributed indexing.	7
3	**Consistency and CAP Theorem:** Consistency models, CAP theorem, ACID vs BASE properties, eventual consistency, quorum-based systems, concurrency control in distributed environments.	6
4	**NoSQL Database Systems:** Key-value stores, document databases, column-family databases, graph databases; systems such as MongoDB, Cassandra, HBase; schema design for scalability.	7

5	**Large-Scale Data Processing Frameworks:** MapReduce paradigm, Hadoop ecosystem, Spark architecture, distributed storage using HDFS, batch vs stream processing concepts.	6
6	**Fault Tolerance and Reliability:** Data replication, failure detection, recovery mechanisms, consensus algorithms (Paxos/Raft overview), load balancing, distributed monitoring.	6
7	**Cloud Databases and Modern Data Platforms:** Database-as-a-Service (DBaaS), serverless databases, data lakes and data warehouses, real-time analytics systems, security and privacy in large-scale databases.	5
Total lectures		**42**

Suggested Text Book(s)

1. Martin Kleppmann, Designing Data-Intensive Applications, O'Reilly Media.
2. Hector Garcia-Molina, Jeffrey Ullman & Jennifer Widom, Database Systems: The Complete Book, Pearson.
3. Raghu Ramakrishnan & Johannes Gehrke, Database Management Systems, McGraw-Hill.Tom White, Hadoop: The Definitive Guide, O'Reilly.

Suggested Reference Book(s)

1. Ozsu&Valduriez, Principles of Distributed Database Systems, Springer.
2. George Coulouris et al., Distributed Systems: Concepts and Design, Pearson.
3. Bill Chambers &MateiZaharia, Spark: The Definitive Guide, O'Reilly.
4. Kleppmann Lecture Notes on Distributed Systems (University of Cambridge).

Other useful resource(s):

1. MIT OpenCourseWare – Distributed Computer Systems Engineering (6.824): https://ocw.mit.edu/courses/6-824-distributed-computer-systems-engineering-spring-2006/
2. NPTEL – Big Data Computing (IIT Patna): https://onlinecourses.nptel.ac.in/noc24_cs130/
3. NPTEL – Distributed Systems: https://onlinecourses-archive.nptel.ac.in/noc17_cs42/
4. Apache Hadoop Official Documentation: https://hadoop.apache.org/docs/
5. MongoDB University Courses: https://university.mongodb.com

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	2	1	1	2	**1.8**
CO-2	3	3	3	2	3	1	1	2	2	2	2	**2.2**
CO-3	3	3	2	3	3	1	1	2	2	2	2	**2.2**
CO-4	2	3	3	3	3	1	1	2	2	2	3	**2.3**
CO-5	3	2	3	2	3	1	1	3	3	2	2	**2.3**
Average	**2.8**	**2.6**	**2.6**	**2.2**	**2.8**	**1.0**	**1.0**	**2.2**	**2.0**	**1.8**	**2.2**	

Introduction to DevOps

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: N/A

Course Objectives:

1. To understand DevOps principles, culture, and evolution in software development.
2. To apply DevOps practices like CI/CD, TDD, and automation.
3. To analyze cloud-native architectures and microservices.
4. To evaluate collaboration, workflows, and DevOps tools.
5. To measure DevOps performance using metrics and continuous improvement strategies.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain DevOps concepts, principles, and cultural transformation.	Understand
CO-2	Apply DevOps practices like CI/CD, TDD, and automation tools.	Apply
CO-3	Analyze cloud-native architectures and microservices-based systems.	Analyze
CO-4	Evaluate collaboration, workflows, and DevOps lifecycle processes.	Analyze
CO-5	Measure DevOps performance using metrics and continuous improvement strategies.	Evaluate

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to DevOps:** Evolution of software development (Waterfall vs Agile), innovation and business disruption, role of technology, DevOps definition, misconceptions, DevOps culture, characteristics, history and evolution of DevOps movement	**8**
2	**Thinking DevOps:** Social coding, version control and shared repositories, pair programming, working in small batches, MVP concept, feedback loops, TDD, BDD, cloud-native architecture, microservices, resilience engineering	**10**
3	**Working with DevOps:** Software development as craft vs factory, silos and bottlenecks, collaboration, Infrastructure as Code (IaC), ephemeral infrastructure, containers and Docker basics, immutable infrastructure, CI/CD pipelines, automation	**10**

4	**Organizing for DevOps:** Organizational impact, DevOps team structures, cross-functional teams, shared responsibility, misconceptions, consequences in workflows, building shared mindset	**7**
5	**Measuring DevOps:** DevOps metrics (deployment frequency, lead time), social metrics, continuous improvement, actionable metrics, blameless culture, measuring culture, reward systems	**7**
Total lectures		**42**

Suggested Text Book(s):

1. The Phoenix Project – Gene Kim, Kevin Behr, George Spafford
2. The DevOps Handbook – Gene Kim, Jez Humble, Patrick Debois
3. Accelerate – Nicole Forsgren, Jez Humble, Gene Kim

Suggested Reference Book(s):

1. Continuous Delivery – Jez Humble and David Farley
2. Site Reliability Engineering – Betsy Beyer et al.

Other useful resource(s):

1. NPTEL DevOps Courses:https://elearn.nptel.ac.in/shop/completed-courses/short-term-programs-completed/cicd-devops-automation-and-devsecops-automation/
2. IBM DevOps Learning Resources:https://developer.ibm.com/devpractices/devops/tutorials/
3. AWS DevOps Tutorials: https://market.tutorialspoint.com/course/aws-devops-engineer-certification-2022/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	2	1	1	1	2	1	2	1.8
CO-2	3	3	3	2	3	1	1	1	2	2	2	2.1
CO-3	3	2	3	3	3	1	1	1	2	2	2	2.2
CO-4	2	3	2	3	2	1	1	1	2	2	2	2.0
CO-5	3	2	3	2	3	1	1	2	3	2	2	2.2
Average	**2.8**	**2.4**	**2.6**	**2.4**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.0**	

Industrial Automation and IoT

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Basic Electronics, Microprocessors and Microcontrollers, Computer Networks, Programming in C/Python

Course Objectives:

1. To understand the architecture, components, and hierarchy of modern industrial automation systems including sensors, actuators, PLCs, and DCS.
2. To study SCADA systems, HMI design, and industrial communication protocols used in process control and manufacturing environments.
3. To explore the architecture, hardware platforms, and communication protocols of the Internet of Things and Industrial IoT.
4. To analyse edge computing, digital twin concepts, and IIoT data pipelines for industrial monitoring and predictive maintenance.
5. To examine cybersecurity threats, safety standards, and regulatory frameworks governing industrial automation and IIoT deployments.

Course Outcomes:

S.No.	Course Outcomes	Level of Attainment
CO-1	Explain the fundamental concepts of industrial automation, control systems, sensors, actuators, and programmable logic controllers (PLCs) used in modern manufacturing environments.	Understand
CO-2	Explain the architecture, communication protocols, and standards of the Industrial Internet of Things (IIoT) including MQTT, OPC-UA, Modbus, and PROFINET.	Understand
CO-3	Apply SCADA systems, HMI design principles, and industrial networking concepts to model real-world automated process control scenarios.	Apply
CO-4	Apply IoT sensing, edge computing, and data acquisition techniques to design IIoT data pipelines for industrial monitoring and predictive maintenance.	Apply
CO-5	Analyse industrial automation systems and IIoT deployments with respect to reliability, cybersecurity threats, real-time constraints, and safety standards.	Analyze

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Industrial Automation: Evolution of automation — manual, mechanised, automated, and smart manufacturing; automation hierarchy — field level, control level, supervisory level, enterprise level; types of automation — fixed, programmable, flexible; key components — sensors, actuators, controllers, HMI; introduction to Industry 4.0 and cyber-physical systems (CPS); overview of smart factories.	**5**

2	Sensors, Actuators and Signal Conditioning: Classification of sensors — temperature (RTD, thermocouple), pressure, flow, proximity, vision; signal types — analog vs. digital, 4–20 mA current loops; signal conditioning — amplification, filtering, ADC/DAC; actuators — electric motors (DC, stepper, servo), pneumatic and hydraulic actuators, solenoid valves; transducer characteristics — accuracy, resolution, linearity, repeatability.	**6**
3	Programmable Logic Controllers (PLCs) and Control Systems: PLC architecture — CPU, memory, I/O modules, power supply; PLC programming languages (IEC 61131-3) — Ladder Diagram, Function Block Diagram, Structured Text, Sequential Function Chart; closed-loop control — PID controllers, tuning methods; motion control basics; safety PLCs and functional safety standards (IEC 61508, IEC 62061); introduction to Distributed Control Systems (DCS).	**8**
4	SCADA, HMI and Industrial Networking: SCADA system architecture — field devices, RTUs/PLCs, communication network, SCADA server, historian; HMI design principles — alarm management, data visualisation, operator interface; industrial communication protocols — Modbus RTU/TCP, PROFIBUS, PROFINET, EtherNet/IP, OPC and OPC-UA; fieldbus vs. industrial Ethernet; network topologies in automation; introduction to time-sensitive networking (TSN).	**7**
5	Internet of Things (IoT) — Architecture and Protocols: IoT reference architecture — perception, network, and application layers; IoT hardware — microcontrollers (Arduino, ESP32), single-board computers (Raspberry Pi); wireless communication technologies — Wi-Fi, Bluetooth, Zigbee, Z-Wave, LoRa, NB-IoT, 5G; IoT application protocols — MQTT, CoAP, AMQP, HTTP/REST; IoT platforms — AWS IoT, Azure IoT Hub, Google Cloud IoT, Things Board.	**6**
6	Industrial IoT (IIoT) and Edge Computing: IIoT vs. consumer IoT; IIoT reference architecture (RAMI 4.0, IIRA); edge computing in industrial settings — edge gateways, fog nodes; real-time data acquisition and preprocessing; time-series data handling; digital twins — concept, architecture, and industrial applications; OPC-UA over MQTT (Sparkplug B); IIoT use cases — predictive maintenance, condition monitoring, energy management, asset tracking.	**7**
7	Cybersecurity, Safety and Standards in Industrial Automation and IIoT: Cybersecurity threats in OT/ICS environments — ransomware, man-in-the-middle, denial of service; ICS security frameworks — IEC 62443, NIST SP 800-82; network segmentation — DMZ, Purdue model, zero-trust architecture; secure remote access; safety instrumented systems (SIS) and safety integrity levels (SIL); regulatory and interoperability standards — ISO 9001, ISO 50001, IEEE 2413; emerging trends — AI in automation, autonomous robots, collaborative robots (cobots), 5G in IIoT.	**3**
Total lectures		**42**

Suggested Text Book(s):

1. Groover, M. P., Automation, Production Systems, and Computer-Integrated Manufacturing, 4th Edition, Pearson, 2015.
2. Mackay, S., Wright, E., Reynders, D., and Park, J., Practical Industrial Data Networks: Design, Installation and Troubleshooting, Newnes, 2004.
3. Bahga, A. and Madisetti, V., Internet of Things: A Hands-On Approach, VPT, 2014.
4. Bolton, W., Programmable Logic Controllers, 6th Edition, Newnes, 2015.

Suggested Reference Book(s):

1. Mahalik, N. P., Fieldbus Technology: Industrial Network Standards for Real-Time Distributed Control, Springer, 2003.
2. Buyya, R. and Dastjerdi, A. V., Internet of Things: Principles and Paradigms, Morgan Kaufmann, 2016.
3. Zurawski, R. (Ed.), Industrial Communication Technology Handbook, 2nd Edition, CRC Press, 2014.
4. Benaben, F. and Stuber, W. (Eds.), Interoperability in Digital Industry, Springer, 2021.

Other useful resource(s):

1. NPTEL – Industrial Automation (IIT Kharagpur): https://nptel.ac.in/courses/10710506.
2. NPTEL – Internet of Things (IIT Bombay): https://nptel.ac.in/courses/108101093
3. Siemens Automation Learning Portal: https://support.industry.siemens.com
4. Eclipse IoT (Open-source IoT Frameworks): https://iot.eclipse.org
5. ThingsBoard IoT Platform (Open-source): https://thingsboard.io
6. IEC 61131-3 PLC Programming Reference: https://plcopen.org
7. Papers With Code – IoT and Edge Computing: https://paperswithcode.com

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	2	1	2	1	2	1.7
CO-2	3	2	2	2	3	1	1	1	2	1	2	1.8
CO-3	3	3	3	2	3	1	1	1	2	2	2	2.1
CO-4	3	3	2	2	3	1	1	1	2	2	2	2.0
CO-5	2	3	2	3	2	2	2	2	2	2	2	2.2
Average	**2.8**	**2.6**	**2.2**	**2.0**	**2.6**	**1.2**	**1.4**	**1.2**	**2.0**	**1.6**	**2.0**	