THE ASSAM
ROYAL GLOBAL UNIVERSITY
GUWAHATI

Royal School of Information Technology (RSIT)

Course Structure & Syllabus

(Based on National Education Policy 2020)

For

Master of Computer Applications

W.E.F

AY: 2025-2026

TABLE OF CONTENTS

Section	Header	Page no.
1	Overview	3
2	Award of Degree	10
3	Credit, Credit Points & Credit hours for different types of courses	11
4	Level of Courses	13
5	Graduate Attributes & Learning Outcomes	14
6	Course Structure and Syllabus of the Framework	21

Section 1:
Overview

1.1. Introduction

India is one of the fastest-growing economies globally, with knowledge creation and research playing a pivotal role in sustaining this momentum. As the nation aspires to establish itself as a leading knowledge society and one of the largest economies, there is an urgent need to expand research capabilities and outputs across disciplines.

At Royal Global University, we align ourselves with this national vision by fostering a robust research and innovation ecosystem, nurturing a vast talent pool that is critical for achieving these ambitious goals.

The National Education Policy (NEP) 2020 emphasizes the transformation of higher education to support India's transition to a knowledge-driven economy. Key initiatives such as multidisciplinary education with multiple entry and exit options, undergraduate research opportunities, and a learning outcomes-based curriculum are at the forefront of this transformation.

The postgraduate (PG) programmes at Royal Global University are designed to advance students' expertise in their chosen fields and equip them for higher research pursuits. These programs provide the advanced knowledge and specialized skills necessary for students to evolve from learners to innovators, contributing meaningfully to the nation's knowledge economy.

In line with NEP 2020, Royal Global University offers restructured degree programs to provide flexible and holistic education. The policy envisions undergraduate programmes with various certification options, including:

- A UG certificate after completing 1 year of study,
- A UG diploma after 2 years,
- A Bachelor's degree after a 3-year programme, or
- A preferred 4-year multidisciplinary Bachelor's degree, offering students the opportunity to explore holistic and multidisciplinary education alongside their chosen major and minors.

Similarly, postgraduate programmes at Royal Global University are designed with flexibility to cater to diverse academic and professional aspirations, fostering a new generation of knowledge creators who will shape India's future as a global leader.

Royal Global University remains committed to empowering students and creating an educational environment that embodies the principles of NEP 2020, driving innovation and excellence in higher education.

1.2. Recommendations of NEP 2020 Pertinent to Postgraduate Education

- A 2-year PG programme may be offered, with the second year exclusively dedicated to research for students who have completed a 3-year Bachelor's programme.
- For students who have completed a 4-year Bachelor's programme with Honours or Honours with Research, a 1-year PG programme could be introduced.
- An integrated 5-year Bachelor' s/Master's programme may also be offered.
- Universities are encouraged to provide PG programs in core areas such as Machine Learning, multidisciplinary fields like AI + X, and professional domains such as healthcare, agriculture, and law.
- A National Higher Education Qualifications Framework (NHEQF) will define higher education qualifications in terms of learning outcomes. The PG programme levels will correspond to Levels 6, 6.5, and 7 under the NHEQF.
- The PG framework must align with the National Credit Framework (NCrF) to facilitate the creditization of learning, including the assignment, accumulation, storage, transfer, and redemption of credits, subject to appropriate assessment.
- For a 2-year (4-semester) MCA PG program is at at level 6.5 of the NHEQF requires a 3-year Bachelor's degree with a minimum of 120 credits.

1.3. About MCA Course

The Master of Application (MCA) in the Royal School of Information Technology program at the Assam Royal Global University is designed to provide advanced knowledge and skills in various domains of computer science, aligning with the guidelines of the National Credit Framework (NCrF).

1.3.1. Program Objectives

- *Advanced Knowledge Acquisition:* Equip students with an in-depth understanding of core and emerging areas in computer science, such as Artificial Intelligence, Data Analytics, Internet of Things, and Networking.
- *Research and Innovation:* Foster a research-oriented mindset, encouraging students to undertake innovative projects that address real-world challenges.
- Skill Development: Enhance practical skills through hands-on experience, ensuring graduates are proficient in modern tools and technologies relevant to the industry.
- *Interdisciplinary Approach:* Promote an interdisciplinary learning environment, enabling students to integrate knowledge from various fields to develop

comprehensive solutions.

1.3.2. Alignment with National Credit Framework (NCrF)

By the NCrF, the program ensures a holistic and flexible education system by:

- *Credit Assignment and Accumulation:* Implementing a standardized credit system where 30 notional learning hours equate to one credit, facilitating the accumulation and transfer of credits across different educational levels and institutions.
- *Multiple Entry and Exit Options:* Providing students with the flexibility to enter and exit the program at various stages, with appropriate certification, diploma, or degree awarded based on the credits earned, thereby accommodating diverse learning needs and career paths.
- *Integration of Academic and Vocational Education:* Bridging the gap between theoretical knowledge and practical application by incorporating skill-based modules and experiential learning opportunities into the curriculum.

1.3.3. Program Structure

The MCA program spans two years, divided into four semesters, with a total of 80 credits. Each semester comprises core courses, electives, and project work, designed to provide both breadth and depth in the subject matter. Specializations offered include:

- Artificial Intelligence: Focusing on machine learning, neural networks, and intelligent systems.
- Data Analytics: Emphasizing data mining, big data technologies, and statistical analysis.
- Image Processing: Covering sensor networks, IoT architectures, and applications.

1.3.4. Learning Outcomes

Graduates of the program will:

- Demonstrate advanced knowledge in specialized areas of computer science and engineering.
- Exhibit proficiency in research methodologies, contributing to technological advancements.
- Apply interdisciplinary approaches to solve complex engineering problems.
- Possess the skills and knowledge required for successful careers in academia, industry, or entrepreneurship.

By integrating the principles of the National Credit Framework, the Assam Royal Global University's MCA program ensures a comprehensive, flexible, and industry-relevant education, preparing students to excel in the dynamic field of computer science and engineering.

1.4. Vision

To offer globally integrated opportunities in the domain of computer science and engineering, fostering the development of students as global citizens with the skills and perspectives needed to thrive in an interconnected world.

1.5. Mission

- To achieve academic excellence in computer science education through a dynamic curriculum, research-driven initiatives, and industry-aligned programs.
- To instil ethical values and a spirit of community service
- To give back responsible leaders equipped to drive positive change and innovation in the global technological landscape.

1.6. Credits in the Indian Context

1.6.1. Choice Based Credit System (CBCS)

Under the CBCS system, the requirement for awarding a degree or diploma or certificate is prescribed in terms of the number of credits to be earned by the students. This framework is being implemented in several universities across States in India. The main highlights of CBCS are as follows:

- The CBCS provides flexibility in designing curriculum and assigning credits based on the course content and learning hours.
- The CBCS provides for a system wherein students can take courses of their choice, learn at their own pace, undergo additional courses and acquire more than the required credits, and adopt an interdisciplinary approach to learning.
- CBCS also provides an opportunity for vertical mobility to students from a bachelor's degree program to master's and research degree programs.

1.6.2. Academic Credit

An academic credit is a unit by which a course is weighted. It is fixed by the number of hours of instruction offered per week. As per the National Credit Framework:

1 Credit = 30 NOTIONAL CREDIT HOURS (NCH)

Yearly Learning Hours = 1200 Notional Hours (@40 Credits x 30 NCH)

30 Notional Credit Hours		
Lecture/Tutorial	**Practicum**	**Experiential Learning**
1 Credit = 15 -22 Lecture Hours	10-15 Practicum Hours	0-8 Experiential Learning Hours

1 Hr. Lecture (L) per week 1 credit	1 credit
1 Hr. Tutorial (T) per week	1 credit
1 Hr. Practical (P) per week	0.5 credits
2 Hours Practical (Lab) per week	1 credit

1.6.3. Course of Study

Couse of study indicates pursuance of study in a particular discipline/programme. Discipline/Programmes shall offer Professional Core Courses, Professional Elective Courses relevant to chosen specialization, Project Dissertation, and Summer Training/ Internship.

1.6.3.1. Disciplinary Major/ Professional Core Courses

Professional core courses in M. Tech. Programs are those that directly relate to the specific field of engineering in which a student is majoring. These courses delve deep into the foundational principles, theories, and practical applications of the chosen engineering discipline. These courses focus on specific areas of specialization. Many professional core courses include laboratory work and design projects to provide students with hands-on experience and practical skills. In laboratory sessions, students may conduct experiments to reinforce theoretical concepts and develop their technical skills. Design projects challenge students to apply their knowledge to solve real-world engineering problems and to work collaboratively in teams.

1.6.3.2. Disciplinary Minor/ Professional Elective Courses

These subjects are offered to allow students to tailor their education to align with their interests, career goals, and emerging industry trends within their chosen engineering discipline. These courses allow students to delve deeper into specific areas of specialization or to explore interdisciplinary topics that complement their core engineering curriculum. By offering a range of professional elective courses, students are empowered to customize their education according to their individual interests and career aspirations. These elective courses complement the core engineering curriculum and enable students to develop specialized expertise, practical skills, and professional competencies that enhance their competitiveness in the job market and prepare them for future leadership roles in their field.

1.6.3.3. Dissertation

Students need to do a mini project and publish a conference paper.

1.6.3.4. *Industrial training / Internship*

Students need to undergo a minimum of 6 months of mandatory internship during their course of study, which is a total of 20 credits, and will be evaluated towards the end of the 6th semester. The students The intention is induction into actual work situations. All students must undergo internships / Apprenticeships in a firm, industry, or organization or Training in labs with faculty and researchers in their own or other HEIs/research institutions during the summer/winter term. Students should take up opportunities for internships with local industry, business organizations, health and allied areas, local governments (such as panchayats, municipalities), Parliament or elected representatives, media organizations, artists, crafts persons, and a wide variety of organizations so that students may actively engage with the practical side of their learning and, as a by-product, further improve their employability. Students who wish to exit after the first two semesters will undergo a 4-credit work-based learning/internship during the summer term to get a UG Certificate.

- *Community engagement and service:* The curricular component of 'community engagement and service' seeks to expose students to the socio-economic issues in society so that the theoretical learnings can be supplemented by actual life experiences to generate solutions to real-life problems. This can be part of a summer term activity or part of a major or minor course, depending upon the major discipline.
- *Field-based learning/minor project:* The field-based learning/minor project will attempt to provide opportunities for students to understand the different socio-economic contexts. It will aim at giving students exposure to development-related issues in rural and urban settings. It will provide opportunities for students to observe situations in rural and urban contexts and to observe and study actual field situations regarding issues related to socioeconomic development. Students will be given opportunities to gain a first-hand understanding of the policies, regulations, organizational structures, processes, and programmes that guide the development process. They would have the opportunity to gain an understanding of the complex socio-economic problems in the community and the innovative practices required to generate solutions to the identified problems. This may be a summer term project or part of a major or minor course, depending on the subject of study.

1.6.3.5. *Experiential Learning*

One of the most unique, practical & beneficial features of the National Credit Framework is the assignment of credits/credit points/ weightage to the experiential learning, including relevant experience and professional levels acquired/ proficiency/ professional levels of a learner/student. Experiential learning is of two types:

a. Experiential learning as part of the curricular structure of academic or vocational program. E.g., projects/OJT/internship/industrial attachments, etc. This could be either within the Program internship/ summer project undertaken relevant to the program being studied or as a part-time employment (not relevant to the program being studied- up to certain NSQF level only). In cases where experiential learning is a part of the curricular structure, the credits would be calculated and assigned as per the basic principles of NCrF, i.e., 40 credits for 1200 hours of notional learning.

b. Experiential learning as active employment (both wage and self) post-completion of an academic or vocational program. This means that the experience attained by a person after undergoing a particular educational program shall be considered for the assignment of credits. This could be either Full or part-time employment after undertaking an academic/vocational program.

In cases where experiential learning is a part of employment, the learner would earn credits as weightage. The maximum credit points earned in this case shall be double the credit points earned concerning the qualification/ course completed. The credit earned and assigned by relevant experience would enable learners to progress in their career through the work hours put in during a job/employment.

Section 2

Award of Degree

The structure and duration of Postgraduate programmes of study offered by the University as per NEP 2020 include:

2.1. Postgraduate programmes of 4-year duration with Single Major, with multiple entry and exit options, with appropriate certifications:

2.1.1. PG Diploma: Students who opt to exit after completion of the first year and have secured 40 credits will be awarded a PG Diploma certificate if, in addition, they complete one vocational course of 4 credits during the summer vacation of the first year.

2.1.2. MCA: A Master of Computer Applications degree in the major discipline will be awarded to those who complete a two-year degree program with 80 credits and have satisfied the credit requirements along with a mention of the specialized domain like M. Tech-CSE in Artificial Intelligence, etc.

Section 3

Credit, Credit Points & Credit hours for different types of courses

3.1. Introduction:

'Credit' is recognition that a learner has completed a prior course of learning, corresponding to a qualification at a given level. For each such prior qualification, the student would have put in a certain volume of institutional or workplace learning, and the more complex a qualification, the greater the volume of learning that would have gone into it. Credits quantify learning outcomes that are subject achieving the prescribed learning outcomes to valid, reliable methods of assessment.

The ***credit points*** will give the learners, employers, and institutions a mechanism for describing and comparing the learning outcomes achieved. The credit points can be calculated as credits attained multiplied by the credit level.

The workload relating to a course is measured in terms of credit hours. A credit is a unit by which the coursework is measured. It determines the number of hours of instruction required per week over a semester (minimum 15 weeks).

Each course may have only a lecture component, a lecture and tutorial component, a lecture and practicum component, a lecture, tutorial, and practicum component, or only a practicum component.

A course can have a combination of ***lecture credits, tutorial credits, practicum credits, and experiential learning credits.*** The following types of courses/activities constitute the programmed of study. Each of them will require a specific number of hours of teaching/guidance and laboratory/studio/workshop activities, field-based learning/projects, internships, and community engagement and service.

- **Lecture courses:** Courses involving lectures relating to a field or discipline by an expert or qualified personnel in a field of learning, work/vocation, or professional practice.
- **Tutorial courses:** Courses involving problem-solving and discussions relating to a field or discipline under the guidance of qualified personnel in a field of learning, work/vocation, or professional practice. Should also refer to the Remedial Classes, flip classrooms and focus on both Slow and Fast Learners of the class according to their merit.

- **Practicum or Laboratory work:** A course requiring students to participate in a project or practical or lab activity that applies previously learned/studied principles/theory related to the chosen field of learning, work/vocation, or professional practice under the supervision of an expert or qualified individual in the field of learning, work/vocation or professional practice.
- **Internship:** A course requiring students to participate in a professional activity or work experience or cooperative education activity with an entity external to the education institution, normally under the supervision of an expert of the given external entity. A key aspect of the internship is induction into actual work situations. Internships involve working with local industry, government or private organizations, business organizations, artists, crafts persons, and similar entities to provide opportunities for students to actively engage in on-site experiential learning.
- **Field practice/projects:** Courses requiring students to participate in field-based learning/projects generally under the supervision of an expert of the given external entity.

Table 2: Course-wise Distribution of Credits

Sl. No	***Category***	***Abbreviation***	***Credit Breakup***
1	Professional core courses	PCC	35
2	Professional Elective courses relevant to chosen specialization/branch	PEC	19
3	Project work, seminar, and internship in industry or elsewhere	PROJ	26
		Total	**80**

Section 4

Level of Courses

4.1 NHEQF levels:

The NHEQF levels represent a series of sequential stages expressed in terms of a range of learning outcomes against which typical qualifications are positioned/located. Postgraduate programmes fall between Level 6.5 and Level 7, as outlined in the NHEQF. The framework ensures that PG students acquire both depth in their subject knowledge and the ability to apply their learning to complex, real-world challenges.

Table: 4.1: NHEQF Levels

NHEQF level	Examples of higher education qualifications located within each level	Credit Requirements
Level 4.5	Undergraduate Certificate. Programme duration: First year (first two semesters) of the undergraduate programme, followed by an exit 4-credit skills-enhancement course(s).	40
Level 5	Undergraduate Diploma. Programme duration: First two years (first four semesters) of the undergraduate programme, followed by an exit 4-credit skills-enhancement course(s) lasting two months.	80
Level 5.5	Bachelor's Degree. Programme duration: First three years (Six semesters) of the four-year undergraduate programme.	120
Level 6	Bachelor's Degree (Honours/ Honours with Research). Programme duration: Four years (eight semesters).	160
Level 6	Post-Graduate Diploma. Programme duration: One year (two semesters) for those who exit after successful completion of the first year (two semesters) of the 2-year master's programme	160
Level 6.5	Master's degree. Programme duration: Two years (four semesters) after obtaining a 3-year Bachelor's degree (e.g., B.A., B.Sc., B.Com, etc.).	80
Level 6.5	Master's degree. Programme duration: One year (two semesters) after obtaining a 4-year Bachelor's degree (Honours/ Honours with Research) (e.g., B.A., B.Sc., B.Com. etc.).	40
Level 7	Master's degree. (e.g., M.E./M.Tech. etc.) Programme duration: Two years (four semesters) after obtaining a 4-year Bachelor's degree. (e.g., B.E./B.Tech. etc.)	80
Level 8	Doctoral Degree	Credits for course work, Thesis, and published work

Section 5

Graduate Attributes & Learning Outcomes

5.1 Introduction

As per the NHEQF, each student, on completion of a programme of study, must possess and demonstrate the expected ***Graduate Attributes*** acquired through one or more modes of learning, including direct in-person or face-to-face instruction, online learning, and hybrid/blended modes. The graduate attributes indicate the quality and features or characteristics of the graduate of a programme of study, including learning outcomes relating to the disciplinary area(s) relating to the chosen field(s) of learning and generic learning outcomes that are expected to be acquired by a graduate on completion of the programme(s) of study.

5.2 Graduate Attributes

Qualifications that signify completion of the postgraduate degree are awarded to students who:

GA1: Have demonstrated knowledge and understanding that is founded upon and extends and/or enhances that typically associated with the first cycle, and that provides a basis or opportunity for originality in developing and/or applying ideas, often within research context.

GA2: Can apply their knowledge and understanding and problem-solving abilities in new or unfamiliar environments within broader (or multidisciplinary) contexts related to their field of study.

GA3: Have the ability to integrate knowledge and handle complexity, and formulate judgments with incomplete or limited information, but that includes reflecting on social and ethical responsibilities linked to the application of their knowledge and judgments.

GA4: can communicate their conclusions, and the knowledge and rationale underpinning these, to specialist and non-specialist audiences clearly and unambiguously.

GA5: Have the learning skills to allow them to continue to study in a manner that may be largely self-directed or autonomous.

The PG degree (e.g., M.C.A., M.Com., M.Sc., etc.) will be awarded to students who have demonstrated the achievement of the outcomes located at level 6.5 on the NHEQF. Refer to Table 5.1.1

Table 5.1.1

Element of the descriptor	**NHEQF level descriptors** *The graduates should be able to demonstrate the acquisition of:*
Knowledge and understanding	• advanced knowledge about a specialized field of enquiry with a critical understanding of the emerging developments and issues relating to one or more fields of learning, • advanced knowledge and understanding of the research principles, methods, and techniques applicable to the chosen field(s) of learning or professional practice, • Procedural knowledge required for performing and accomplishing complex and specialized and professional tasks relating to teaching, research, and development.
General, technical, and professional skills required to perform and accomplish tasks	• Advanced cognitive and technical skills required for performing and accomplishing complex tasks related to the chosen fields of learning. • Advanced cognitive and technical skills required for evaluating research findings and designing and conducting relevant research that contributes to the generation of new knowledge. • Specialized cognitive and technical skills relating to a body of • Knowledge and practice to analyze and synthesize complex information and problems.
Application of knowledge and skills	• Apply the acquired advanced theoretical and/or technical knowledge about a specialized field of enquiry or professional practice and a range of cognitive and practical skills to identify and analyse problems and issues, including real-life problems, associated with the chosen fields of learning. • apply advanced knowledge relating to research methods to carry out research and investigations to formulate evidence-based • solutions to complex and unpredictable problems.

Generic learning outcomes	**Effective Communication and Presentation** • Listen attentively, analyze texts and research papers, and present complex information clearly to diverse audiences. • Communicate technical information, research findings, and explanations in a structured manner. • Concisely discuss the relevance and applications of research findings in the context of emerging developments and issues. **Critical Thinking and Analytical Skills** • Evaluate evidence reliability, identify logical flaws, and synthesize data from multiple sources to draw valid conclusions. • Support arguments with evidence, address opposing viewpoints, and critique the reasoning of others. **Self-Directed Learning and Professional Development** • Address personal learning needs in chosen fields of study, work, or professional practice. • Pursue self-paced learning to enhance knowledge and skills, particularly for advanced education and research. • **Research Design and Methodology** • Define and articulate research problems, formulate hypotheses, and design relevant research questions. • Develop appropriate tools and techniques for data collection and analysis. • Used statistical and analytical methods to interpret data and establish cause-and-effect relationships. **Research Execution and Ethics** • Plan, conduct, and report investigations while adhering to ethical standards in research and practice. • Apply research ethics rigorously in fieldwork and personal research activities. **Problem-Solving and Decision-Making** • Make informed judgments and decisions based on empirical evidence and analysis to solve real-world problems. • Take responsibility for individual and group actions in • generating solutions within specific fields of study or professional practice.

Constitutional, humanistic, ethical, and moral values	• embrace and practice constitutional, humanistic, ethical, and moral values in one’s life, • adopt objective and unbiased actions in all aspects of work related to the chosen fields/subfields of study and professional practice, • participate in actions to address environmental protection and sustainable development issues, • support relevant ethical and moral issues by formulating and presenting coherent arguments, • Follow ethical principles and practices in all aspects of research and development, including inducements for enrolling participants, avoiding unethical practices such as fabrication, falsification or misrepresentation of data or committing plagiarism.
Employability & job-ready skills, entrepreneurship skills and capabilities/qualities and mindset	• Adapting to the future of work and responding to the demands of the fast pace of technological developments and innovations that drive the shift in employers’ demands for skills, particularly with respect to the transition towards more technology-assisted work involving the creation of new forms of work and rapidly changing work and production processes. • Exercising full personal responsibility for the output of my work as well as for group/team outputs and for managing work that is • Complex and unpredictable, requiring new strategic approaches.

5.3 Programme Learning Outcomes (PLO)

The term ‘programme’ refers to the entire scheme of study followed by learners leading to a qualification. Individual programmes of study will have defined learning outcomes that must be attained for the award of a specific certificate/diploma/degree. Programme Learning Outcomes describe what students are expected to know or be able to do by the time of graduation. PLOs are statements about the knowledge, skills, and attitudes (attributes) the graduate of a formal engineering program should have. PLOs deal with the general aspect of graduation for a particular program and the competencies and expertise a graduate will possess after completion of the program. Apply the knowledge of mathematics and computing fundamentals to various real-life applications for any given requirement. Design and develop applications to analyse and solve all computer science-related problems. This is accomplished through the following learning goals and objectives:

- **PO1- Knowledge of mathematics and computing fundamentals:** Apply the knowledge of mathematics and computing fundamentals to various real-life applications for any given requirement.
- **PO2- Design and develop applications:** Design and develop applications to analyze and solve all computer science-related problems.
- **PO3- Effective Communication:** Students will use various forms of business communication, supported by effective use of appropriate technology, logical reasoning, and articulation of ideas. Graduates are expected to develop effective oral and written communication, especially in business applications, with the use of appropriate technology (business presentations, digital communication, social network platforms, and so on).
- **PO4- Leadership and Teamwork:** Students will acquire skills to demonstrate leadership roles at various levels of the organization and leading teams. Graduates are expected to collaborate and lead teams across organizational boundaries and demonstrate leadership qualities, maximizing the usage of diverse skills of team members in the related context.
- **PO5- Global Exposure and Cross-Cultural Understanding:** The Graduate will be able to demonstrate a global outlook with the ability to identify aspects of the global business and Cultural Understanding.
- **PO6- Integrate and apply efficient tools.** Integrate and apply the contemporary IT tools efficiently to all computer applications.
- **PO7- Designing innovative methodologies:** Create and design innovative methodologies to solve complex problems for the betterment of society.
- **PO8- Applying inherent skills:** Apply the inherent skills with absolute focus to function as a successful entrepreneur.
- **PO9- Social Responsiveness and Ethics:** Students will demonstrate responsiveness to contextual social issues/ problems and explore solutions, understanding ethics and resolving ethical dilemmas. Demonstrate awareness of ethical issues and distinguish ethical and unethical behaviour.

5.4 Programme Educational Objectives (PEOs)

The Programme Educational Objectives (PEOs) are defined and developed for each program with the consultation and involvement of various stakeholders such as management, students, industry, regulating authorities, alumni, faculty, and parents. Their interests, social relevance, and contributions are taken into account in defining and developing the PEOs. The Program Educational Objectives (PEOs) of the Computer Science and Engineering are listed below:

- **PEO1**: Independently design and develop computer software systems and products based on sound theoretical principles and appropriate software development skills.
- **PEO2**: Demonstrate knowledge of technological advances through active participation in life-long learning.
- **PEO3**: Accept to take up responsibilities upon employment in the areas of teaching, research, and software development
- **PEO4**: Exhibit technical communication, collaboration, and mentoring skills and assume roles both as team members and as team leaders in an organization.

5.5 Programme-Specific Outcomes (PSOs)

- **PSO1:** Analyze and understand the need for research and development, Intellectual property rights, patents, and plagiarism checking tools.
- **PSO2:** Ability to understand the need for human values and professional ethics while publishing research papers, writing and developing research projects, research grants, books, and dissertations.
- **PSO3:** Pursue a career in software development, entrepreneurship, database administration, network and cyber security, artificial intelligence, machine learning, higher studies, teaching, or quality testing using available CASE tools.

5.6 The Qualification Specifications

The levels of PG programs as per the NHEQF and Royal Global University are summarized in Table 5.2

Table 5.2:

Level	Credits	Qualification	Credit Requirement Per Year	Credit Points	Total Notional Learning Hours
6	160	1–yr P.G. Diploma	40	240	1200
6.5	160	1-Year PG after a 4-year UG	40	260	1200
6.5	120	2-Year PG after a 3-year UG	40	260	1200
7	160	2-Year PG after a 4-year UG such as B.E., B. Tech. etc	40	280	1200

5.7 Credit Distribution for 2-year PG

Table: 5.3

Curricular Components		**PG Programme (one year) for 4-year UG (Hons. /Hons. with Research)** **Minimum Credits**			
		Course Level	**Coursework**	**Research thesis/project /Patent**	**Total Credits**
PG Diploma		400	40+4(MOOCS)	--	40+4
1st Year (1st & 2nd Semester)		400 500	24 16	--	40+4
Students who exit at the end of 1st year shall be awarded a Postgraduate Diploma.					
2nd Year (3rd & 4th Semester)	Coursework & Research	500	20 +4(MOOCS)	20	40+4
	Coursework (or)	500	40+4(MOOCS)	--	40+4
	Research		4(MOOCS)	40	40+4

- *Exit Point: For those who join 2-year PG programmes, there shall only be one exit point. Students who exit at the end of 1st year shall be awarded a Postgraduate Diploma.*

5.8 Course Levels

- **400-499**: Advanced courses which would include courses taught with practicum, seminar-based courses, term papers, research methodology, advanced laboratory experiments/software training, research projects, hands-on-training, internship / apprenticeship projects at the under-graduate level or First year post-graduate theoretical and practical courses.
- **500-599:** For students who have graduated with a 4-year bachelor's degree. It provides an opportunity for original study or investigation in the major or field of specialization on an individual and more autonomous basis at the postgraduate. These Courses can be placed at first-year master's degree level for 2-year Master's degree programme
- **600-699**: Courses for 2nd year of 2-years master's or 1-year Master's degree programme

Section 6

Course Structure and Syllabus of the Framework

6.1 Course Structure of MCA

Proposed Course Structure of 2-year PG Programme RSIT :
Master of Computer Applications

Semester-I					
S.N	**Subject Code**	**Names of subjects**	**LEVEL**	**C**	**L-T-P**
1	CAP054C101	Mathematical Foundation for Computer Science	400	4	3-1-0
2	CAP054C102	Advanced Data Structures and Algorithms	499	3	3-0-0
3	CAP054C103	Database Management and Data Warehousing	400	3	3-0-0
4	CAP054C112	Advanced Data Structures and Algorithms Lab	499	1	0-0-2
5	CAP054C113	Database Management and Data Warehousing Lab	400	1	0-0-2
6	CAP054C104 / CAP054C105/ CAP054C106/	Artificial Intelligence/ Data Mining / Digital Image Processing	500	4	4-0-0
7	CAP054C107 / CAP054C108 / CAP054C109/	Introduction to Deep Learning / Statistical Computing /Machine Learning Applications	500	4	4-0-0
8	MOOCS	One 8-Week Course from SWAYAM /MOOCS as per the Department Directives	400	2	-
TOTAL				**22**	

Semester-II					
S.N	**Subject Code**	**Names of subjects**	**LEVEL**	**C**	**L-T-P**
1	CAP054C201	Web Technology	499	3	3-0-0
2	CAP054C202	System Administration	499	3	3-0-0
3	CAP054C203	Internet Protocols and Network Design	500	3	3-0-0
4	CAP054C211	Web Technology Lab	499	1	0-0-2
5	CAP054C212	System Administration Lab	499	1	0-0-2
6	CAP054C213	Internet Protocols and Network Design Lab	500	1	0-0-2
7	CAP054C204/ CAP054C205/ CAP054C206	Natural Language Processing / Big Data Analytics / Remote Sensing and GIS	500	4	4-0-0
8	CAP054C207/ CAP054C208/ CAP054C209	Computer Vision / Cloud Computing for Big Data/ Machine Vision	500	4	4-0-0
6	MOOCS	One 8-Week Course from SWAYAM /MOOCS as per the Department Directives	**400**	**2**	**-**
TOTAL				**22**	

2nd Year Course Structure (Course Work + Research)

Semester-III					
S.N	**Subject Code**	**Names of subjects**	**Level**	**C**	**L-T-P**
1.	CAP054C301	Software Project Management	600	3	3-0-0
2.	CAP054C302	Network Security and Cryptography	600	3	3-0-0
3.	CAP054C311	Software Project Management Lab	600	1	0-0-2
4.	CAP054C312	Network Security and Cryptography Lab	600	1	0-0-2
5.	CAP054C303/ CAP054C304	Internet of Things /Embedded Systems	600	4	3-1-0
5.	MOOCS	One 8-Week Course from SWAYAM /MOOCS as per the Department Directives	600	2	-
6.	CAP054C321	Dissertation-I	600	8	-
TOTAL				**22**	

Semester-IV					
S.N	**Subject Code**	**Names of subjects**	**Level**	**C**	**L-T-P**
1.	CAP054C401/ CAP054C402	Quantum Computing / Blockchain Technologies	600	4	3-1-0
2.	CAP054C403/ CAP054C404	Soft Computing / Edge Computing	600	4	3-1-0
3.	**MOOCS**	One 8-Week Course from SWAYAM /MOOCS as per the Department Directives	**600**	**2**	-
4.	CAP054C421	Dissertation-II	600	12	-
TOTAL				**22**	

2nd Year Course Structure (Only Research Work)

Semester-III					
S.N	**Subject Code**	**Names of subjects**	**Level**	**C**	**L-T-P**
1.	MOOCS	One 8-Week Course from SWAYAM /MOOCS as per the Department Directives	600	2	-
2.	CAP054C321	Dissertation-I	600	20	-
TOTAL				**22**	

Semester-IV					
S.N	**Subject Code**	**Names of subjects**	**Level**	**C**	**L-T-P**
1.	**MOOCS**	One 8-Week Course from SWAYAM /MOOCS as per the Department Directives	**600**	**2**	-
2.	CAP054C421	Dissertation-II	600	20	-
TOTAL				**22**	

6.1 Detailed Syllabus of 1st Semester

Paper I/Subject Name: Mathematical Foundations of Computer Science		**Subject Code: CAP054C101**
L-T-P-C – 3-1-0-4	**Credit Units: 03**	**Scheme of Evaluation: T**

Objective:

The objectives of the course are to enable students to develop a strong mathematical foundation for computing and problem-solving for concepts like logic, set theory, relations, functions, graph theory, combinatorics, number theory, probability, and statistical techniques.

Prerequisites: Discrete Mathematics, Linear Algebra, Probability and Statistics, Basic Understanding of Algorithms.

Course Outcomes

On successful completion of the course, the students will be able to:		
SI No	**Course Outcome**	**Blooms Taxonomy Level**
CO 1	**Understand** number theory in cryptographic algorithms.	**BT 2**
CO 2	**Apply** set theory, logic, and proof techniques in computing problems	**BT 3**
CO 3	**Analyze** and **evaluate** probabilistic models in computing, machine learning, and cryptography.	**BT 4 & 5**
CO 4	**Solve** problems related to formal languages, automata, and computational complexity.	**BT 6**

Detailed Syllabus:

Modules	**Topics**	**Course Contents**	**Hours**
I.	**Logic, Set Theory**	Propositional Logic and Predicate Logic, Logical Connectives, Truth Tables, Normal Forms, Logical Inference, Resolution, Proof Techniques, Set Theory and Relations, Sets, Operations, Power Sets, Types of Relations: Reflexive, Symmetric, Transitive, Equivalence Relations, Functions: Injective, Surjective, Bijective, Mathematical Induction and Recursion, Inductive Proofs, Recursive Definitions and Structural Induction	**22**
II.	**Combinatorics, Graph Theory & Number Theory**	Counting Principles, Permutations and Combinations, Pigeonhole Principle, Inclusion-Exclusion Principle, Graph Theory and Applications, Graph Representation: Adjacency Matrix, Adjacency List, Eulerian and Hamiltonian Graphs, Shortest Path Algorithms (Dijkstra, Floyd-Warshall), Planar Graphs and Graph Coloring, Number Theory and Applications, Divisibility, Prime Numbers, Congruences, Fermat's Theorem, Euler's Theorem, Modular Arithmetic and Cryptography	**22**
III.	**Probability, Statistics and Randomized Algorithms**	Probability Theory, Axioms of Probability, Conditional Probability and Bayes' Theorem, Random Variables and Expectation, Statistical Methods, Mean, Variance, Standard Deviation, Probability Distributions (Binomial, Poisson, Normal), Hypothesis Testing and Confidence Intervals, Randomized	**22**

		Algorithms and Markov Chains, Monte Carlo and Las Vegas Algorithms, Markov Chains and Applications	
IV	**Formal Languages, Automata, and Computational Complexity**	Formal Languages and Automata, Regular Expressions and Finite Automata, Pushdown Automata and Context-Free Grammars, Turing Machines and Computability, Turing Machine Models, Decidability and Undecidability, Computational Complexity, P, NP, NP-Complete, and NP-Hard Problems, Approximation Algorithms and Hardness of Approximation	**22**
		TOTAL	**88**

Mathematical Foundations of Computer Science Practice Sessions

Minimum 10 Laboratory experiments based on the following-

- Implement propositional logic in Prolog.
- Write Python programs for set operations and relation properties.
- Implement graph traversal algorithms (BFS, DFS).
- Implement modular exponentiation for cryptography.
- Simulate randomized algorithms in Python.
- Implement Bayes' Theorem for spam filtering.
- Implement finite automata for pattern matching.
- Simulate Turing Machines using Python.

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4*22 NCH = 88 NCH	1*30 NCH = 30 NCH	4*8 = 32 NCH (Problem Solving, Internship, Seminar, Case Study, Discussion)

Text Books:

1. Discrete Mathematics and Its Applications*,* Kenneth H. Rosen, 7th Edition, 2017, McGraw-Hill
2. *Introduction to Algorithms,* Cormen, Leiserson, Rivest, & Stein (CLRS), 3rd Edition, 2009, MIT Press
3. *Introduction to Automata Theory, Languages, and Computation,* John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, 3rd Edition, 2008, Pearson

Reference Books:

1. Sheldon Ross, *Introduction to Probability Models*, 11th Edition, 2014, Academic Press
2. Richard Johnsonbaugh, *Discrete Mathematics*, 7th Edition, 2014, Pearson
3. Michael Sipser, *Introduction to the Theory of Computation*, 3rd Edition, 2014, Cengage
4. Narsingh Deo, *Graph Theory with Applications to Engineering and Computer Science,* New Edition, 1979, PHI

Paper II/Subject Name: Advanced Data Structure and Algorithms		**Subject Code: CAP054C102**
L-T-P-C – 3-0-0-3	**Credit Units: 03**	**Scheme of Evaluation: T**

Objective:

This course aims to provide in-depth knowledge of complex data structures and advanced algorithms, focusing on optimization techniques, real-world applications, and competitive programming skills.

Prerequisites: Basic Data Structures (Arrays, Linked Lists, Stacks, Queues), Knowledge of Sorting & Searching Algorithms, Basics of Graph Theory, and Recursion

Course Outcomes:

On successful completion of the course, the students will be able to:		
SI No	**Course Outcome**	**Blooms Taxonomy Level**
CO 1	**Define** and **demonstrate** how Data Structures work.	**BT 1 & 2**
CO 2	**Apply** the Data Structures concepts to solve various problems.	**BT 3**
CO 3	**Analyze** and debug the errors while writing the programs.	**BT 4**
CO 4	**Assess** and design a new algorithm to solve a new real-life problem	**BT 5**

Detailed Syllabus:

Modules	**Topics**	**Course content**	**Periods**
I	**Advanced Data Structures**	Persistent Data Structures, Skip Lists, Self-balancing Trees (AVL, Red-Black, B-Trees, Splay Trees), Segment Trees, Fenwick Trees, Fibonacci Heaps, Graph Representations (Adjacency List, Adjacency Matrix, Incidence Matrix)	**22**
II	**Graph Algorithms**	Shortest Path Algorithms (Dijkstra, Bellman-Ford, Floyd-Warshall, Johnson's Algorithm), Minimum Spanning Tree (Kruskal, Prim's), Maximum Flow (Ford-Fulkerson, Edmonds-Karp), Eulerian and Hamiltonian Paths, Topological Sorting (Kahn's Algorithm, DFS-based approach)	**22**
III	**Advanced Algorithmic Techniques**	Divide & Conquer (Merge Sort, Quick Sort, Strassen's Matrix Multiplication, Closest Pair of Points), Greedy Algorithms (Huffman Coding, Activity Selection, Job Scheduling, Fractional Knapsack), Dynamic Programming (0/1 Knapsack, LCS, Floyd-Warshall, Matrix Chain Multiplication, Bellman-Ford), String Matching Algorithms - KMP, Rabin-Karp, Aho-Corasick)	**22**
IV	**Complexity & NP-Hard Problems**	Complexity Classes (P, NP, NP-Hard, NP-Complete), Reduction Techniques, Approximation Algorithms (Vertex Cover, Traveling Salesman Problem, Set Cover), Real-World Applications (AI, Bioinformatics, Game Theory, Computational Geometry,	**22**
Total			**88**

Subject Name:	**Advanced Data Structure and Algorithms Lab**	**Subject Code: CAP054C112**
L-T-P-C – 0-0-2-1	**Credit Units: 01**	**Scheme of Evaluation: P**

Total Lab Hours for the semester = 30 (2 hours per week)

Minimum 10 Laboratory experiments based on the following-

Experiment No.	**Title**	**Objective**
1	Implementation of Linked Lists (Singly, Doubly, Circular)	Understanding dynamic memory allocation and pointer manipulation.
2	Stack & Queue Implementation	Using arrays and linked lists to implement stack and queue operations.
3	Priority Queue & Heap Implementation	Understanding heap properties and implementing Min-Heap & Max-Heap.
4	Binary Search Tree (BST) Operations	Implementing insert, delete, and search operations in BST.
5	AVL Tree Implementation	Implementing AVL rotations (Left, Right, Left-Right, Right-Left) for self-balancing.
6	Graph Representations & Traversals	Implemented adjacency list/matrix and BFS, DFS traversals.
7	Dijkstra's Algorithm for Shortest Path	Implementing Dijkstra's algorithm for weighted graphs.
8	Floyd-Warshall Algorithm	Understanding and implementing all-pairs shortest paths.
9	Kruskal's & Prim's MST Algorithms	Implementing Minimum Spanning Tree algorithms.
10	Bellman-Ford Algorithm	Understanding negative-weight edge handling in shortest path problems.
11	Topological Sorting	Implementing Kahn's algorithm and DFS-based topological sort.
12	0/1 Knapsack Problem (Dynamic Programming)	Implementing a dynamic programming approach for knapsack optimization.
13	Longest Common Subsequence (LCS) Algorithm	Implementing dynamic programming-based LCS calculation.
14	String Matching Algorithms (KMP, Rabin-Karp)	Efficient pattern searching in text processing applications.
15	Hashing Techniques & Collision Resolution	Implementing various hashing methods (Chaining, Open Addressing).
16	Segment Trees	Implementing segment trees for range queries and modifications.
17	Fenwick Trees (Binary Indexed Trees)	Understanding how to perform efficient cumulative frequency calculations.
18	Graph Coloring Problem (Backtracking)	Implementing graph coloring to solve scheduling and register allocation problems.

19	Approximation Algorithms (Vertex Cover, TSP)	Implementing heuristic-based approximation for NP-hard problems.
20	Competitive Programming Challenge	Solving real-world problems using efficient data structures and algorithms.

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4 * 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH	8 * 4 NCH = 32 NCH (Problem Solving, Seminar, Case Study, Discussion, Internship, Projects)

Textbooks:

1. Introduction to Algorithms, Cormen, Leiserson, Rivest & Stein (CLRS), 3rd Edition
2. Introduction to Data Structure, Reema Thereja, Pearson 2020

Reference Books:

1. Algorithm Design, Jon Kleinberg & Eva Tardos
2. The Art of Computer Programming, Donald Knuth
3. Competitive Programming Handbook, Antti Laaksonen

Subject Name: Database Management Systems and Data warehousing		**Subject Code: CAP054C103**
L-T-P-C – 3-0-0-3	**Credit Units: 03**	**Scheme of Evaluation: T**

Objective:

To provide comprehensive knowledge of relational database concepts along with advanced topics in data warehousing. This course aims to teach students how to design, query, normalize, and manage databases, and to introduce modern warehousing techniques.

Prerequisites: Basic SQL and relational database design, Understanding of Normalization & Indexing, Fundamentals of Transaction Management

Course Outcomes:

On successful completion of the course, the students will be able to:		
SI No	**Course Outcome**	**Blooms Taxonomy Level**
CO 1	Understand fundamental concepts of database design and data models	**BT 1 & 2**
CO 2	Apply SQL for data retrieval, manipulation, and transactions	**BT 3**
CO 3	Analyze and optimize relational schema using normalization techniques	**BT 4**
CO 4	Design and implement data warehouses using dimensional modeling	**BT 5**

Detailed Syllabus:

Modules	Topics	Course content	Periods
I	**Fundamentals of DBMS**	Introduction to DBMS and RDBMS; data models (hierarchical, network, relational); entity-relationship (ER) model; keys and constraints; relational algebra and calculus; database architecture and design; schema, instance, and independence	**22**
II	**Structured Query Language & Normalization**	Basic and advanced SQL: DDL, DML, DCL, joins, subqueries, views, indexes, triggers, stored procedures. Integrity constraints and transaction control. Normalization: 1NF to BCNF, multi-valued and join dependencies, lossless decomposition	**22**
III	**Data Warehousing Concepts**	Introduction to data warehousing; data warehouse architecture; differences between OLTP and OLAP; data marts; schemas: star, snowflake, fact constellation; ETL process: data extraction, cleaning, loading; metadata and warehouse governance	**22**
IV	**OLAP and Warehouse Implementation**	OLAP operations: slicing, dicing, roll-up, drill-down; MOLAP, ROLAP, HOLAP architectures; indexing in DW; data cube computation; performance optimization; case studies in warehouse implementations; BI tools overview (e.g., Power BI, Tableau)	**22**
		Total	**88**

Subject Name: Database Management Systems and Data warehousing Lab **Subject Code: CAP054C113**

L-T-P-C – 0-0-2-1 **Credit Units: 01** **Scheme of Evaluation: P**

Total Lab Hours for the semester = 30 (2 hours per week)

Minimum 15 Laboratory experiments based on the following-

Experiment No.	**Title**	**Objective**
1	Advanced SQL Queries	Write and optimize complex SQL queries using Joins, Subqueries, and Aggregations.
2	Indexing and Performance Tuning	Implement indexing strategies and measure query performance improvements.
3	NoSQL Data Models	Design and implement NoSQL data models in MongoDB.
4	Replication and Sharding in MongoDB	Configure database replication and horizontal partitioning in MongoDB.
5	Query Execution Plan Analysis	Analyze query execution plans to optimize database queries.
6	Transaction Management	Implement ACID transactions and concurrency control in MySQL/PostgreSQL.

7	Distributed Databases	Set up a simple distributed database system and test query performance.
8	Graph Database Implementation	Store and query graph data using Neo4j and Cypher queries.
9	Cloud Database Deployment	Deploy and manage a relational database on AWS RDS or Google Cloud SQL.
10	Stream Processing with Apache Kafka	Implement real-time data streaming using Apache Kafka.
11	Two-Phase Commit Implementation	Simulate two-phase commit protocol for distributed transactions.
12	Database Security & SQL Injection Testing	Perform SQL Injection attacks and apply security patches.
13	Big Data Querying	Execute SQL queries over large datasets using Apache Hive or Google BigQuery.
14	Data Warehouse Implementation	Design a basic data warehouse schema and implement ETL pipelines.
15	Time-Series Databases	Implemented and queried time-series data using InfluxDB.
16	Full-Text Search in Databases	Implement full-text search indexing in PostgreSQL or Elasticsearch.
17	Cloud NoSQL Database Integration	Worked with Firebase Realtime Database and Firestore.
18	Performance Benchmarking	Compare performance differences between relational and NoSQL databases.

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
2 * 22 NCH = 44 NCH	2 * 15 NCH = 30 NCH	8 * 2 NCH = 16 NCH (Problem Solving, Seminar, Case Study, Discussion, Internship, Projects)

Textbook:

1. Fundamentals of Database Systems, Elmasri & Navathe, 7th Edition

Reference Books:

1. NoSQL Distilled, Pramod J. Sadalage & Martin Fowler
2. Hadoop: The Definitive Guide, Tom White
3. Graph Databases, Ian Robinson, Jim Webber
4. Database System Concepts, Silberschatz, Korth & Sudarshan

Subject Name: Artificial Intelligence		**Subject Code: CAP054C104**
L-T-P-C – 4-0-0-4	**Credit Units: 04**	**Scheme of Evaluation: T**

Objective:

The objectives of the course are to make the students understand fundamental AI concepts and core AI techniques, explore machine learning and neural networks as key AI components, etc.

Prerequisites: Fundamentals of Propositional Logic, mathematics.

Course Outcomes

On successful completion of the course, the students will be able to:		
SI No	**Course Outcome**	**Blooms Taxonomy Level**
CO 1	**Explain the** fundamental concepts, applications, and ethical implications of AI.	**BT 2**
CO 2	**Apply** uninformed and informed search algorithms to solve AI problems	**BT 3**
CO 3	**Analyze** and implement knowledge representation techniques, including logic-based and probabilistic reasoning.	**BT 4**
CO 4	**Assess** and **design** AI-based solutions using reasoning, decision-making, and planning techniques.	**BT 5 & 6**

Detailed Syllabus:

Module	**Topics**	**Course Content**	**Periods**
I.	**Introduction**	Definition, History, and Evolution of AI, Applications of AI (Healthcare, Finance, Robotics, NLP, etc.), AI vs. Machine Learning vs. Deep Learning, Strong AI vs. Weak AI, AI as Search: Problem Formulation, State-Space Representation, Rational Agents, Types of Agents, Breadth-First Search (BFS), Depth-First Search (DFS), Depth-Limited Search & Iterative Deepening DFS, Uniform Cost Search, Heuristic Function & Admissibility, Greedy Best-First Search, A* Algorithm (Manhattan, Euclidean Heuristics), Hill Climbing & Local Search Algorithms, Definition and Examples (Sudoku, N-Queens), Backtracking Algorithm, Constraint Propagation: Forward Checking, Arc Consistency (AC-3)	**22**
II.	**Knowledge Representation & Reasoning**	Types of Knowledge: Declarative vs. Procedural, Common Sense Knowledge, Knowledge-Based Systems, Propositional Logic: Syntax, Semantics, Logical Connectives, Truth Tables, First-Order Logic (FOL): Predicates, Functions, Quantifiers, Unification & Resolution in FOL, Forward Chaining vs. Backward Chaining, Expert Systems & Case Study: MYCIN (Medical Diagnosis System), Bayesian Networks: Structure, Conditional Probability Tables (CPT), Exact & Approximate Inference in Bayesian Networks, Hidden Markov Models (HMM), Fuzzy Sets, Membership Functions Fuzzy Inference Systems (Mamdani & Sugeno), Defuzzification Techniques	**22**
III.	**Planning in AI**	Definition of Planning in AI, STRIPS Representation and PDDL (Planning Domain Definition Language), State-Space Search in Planning, Forward & Backward Planning, Partial Order Planning (POP), Graph Plan Algorithm, Decision Trees & Utility Theory, Game	**22**

		Theory in AI, Adversarial Search: Minimax Algorithm & Alpha-Beta Pruning, MDP Formulation, Bellman Equations, Policy Evaluation & Policy Iteration, Q-Learning Algorithm	
IV	**AI Applications**	NLP: Text Processing & Tokenization, Named Entity Recognition (NER), Sentiment Analysis Computer Vision: Image Classification & Object Detection, Feature Extraction Techniques Reinforcement Learning: Deep Q-Learning & Neural Networks in RL, Case Study: AI for Self-Driving Cars AI Bias & Fairness, Explainable AI (XAI), AI for Social Good	**22**
TOTAL			**88**

Artificial Intelligence Practice Sessions

Total Practice Hours for the semester = 30 (2 hours per week)

Minimum 10 Laboratory experiments based on the following-

- Implement BFS & DFS in Python
- Solve a pathfinding problem using A* Search
- Constraint satisfaction solver for Sudoku
- Implement logical inference using Propositional Logic.
- Build a Rule-Based Expert System for disease diagnosis.
- Implement a Bayesian Network for predicting weather conditions.
- Develop a Fuzzy Logic Controller for temperature regulation.
- Implement STRIPS-based AI Planning for a block-stacking problem.
- Develop a Tic-Tac-Toe AI using Minimax Algorithm.
- Implement Q-Learning for a simple game (Grid World).
- Sentiment Analysis on Twitter Data using NLP.
- Implement a Handwritten Digit Classifier using OpenCV.
- Train an AI model using Q-Learning for a custom environment.

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4 * 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH	8 * 4 NCH = 32 NCH (Problem Solving, Seminar, Case Study, Discussion, Internship, Projects)

Text Books

1. *Artificial Intelligence: A Modern Approach,* Stuart Russell & Peter Norvig, 4th Edition, 2020, PHI
2. *Artificial Intelligence,* Elaine Rich, Kevin Knight, Shivashankar B. Nair, 3rd Edition, 2017, Tata McGraw Hill

Reference Books:

1. Nils J. Nilsson, *Principles of Artificial Intelligence,* 1993, Morgan Kaufmann Publishers

Subject Name: Data Mining		**Subject Code:CAP054C105**
L-T-P-C – 4-0-0-4	**Credit Units: 04**	**Scheme of Evaluation: T**

Objective:

The objectives of the course are to make the students understand the fundamentals of data mining, explore data mining techniques & algorithms, analyze advanced topics in data mining, and apply data mining for real-world applications.

Prerequisites: Probability & Statistics, Database Management Systems (DBMS)

Course Outcomes

On successful completion of the course, the students will be able to:		
SI No	**Course Outcome**	**Blooms Taxonomy Level**
CO 1	**Explain** the fundamentals of data mining, preprocessing techniques, and data warehousing.	**BT 2**
CO 2	**Apply** classification, clustering, and association rule mining techniques to real-world datasets.	**BT 3**
CO 3	**Analyze** data mining models and **evaluate** their effectiveness using appropriate performance metrics.	**BT 4 & 5**
CO 4	**Develop** and optimize machine learning models for predictive data mining applications.	**BT 6**

Detailed Syllabus:

Module	Topics	Course Content	Periods
I.	**Introduction**	Introduction to Data Mining, Definition and significance of Data Mining, Applications in Business, Healthcare, and Cybersecurity, Data Mining vs. Machine Learning vs. Statistics, Types of Data & Data Exploration, Structured, Semi-structured, and Unstructured Data, Data Quality: Missing Values, Noisy Data, Inconsistencies, Exploratory Data Analysis (EDA), Data Preprocessing & Feature Engineering, Data Cleaning: Handling Missing & Noisy Data, Data Transformation: Normalization, Standardization, Binning, Feature Selection & Feature Extraction, Data Warehousing & OLAP, Data Warehouses vs. Databases, Online Analytical Processing (OLAP) and its operations, ETL (Extract, Transform, Load) Process	**22**
II.	**Classification and Association Rule Mining**	Classification & Prediction, Decision Tree Classifiers (ID3, C4.5, CART), Naïve Bayes Classifier, k-Nearest Neighbors (k-NN), Model Evaluation & Performance Metrics, Confusion Matrix, Precision, Recall, F1-Score, ROC-AUC Curve, Cross-Validation, Association Rule Mining, Market Basket Analysis, Apriori Algorithm, FP-Growth Algorithm, Advanced Classification Techniques. Random Forest Classifier, Support Vector Machines (SVM), Ensemble Learning	**22**
III.	**Clustering and Anomaly Detection**	Clustering Techniques, K-Means Clustering, Hierarchical Clustering (Agglomerative & Divisive), DBSCAN, Cluster Evaluation Methods, Silhouette Score, Davies-Bouldin Index, Outlier Detection & Anomaly Detection, Statistical Outlier Detection (Z-Score, IQR), Isolation Forests, Local Outlier Factor (LOF), Text Mining & Web Mining, Sentiment Analysis, Web Crawling, and Web Data Mining PageRank Algorithm	**22**
IV	**Advanced Topics**	Big Data & Scalable Data Mining, Challenges in Big Data Mining, Hadoop & MapReduce for Data Mining, Apache Spark for Large-Scale Data Processing, Sequential Pattern Mining & Time Series Analysis, Sequence Mining Algorithms, Time Series Forecasting (ARIMA,	**22**

		LSTMs), Privacy & Ethical Considerations in Data Mining, Data Privacy Challenges, GDPR & Data Protection Laws, Fairness & Bias in Data Mining, Real-World Case Studies & Applications, Data Mining in Healthcare (Predictive Analytics), Fraud Detection in Finance, Recommendation Systems	
TOTAL			**88**

Data Mining Lab Syllabus

Total Lab Hours for the semester = 30 (2 hours per week)

Minimum 10 Laboratory experiments based on the following-

- Perform exploratory data analysis (EDA) on a dataset using Pandas and Matplotlib.
- Implement missing data handling techniques and data normalization.
- Used SQL queries for data warehouse operations
- Implement Decision Tree and Naïve Bayes for classification tasks.
- Apply Apriori Algorithm for market basket analysis.
- Evaluate model performance using precision, recall, and ROC curves
- Implement k-Means and Hierarchical Clustering on a real-world dataset.
- Perform anomaly detection using Isolation Forest.
- Conduct sentiment analysis using text mining techniques
- Use Apache Spark for handling big data tasks.
- Perform time series forecasting on stock market data.
- Implement a recommendation system using collaborative filtering.

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4 * 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH	8 * 4 NCH = 32 NCH (Problem Solving, Seminar, Case Study, Discussion, Internship, Projects)

Text Books

1. *Data Mining: Concepts and Techniques,* Jiawei Han, Micheline Kamber, Jian Pei, 3rd Edition, 2011, Morgan Kaufmann
2. *Introduction to Data Mining,* Pang-Ning Tan, Michael Steinbach, Vipin Kumar, 1st Edition, 2016, Pearson

Reference Books:

1. Christopher Bishop, *Pattern Recognition and Machine Learning,* 1st Edition, 2006, Springer
2. Jure Leskovec, Anand Rajaraman, Jeff Ullman, *Mining of Massive Datasets,* 2nd Edition, 2016, Dreamtech Press

Subject Name: Digital Image Processing		**Subject Code: CAP054C106**
L-T-P-C – 4-0-0-4	**Credit Units: 04**	**Scheme of Evaluation: T**

Objective:

The objectives of the course are to make the students understand the fundamentals of digital image processing, learn image enhancement and restoration techniques, analyze image segmentation, feature extraction, and object recognition techniques, and implement advanced techniques in image processing etc.

Prerequisites: Linear Algebra, Probability and Statistics, Signal Processing, Python Programming

Course Outcomes

On successful completion of the course, the students will be able to:		
SI No	**Course Outcome**	**Blooms Taxonomy Level**
CO 1	Understand the fundamental concepts of digital images, image formation, and the visual perception system	**BT 1 & 2**
CO 2	Apply image enhancement techniques in spatial and frequency domains.	**BT 3 & 4**
CO 3	Analyze and implement image restoration, denoising, and color image processing techniques.	**BT 4 & 5**
CO 4	Utilize feature extraction and image compression methods in real-world image analysis applications.	**BT 6**

Detailed Syllabus:

Module	Topics	Course Content	Periods
I.	**Introduction**	Fundamentals of Digital Image Processing, Definition and Applications, Components, Image Representation: Pixels, Resolution, and Bit Depth, Image Perception & Color Models, Human Visual System and Image Perception, Color Spaces: RGB, CMY, HSV, YCbCr, Converting Between Color Models, Image Sampling & Quantization, Sampling and Aliasing, Quantization and Bit-Depth Reduction, Histogram Analysis and Contrast Stretching, Image File Formats & Transformations: BMP, JPEG, PNG, TIFF, Geometric Transformations (Translation, Scaling, Rotation), Affine and Perspective Transformations	**22**
II.	**Image Enhancement and Restoration**	Spatial Domain Processing, Point Processing: Log Transform, Power-Law Transform, Histogram Equalization and Contrast Stretching, Smoothing Filters: Mean, Median, Gaussian, Frequency Domain Processing, Fourier Transform and Frequency Representation of Images, Low-pass and High-pass Filtering, Image Sharpening using Laplacian and Unsharp Masking, Noise Models & Image Restoration, Types of Noise: Gaussian, Salt & Pepper, Speckle Image Denoising Techniques: Spatial and Frequency Domain Filters Wiener Filter and Inverse Filtering, Edge Detection & Morphological Processing, Gradient-Based Edge Detection: Sobel, Prewitt, Canny, Morphological Operations: Dilation, Erosion, Opening, Closing, Skeletonization and Boundary Detection	**22**
III.	**Segmentation, Feature Extraction and**	Thresholding-Based Segmentation, Global vs. Adaptive Thresholding, Otsu’s Method, Watershed Algorithm, Region-Based Segmentation, Region Growing and Region Splitting & Merging, K-Means and Mean-Shift Clustering, Active Contours (Snakes), Feature Extraction Techniques,	**22**

	Object Recognition	Shape Features: Area, Perimeter, Circularity, Texture Features: Gray Level Co-occurrence Matrix (GLCM), Histogram of Oriented Gradients (HOG), Object Recognition & Classification, Template Matching, Feature Matching using SIFT and SURF, Introduction to Convolutional Neural Networks (CNNs) for Image Recognition	
IV	**Image Compression, Wavelets and Advanced Applications**	Image Compression Techniques, Lossless Compression: Huffman Coding, Run-Length Encoding, Lossy Compression: JPEG, MPEG, WebP, Discrete Cosine Transform (DCT) and Quantization, Wavelet Transform & Multiresolution Analysis, Introduction to Wavelets, Discrete Wavelet Transform (DWT), Applications of Wavelets in Image Compression and Denoising, Deep Learning for Image Processing, Introduction to CNNs (LeNet, AlexNet, ResNet), Transfer Learning for Image Classification, Object Detection (YOLO, SSD, Faster R-CNN), Real-Time Image Processing & Applications, Image Processing for Medical Imaging (MRI, X-Ray, CT), Remote Sensing & Satellite Image Processing, Augmented Reality (AR) & Virtual Reality (VR) in Image Processing	**22**
TOTAL			**88**

Digital Image Processing Practice Sessions

Total Practice Hours for the semester = 30 (2 hours per week)

Minimum 10 Laboratory experiments based on the following-

- Read, display, and manipulate images using Python (OpenCV, PIL).
- Convert images between different color models.
- Perform geometric transformations on images.
- Apply histogram equalization and contrast enhancement.
- Implement noise reduction techniques (Mean, Median, Gaussian filtering).
- Perform edge detection using Canny and Sobel operators.
- Implement region-based segmentation using K-means clustering.
- Extract shape and texture features from images.
- Perform feature matching using SIFT and ORB descriptors.
- Implement JPEG compression using DCT.
- Apply wavelet-based denoising techniques.
- Build a simple CNN model for image classification.

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4 * 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH	8 * 4 NCH = 32 NCH (Problem Solving, Seminar, Case Study, Discussion, Internship, Projects)

Text Books

1. *Digital Image Processing,* Rafael C. Gonzalez, Richard E. Woods, 4th Edition, 2018, Pearson
2. *Fundamentals of Digital Image Processing,* Anil K. Jain, 1st Edition, 2015, Pearson

Reference Books:

1. Richard Szeliski, *Computer Vision: Algorithms and Applications*, 11th Edition, 2011, Springer

PSE II/Subject Name: Introduction to Deep Learning	**Subject Code: CAP054C106**
L-T-P-C – 3-0-0-3 **Credit Units: 03**	**Scheme of Evaluation: T**

Objective:

The objectives of the course are to make the students understand the fundamentals of machine learning, apply supervised and unsupervised learning techniques, develop advanced machine learning models, explore deep learning architectures and algorithms, and design and train AI models using modern deep learning techniques.

Prerequisites: Linear Algebra, Probability & Statistics, Python Programming

Course Outcomes

On successful completion of the course, the students will be able to:		
SI No	**Course Outcome**	**Blooms Taxonomy Level**
CO 1	**Understand** the key concepts of Deep Learning and their applications.	**BT 2**
CO 2	**Apply data-driven algorithms such as** regression, classification, and clustering.	**BT 3**
CO 3	**Analyze** and **assess** different neural network architectures and training techniques.	**BT 4 & 5**
CO 4	**Design** and implement deep learning models for real-world applications	**BT 6**

Detailed Syllabus:

Module	Topics	Course Content	Periods
I.	**ML Fundamentals**	Definition and Types of ML: Supervised, Unsupervised, Reinforcement Learning, Applications of ML in Healthcare, Finance, NLP, and Computer Vision, Overview of ML Pipelines, Linear Algebra: Vectors, Matrices, Eigenvalues, and Eigenvectors, Probability Theory: Bayes' Theorem, Conditional Probability, Optimization: Gradient Descent, Stochastic Gradient Descent (SGD), Linear Regression: Least Squares Method, Gradient Descent, Polynomial Regression, Ridge & Lasso Regression, Evaluation Metrics: MSE, RMSE, R^2 Score, Logistic Regression, k-Nearest Neighbors (k-NN), Decision Trees & Random Forest, Evaluation Metrics: Confusion Matrix, Precision, Recall, F1-Score	**22**
II.	**Advanced AL Techniques**	Support Vector Machines (SVM): Hard Margin & Soft Margin SVM Kernel Trick: RBF, Polynomial Kernels, Unsupervised Learning, Clustering: k-Means, Hierarchical Clustering, DBSCAN, Dimensionality Reduction: Principal Component Analysis (PCA), t-SNE, Ensemble Learning & Boosting Techniques, Bagging & Random Forest Boosting: AdaBoost, Gradient Boosting, XGBoost Neural Networks Basics, Perceptron & Multi-Layer Perceptron (MLP), Activation Functions: Sigmoid, ReLU, Tanh, Backpropagation Algorithm	**22**

III.	**Deep Learning Fundamentals**	Introduction to Deep Learning, Difference Between ML and DL, Applications of Deep Learning (NLP, Image Recognition, Generative Models), Neural Networks & Optimization, Deep Neural Networks (DNN): Weight Initialization, Vanishing & Exploding Gradient Problems, Optimizers: SGD, Adam, RMSprop, Convolutional Neural Networks (CNNs), Convolution & Pooling Layers: Popular CNN Architectures: LeNet, AlexNet, VGG, ResNet, Recurrent Neural Networks (RNNs) & Sequence Models, RNNs & Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Applications in NLP & Time-Series Forecasting	**22**
IV	**Advanced DL Concepts**	Generative Models: Autoencoders & Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), Transformers & Attention Mechanisms, Self-Attention and Multi-Head Attention, Transformer Architecture (BERT, GPT, T5), Reinforcement Learning Basics, Markov Decision Process (MDP), Q-Learning & Deep Q Networks (DQN), Ethics & Deployment of AI Models. Bias in AI Models, Fairness & Explainability, Model Deployment: Flask, FastAPI, TensorFlow Serving	**22**
TOTAL			**88**

Introduction to Deep Learning Practice sessions

Total Practice Hours for the semester = 30 (2 hours per week)

Minimum 10 Laboratory experiments based on the following-

- Implement Linear and Polynomial Regression on a dataset.
- Implement Logistic Regression for a classification task.
- Apply k-NN and Decision Trees for classification and compare their performance.
- Implement SVM with different kernels.
- Perform k-Means clustering and PCA on real-world datasets.
- Apply Random Forest and boosting techniques for a classification problem.
- Implement a simple Deep Neural Network using TensorFlow/PyTorch.
- Train a CNN for image classification (MNIST/CIFAR-10).
- Build an RNN/LSTM model for sentiment analysis or stock price prediction.
- Implement a GAN for image generation.
- Fine-tune a pre-trained Transformer model for text classification.
- Deploy a deep learning model as an API using Flask or FastAPI.

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4 * 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH	8 * 4 NCH = 32 NCH (Problem Solving, Seminar, Case Study, Discussion, Internship, Projects)

Text Books

1. *Deep Learning,* Ian Goodfellow, Yoshua Bengio, Aaron Courville, 2016, MIT Press
2. *Pattern Recognition and Machine Learning,* Christopher M. Bishop, 1st Edition, 2006, Springer

Reference Books:

1. Kevin P. Murphy, *Machine Learning: A Probabilistic Perspective*, 2012, MIT Press
2. ***N**eural Networks and Deep Learning, Michael Nielsen, 2015, Determination Press*
3. Richard S. Sutton, Andrew G. Barto, *Reinforcement Learning: An Introduction,* 2nd Edition, 1998, Bradford Books
4. Michael Nielsen, *Neural Networks and Deep Learning*, 2010

Subject Name: Statistical Computing		**Subject Code: CAP054D108**
L-T-P-C – 4-0-0-4	**Credit Units: 04**	**Scheme of Evaluation: T**

Objective:

The objectives of the course are to make the students understand the fundamentals of statistical computing, implement statistical methods computationally, analyze real-world datasets using statistical computing techniques, and develop computational tools for data-driven decision making

Prerequisites: Probability and Statistics, Linear Algebra

Course Outcomes

On successful completion of the course, the students will be able to:		
SI No	**Course Outcome**	**Blooms Taxonomy Level**
CO 1	**Understand the** fundamental concepts of statistical computing and probability distributions.	**BT 2**
CO 2	**Apply** statistical inference, hypothesis testing, and regression techniques.	**BT 3**
CO 3	**Analyze** and **assess** multivariate data and use Bayesian inference methods	**BT 4 & 5**
CO 4	**Develop** statistical models using high-performance computing techniques.	**BT 6**

Detailed Syllabus:

Module	**Topics**	**Course Content**	**Periods**
I.	**Fundamental Concepts**	Introduction to Statistical Computing, Importance of statistical computing in data analysis, Statistical computing vs. theoretical statistics, Role of Python/R in statistical computing, Random Variables & Probability Distributions, Discrete & Continuous Probability Distributions, Bernoulli, Binomial, Poisson, Normal, Exponential Distributions, Probability Density Function (PDF) and Cumulative Distribution Function (CDF), Statistical Sampling & Simulation, Random Sampling Techniques, Law of Large Numbers & Central Limit Theorem, Monte Carlo Simulation, Statistical Data Analysis & Visualization, Exploratory Data Analysis (EDA), Data Visualization with Matplotlib, Seaborn (Python) / ggplot2 (R), Histogram, Boxplot, KDE plots	**22**

II.	**Statistical Inference and Regression Analysis**	Estimation & Hypothesis Testing, Maximum Likelihood Estimation (MLE), Confidence Intervals, Parametric vs. Non-Parametric Hypothesis Testing, Resampling Techniques, Bootstrap Method, Jackknife Estimation, Permutation Testing, Regression Analysis, Simple & Multiple Linear Regression, Assumptions of Regression Models, Generalized Linear Models (GLMs), Non-Linear & Robust Regression, Polynomial Regression, Ridge & Lasso Regression, Robust Regression Techniques	**22**
III.	**Multivariate Analysis and Bayesian Computing**	Multivariate Statistical Methods, Principal Component Analysis (PCA), Factor Analysis, Canonical Correlation Analysis, Bayesian Statistics, Bayesian Inference Basics, Conjugate Priors, Bayesian Regression, Markov Chain Monte Carlo (MCMC) Methods, Metropolis-Hastings Algorithm, Gibbs Sampling, Bayesian Networks, Time Series Analysis & Forecasting, Autoregressive (AR) and Moving Average (MA) Models, ARIMA and SARIMA Models, Hidden Markov Models (HMM)	**22**
IV	**High-Performance Statistical Computing and Applications**	Numerical Optimization in Statistics, Gradient Descent & Stochastic Gradient Descent (SGD), Newton-Raphson Method, Convex Optimization in Statistical Models, Parallel Computing & Big Data Statistics, Introduction to Parallel Computing in R (foreach, parallel), Distributed Computing with Apache Spark for Statistical Computing, Cloud-Based Statistical Computing (Google Cloud, AWS), Statistical Learning & Machine Learning Integration, Overview of Supervised & Unsupervised Learning, Statistical Foundations of Machine Learning, Ensemble Methods: Bagging, Boosting, Random Forest, Case Studies & Real-World Applications, Statistical Computing in Finance, Bioinformatics & Healthcare Statistics, Econometrics & Social Science Applications	**22**
TOTAL			**88**

Statistical Computing Practice Session

Total Practice Hours for the semester = 30 (2 hours per week)

Minimum 10 Laboratory experiments based on the following-

- Implement probability distributions and visualize them.
- Perform random sampling and compare theoretical vs. empirical distributions.
- Implement Monte Carlo simulations for probability estimation.
- Implement hypothesis testing using real-world datasets.
- Perform Bootstrap and Jackknife estimation in R/Python.
- Develop a regression model and validate assumptions
- Perform PCA for dimensionality reduction.
- Implement Bayesian inference using PyMC3/Stan.
- Apply ARIMA models for time series forecasting.
- Implement optimization algorithms for statistical models.
- Used Apache Spark for large-scale statistical analysis.
- Perform statistical computing on a cloud platform.

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**

4 * 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH	8 * 4 NCH = 32 NCH (Problem Solving, Seminar, Case Study, Discussion, Internship, Projects)

Text Books

1. *The Elements of Statistical Learning,* Trevor Hastie, Robert Tibshirani, Jerome Friedman, 2nd Edition, 2009, Springer
2. *Statistical Computing with R,* Maria L. Rizzo, 2nd Edition, 2019, Chapman and Hall
3. *Bayesian Data Analysis,* Andrew Gelman, John B. Carlin, 3rd Edition, 2019, Chapman and Hall

Reference Books:

1. Gareth James, Daniela Witten, *Introduction to Statistical Learning with Applications in R,* 7th Edition, 2017, Springer

Subject Name: Machine Learning Applications		**Subject Code: CAP054C109**
L-T-P-C – 4-0-0-4	**Credit Units: 04**	**Scheme of Evaluation: T**

Objective:

The objectives of the course are to make the students understand the fundamentals of machine learning, apply supervised and unsupervised learning techniques, develop advanced machine learning models, explore deep learning architectures and algorithms and design and train ai models using modern deep learning techniques

Prerequisites: Linear Algebra, Probability & Statistics, Python Programming

Course Outcomes

On successful completion of the course the students will be able to:		
SI No	**Course Outcome**	**Blooms Taxonomy Level**
CO 1	**Understand** the key concepts of machine learning and their applications.	**BT 2**
CO 2	**Apply** machine learning (ML) algorithms such as regression, classification, and clustering.	**BT 3**
CO 3	**Analyze** and **assess** different neural network architectures and training techniques for image	**BT 4 & 5**
CO 4	**Design** and implement deep learning models for real-world applications	**BT 6**

Detailed Syllabus:

Module	**Topics**	**Course Content**	**Periods**

I.	**ML Fundamentals**	Definition and Types of ML: Supervised, Unsupervised, Reinforcement Learning, Applications of ML in Healthcare, Finance, NLP, and Computer Vision, Overview of ML Pipelines, Linear Algebra: Vectors, Matrices, Eigenvalues, and Eigenvectors, Probability Theory: Bayes' Theorem, Conditional Probability, Optimization: Gradient Descent, Stochastic Gradient Descent (SGD), Linear Regression: Least Squares Method, Gradient Descent, Polynomial Regression, Ridge & Lasso Regression, Evaluation Metrics: MSE, RMSE, R^2 Score, Logistic Regression, k-Nearest Neighbors (k-NN), Decision Trees & Random Forest, Evaluation Metrics: Confusion Matrix, Precision, Recall, F1-Score	**22**
II.	**Advanced AL Techniques**	Support Vector Machines (SVM): Hard Margin & Soft Margin SVM Kernel Trick: RBF, Polynomial Kernels, Unsupervised Learning, Clustering: k-Means, Hierarchical Clustering, DBSCAN, Dimensionality Reduction: Principal Component Analysis (PCA), t-SNE, Ensemble Learning & Boosting Techniques, Bagging & Random Forest Boosting: AdaBoost, Gradient Boosting, XGBoost Neural Networks Basics, Perceptron & Multi-Layer Perceptron (MLP), Activation Functions: Sigmoid, ReLU, Tanh, Backpropagation Algorithm	**22**
III.	**Deep Learning Fundamentals**	Introduction to Deep Learning, Difference Between ML and DL, Applications of Deep Learning (NLP, Image Recognition, Generative Models), Neural Networks & Optimization, Deep Neural Networks (DNN): Weight Initialization, Vanishing & Exploding Gradient Problems, Optimizers: SGD, Adam, RMSprop, Convolutional Neural Networks (CNNs), Convolution & Pooling Layers: Popular CNN Architectures: LeNet, AlexNet, VGG, ResNet, Recurrent Neural Networks (RNNs) & Sequence Models, RNNs & Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Applications in NLP & Time-Series Forecasting	**22**
IV	**Advanced DL Concepts**	Generative Models: Autoencoders & Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), Transformers & Attention Mechanisms, Self-Attention and Multi-Head Attention, Transformer Architecture (BERT, GPT, T5), Reinforcement Learning Basics, Markov Decision Process (MDP), Q-Learning & Deep Q Networks (DQN), Ethics & Deployment of AI Models. Bias in AI Models, Fairness & Explainability, Model Deployment: Flask, FastAPI, TensorFlow Serving	**22**
TOTAL			**88**

Machine Learning Practice Sessions

Total Practice Hours for the semester = 30 (2 hours per week)

Minimum 10 Laboratory experiments based on the following-

- Implement Linear and Polynomial Regression on a dataset.

- Implement Logistic Regression for a classification task.
- Apply k-NN and Decision Trees for classification and compare their performance.
- Implement SVM with different kernels.
- Perform k-Means clustering and PCA on real-world datasets.
- Apply Random Forest and Boosting techniques for a classification problem.
- Implement a simple Deep Neural Network using TensorFlow/PyTorch.
- Train a CNN for image classification (MNIST/CIFAR-10).
- Build an RNN/LSTM model for sentiment analysis or stock price prediction.
- Implement a GAN for image generation.
- Fine-tune a pre-trained Transformer model for text classification.
- Deploy a deep learning model as an API using Flask or FastAPI.

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4 * 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH	8 * 4 NCH = 32 NCH (Problem Solving, Seminar, Case Study, Discussion, Internship, Projects)

Textbooks

1. *Pattern Recognition and Machine Learning,* Christopher M. Bishop, 1st Edition, 2006, Springer
2. *Deep Learning,* Ian Goodfellow, Yoshua Bengio, Aaron Courville, 2016, MIT Press

Reference Books:

1. Kevin P. Murphy, *Machine Learning: A Probabilistic Perspective*, 2012, MIT Press
2. Richard S. Sutton, Andrew G. Barto, *Reinforcement Learning: An Introduction,* 2nd Edition, 1998, Bradford Books
3. Michael Nielsen, *Neural Networks and Deep Learning*, 2010

6.3 Detailed Syllabus of 2nd Semester

Subject Name: Web Technology		**Subject Code: CAP054C201**
L-T-P-C – 3-0-0-3	**Credit Units: 03**	**Scheme of Evaluation: T**

Objective:

To provide a comprehensive understanding of the modern web ecosystem, including client- and server-side technologies. Students will learn to design responsive web interfaces, develop backend logic, integrate APIs, and deploy secure full-stack applications. The course also introduces modern development practices like CI/CD and cloud-based deployment.

Prerequisites:

- Basic programming knowledge (preferably Python or JavaScript)
- Familiarity with HTML and CSS is recommended
- Basic understanding of how the internet and web browsers work

Course Outcomes:

SI No	Course Outcome	Bloom's Taxonomy Level
CO 1	Understand core web technologies and develop interactive web pages	**BT 1 & 2**
CO 2	Implement web development using modern front-end and back-end frameworks	**BT 3**
CO 3	Develop dynamic web applications using RESTful APIs and cloud integration	**BT 4**
CO 4	Analyze and apply secure web development practices	**BT 5**
CO 5	Design full-stack web applications following industry standards	**BT 6**

Detailed Syllabus:

Modules	Topics	Course content	Periods
I	**Introduction to Internet and Web Page Design**	Introduction to Web Technologies: Evolution of the Web, Web 2.0 & Web 3.0, Web Standards (W3C, ECMA). HTML5 & CSS3: Semantic elements, Forms, Flexbox, Grid Layout, Media Queries, Animations, Transitions. JavaScript & ES6+: DOM Manipulation, Async/Await, Fetch API, Event Handling, JSON, Promises & Callbacks. Front-End Frameworks: React.js (Components, Props, State Management, Hooks), Tailwind CSS, Bootstrap	**22**
II	**Web Browsers, Markup Language Basics, and XML**	Web Browsers: functions and working principle of web browsers; plug-ins & helper applications; conceptual architecture of some typical web browsers. Markup language basics: Standard Generalized Markup Language (SGML)- structures, elements, Content models, DTD, attributes, entities. Extensible Markup Language (XML): Markup Languages: HTML5, XML, JSON, SVG. Data Handling in XML & JSON: XML Schema, XSLT, JSON Schema, Fetching & Parsing APIs. API Development: RESTful API vs GraphQL, API Authentication & Rate Limiting	**22**

III	**Web Server Side**	Web Servers: Architecture of Web Servers, Apache, Nginx, Node.js Express, Serverless Web Apps. Back-End Development: Node.js, Express.js, Flask/Django, Database Integration (MongoDB, MySQL, Firebase, PostgreSQL). Authentication & Authorization: JWT, OAuth2, Role-Based Access Control (RBAC). Cloud Integration: AWS Lambda, Firebase Functions, Cloud Storage Solutions (S3, Google Cloud Storage), Deployment Strategies (Docker, Kubernetes).	**22**
IV	**Advanced Web Technologies and Web Security**	Modern Web Technologies: Progressive Web Apps (PWAs), WebSockets, WebRTC, Headless CMS (Contentful, Strapi). Cloud & DevOps: Serverless Computing, Infrastructure as Code (Terraform, CloudFormation), CI/CD Pipelines (Jenkins, GitHub Actions). Web Security: HTTPS, CSP, CORS, OWASP Top 10 Vulnerabilities, Secure Coding Best Practices, API Security (Rate Limiting, Token-Based Authentication), SQL Injection & Cross-Site Scripting (XSS) Prevention	**22**
		Total	**88**

Subject Name: Web Technology **Subject Code: CAP054C211**
L-T-P-C – 0-0-2-1 **Credit Units: 01** **Scheme of Evaluation: P**

Total Lab Hours for the semester = 30 (2 hours per week)

Minimum 15 Laboratory experiments based on the following-

Experiment No.	Title	Objective
1	Responsive Web Design with HTML5 & CSS3	Develop a responsive website using CSS Flexbox & Grid
2	JavaScript Event Handling & DOM Manipulation	Implement dynamic UI updates using JavaScript
3	RESTful API Development	Build a REST API with Node.js, Express & MongoDB
4	Frontend Development with React.js	Create a Single Page Application (SPA) using React
5	User Authentication with JWT & OAuth2	Implement secure login & authentication in a web app
6	Cloud Deployment of Web Applications	Deploy a web app on AWS Lambda or Firebase
7	API Security with Rate Limiting	Secure APIs using JWT, OAuth2 & API Gateways
8	WebSockets & Real-Time Communication	Develop a real-time chat application with WebSockets
9	Implementing CI/CD Pipelines	Automate web deployment with GitHub Actions & Docker
10	Progressive Web Application (PWA)	Build a PWA with offline support using Service Workers
11	Secure Web Development	Prevent SQL Injection, XSS, and CSRF attacks in web apps
12	Headless CMS Integration	Connect React/Next.js with a headless CMS (Concertful/Strapi)
13	Web Performance Optimization	Analyze website performance using Lighthouse & Chrome DevTools
14	Containerized Web Applications	Deploy a full-stack application using Docker & Kubernetes

15	WebAssembly (WASM) Integration	Run high-performance code using WebAssembly with JavaScript

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
2 * 22 NCH = 44 NCH	2 * 15 NCH = 30 NCH	8 * 2 NCH = 16 NCH (Problem Solving, Seminar, Case Study, Discussion, Internship, Projects)

Textbook:

1. *Web Development and Design Foundations with HTML5*, Terry Felke-Morris, 9th Edition, 2020, Pearson.

Reference Books:

1. *Full Stack Web Development with React and Node.js*, David Griffiths, 1st Edition, 2021, O'Reilly Media.
2. *Eloquent JavaScript: A Modern Introduction to Programming*, Marijn Haverbeke, 3rd Edition, 2018, No Starch Press.
3. *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics*, Jennifer Robbins, 5th Edition, 2018, O'Reilly Media.

Subject Name: System Administration		**Subject Code:CAP054C202**
L-T-P-C – 3-0-0-3	**Credit Units: 03**	**Scheme of Evaluation: T**

Objective:

The course aims to provide students with a deep understanding of operating system fundamentals and essential system administration tasks. It covers operating system concepts such as process management, memory management, and file systems while also introducing hands-on system administration, including user management, software installation, disk operations, and basic network configuration**.**

Prerequisites: Basics of traditional operating systems (Processes, Threads, Memory, I/O), Basic knowledge of computer architecture & networking

Course Outcomes:

On successful completion of the course, the students will be able to:		
SI No	**Course Outcome**	**Blooms Taxonomy Level**
CO 1	Understand modern operating system architectures and design principles.	**BT 1 & 2**
CO 2	Analyze CPU scheduling, memory management, and concurrency mechanisms.	**BT 3**
CO 3	Perform common system administration tasks in Linux-based environments	**BT 4**

CO 4	Configure and troubleshooting user management, file permissions, and system services	**BT 5**

Detailed Syllabus:

Modules	**Topics**	**Course content**	**Periods**
I	**Basics of Operating Systems**	Introduction to OS, types and functions. Process states and scheduling. Threads, concurrency, synchronization using semaphores and monitors. Deadlocks: detection, prevention, and avoidance	**22**
II	**Memory, File, and Device Management**	Memory management techniques: paging, segmentation, virtual memory. File systems: directory structures, file allocation methods, access control. I/O systems and device drivers. Disk scheduling and RAID concepts.	**22**
III	**Introduction to System Administration**	Role of a system administrator. Boot process and system initialization. Package management (apt, yum). User account creation, groups, permissions, sudo. File and process monitoring tools (top, ps, kill). Backup strategies and cron jobs.	**22**
IV	**Networking and Shell Scripting**	Network configuration and tools (ifconfig, netstat, ping, traceroute). Remote access (SSH). System logging, firewall configuration (ufw, iptables). Introduction to shell scripting: variables, conditionals, loops, functions, automation scripts.	**22**
Total			**88**

Subject Name: System Administration	**Lab**	**Subject Code:CAP054C212**
L-T-P-C – 0-0-2-1	**Credit Units: 01**	**Scheme of Evaluation: P**

Total Lab Hours for the semester = 30 (2 hours per week)

Laboratory experiments based on the following-

1. Demonstrate process creation and inter-process communication using fork and pipes
2. Simulate CPU scheduling algorithms (FCFS, SJF, Round Robin)
3. Implement memory management simulation using paging/segmentation
4. Create users and manage groups, passwords, and access permissions
5. Install and remove software packages via command line
6. Monitor system activity using ps, top, vmstat, and netstat
7. Configure cron jobs and automate backups

8. Set up and secure SSH for remote login
9. Write shell scripts for file handling and system automation
10. Configure basic firewall rules and network settings

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
2 * 22 NCH = 44 NCH	2 * 15 NCH = 30 NCH	8 * 2 NCH = 16 NCH (Problem Solving, Seminar, Case Study, Discussion, Internship, Projects)

Textbook:

1. *System Programming and Operating Systems*, Dhamdhere D. M., 2nd Edition, 2014, Tata McGraw-Hill
2. Operating System Concepts, Silberschatz, Galvin & Gagne, 10th Edition

Reference Books: Linux Kernel Development, Robert Love

1. *System Software: An Introduction to Systems Programming*, Leland L. Beck, 3rd Edition, 1997, Pearson Education.
2. Cloud Computing Principles, Rajkumar Buyya
3. The Art of Computer Systems Performance Analysis, Raj Jain

Subject Name: Internet Protocols and Network Design		**Subject Code:CAP054C203**
L-T-P-C – 3-0-0-3	**Credit Units: 03**	**Scheme of Evaluation: T**

Objective:

To provide a comprehensive understanding of network design principles and protocols used in the Internet. The course covers layered architectures, addressing schemes, routing protocols, IP design, subnetting, switching techniques, and security considerations to equip students with practical skills for designing and implementing scalable, efficient, and secure networks.

Prerequisites:

- Basic knowledge of computer networks
- Familiarity with TCP/IP model and networking hardware

SI No	Course Outcome	Bloom's Taxonomy Level
CO 1	Understand Internet architecture and layered protocol models	BT 1 & 2

CO 2	Design and implement IPv4/IPv6 addressing and subnetting schemes	BT 3 & 4	**Course Outcomes:**
CO 3	Analyze and configure routing protocols for network design	BT 4 & 5	
CO 4	Apply switching, NAT, and VPN concepts to network design	BT 5	

Detailed Syllabus:

Module	Topics	Course Content	Periods
I	**Introduction to Internet Protocols**	Overview of network architecture and Internet standards. OSI and TCP/IP models. IP addressing: IPv4, IPv6, private/public addresses, address resolution (ARP, RARP), ICMP. Subnetting and supernetting.	22
II	**Routing and Switching**	Routing concepts, static and dynamic routing, distance vector and link state protocols (RIP, OSPF, BGP). Switching: LAN switching, VLANs, inter-VLAN routing, spanning tree protocol (STP), link aggregation.	22
III	**Network Design Principles**	Hierarchical network design, addressing plans, redundancy and failover, NAT, DHCP, DNS, wireless networks, enterprise topologies, ISP-level networking, remote access (VPN, tunneling).	22
IV	**Security and Performance**	Network security fundamentals: firewalls, IDS/IPS, packet filtering. Secure protocols (HTTPS, SSH, IPSec). Monitoring and troubleshooting: SNMP, NetFlow, Wireshark. Performance optimization and QoS.	22

Subject Name: Internet Protocols and Network Design Lab		**Subject Code:CAP054C213**
L-T-P-C – 0-0-2-1	**Credit Units: 01**	**Scheme of Evaluation: P**

Internet Protocol and Network Design Lab (4 Hours/Week)

1. Design and implement IP addressing schemes for a given scenario

2. Configure static routing between multiple routers using packet tracer

3. Implement dynamic routing using RIP and OSPF

4. Setup and configure VLANs and inter-VLAN routing

5. Configure a basic firewall using access control lists (ACLs)

6. Implement DHCP and DNS in a simulated environment

7. Design and simulate a hierarchical enterprise network topology

8. Analyze packet transmission using Wireshark

9. Implement NAT and port forwarding

10. Configure secure remote access using SSH

11. Setup and test a site-to-site VPN tunnel

12. Use SNMP and NetFlow for network monitoring

13. Simulate IPv6-based network design

14. Configure wireless network access and MAC filtering

15. Analyze routing tables and troubleshoot network loops

16. Create redundancy using HSRP or VRRP

17. Optimize network performance using QoS policies

18. Secure a router with passwords and encryption

19. Monitor network using syslog and SNMP traps

20. Mini project: Design and simulate a secure, scalable network for a multi-branch organization

Subject Name: Natural Language Processing		**Subject Code: CAP054C204**
L-T-P-C – 4-0-0-4	**Credit Units: 04**	**Scheme of Evaluation: T**

Objective:

The objectives of the course are to make the students understand the application of AI in the field of Natural Language Processing, learn the fundamentals of NLP, and design NLP-based applications.

Prerequisites: Probability & Statistics, Linear Algebra, Machine Learning, Python

Course Outcomes

On successful completion of the course, the students will be able to:		
SI No	**Course Outcome**	**Blooms Taxonomy Level**
CO 1	**Understand** fundamental NLP concepts, text processing techniques, and linguistic properties.	**BT 2**
CO 2	**Apply** traditional ML algorithms for text classification, sentiment analysis, and topic modeling.	**BT 3**
CO 3	**Analyze** and **assess** deep learning models for NLP tasks, including transformers and attention mechanisms.	**BT 4 & 5**
CO 4	**Design** and implement NLP applications such as chatbots, summarization, and text generation.	**BT 6**

Detailed Syllabus:

Module	Topics	Course Content	Periods
I.	**Introduction**	Overview of NLP: Definition and importance of NLP, Applications: Chatbots, Machine Translation, Sentiment Analysis, Speech Recognition, Challenges in NLP: Ambiguity, Data Sparsity, Context Understanding, Text Processing & Linguistic Basics, Text Normalization: Tokenization, Stemming, Lemmatization, Stopword Removal and Part-of-Speech (POS) Tagging, Named Entity Recognition (NER), Regular Expressions & Text Representation, Regex for text preprocessing, Bag-of-Words (BoW), TF-IDF, Word Frequency Analysis, Word Embeddings & Semantic Representation, Word2Vec: Skip-gram & CBOW models, GloVe (Global Vectors for Word Representation), FastText	**22**
II.	**Classical NLP Techniques and Language Modelling**	N-gram Language Models: Unigram, Bigram, Trigram Models, Probability Estimation: Smoothing Techniques (Laplace, Kneser-Ney), Perplexity and Evaluation of Language Models, Text Classification & Sentiment Analysis, Naïve Bayes Classifier for Text Classification, Logistic Regression & SVM for NLP Tasks, Sentiment Analysis Using ML Techniques, Topic Modeling & Information Retrieval, Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA), TF-IDF for Document Retrieval, Machine Translation & Sequence Labeling, Statistical Machine Translation (SMT), Hidden Markov Models (HMM) for POS Tagging, Conditional Random Fields (CRF) for Sequence Labeling	**22**
III.	**Deep Learning for NLP**	Neural Networks for NLP: Basics of Neural Networks for NLP, Word Embeddings with Neural Networks (Word2Vec, GloVe), Feedforward and Recurrent Neural Networks (RNNs), Sequence Models & Attention Mechanism, Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) & Gated Recurrent Unit (GRU), Attention Mechanism & Self-Attention, Transformers & Pretrained Language Models, Transformer Architecture (Vaswani et al.), BERT (Bidirectional Encoder Representations from Transformers), GPT (Generative Pretrained Transformer), T5, XLNet, Text Generation & Summarization, Seq2Seq Models for Text Generation, Abstractive & Extractive Text Summarization, Fine-Tuning Transformers for Summarization	**22**
IV	**Advanced NLP Applications**	Conversational AI & Chatbots: Rule-Based Chatbots vs. AI-Based Chatbots, Intent Recognition and Response Generation, DialogFlow, Rasa, GPT-based Chatbots, Speech Processing & Text-to-Speech (TTS). Speech Recognition Models (CMU Sphinx, DeepSpeech, Whisper), Text-to-Speech Synthesis (Tacotron, WaveNet), Bias & Ethics in NLP, Challenges of Bias in NLP Models, Fairness in NLP & Model Interpretability, Ethical Considerations in AI-Powered Language Models, NLP Model Deployment,	**22**

		Deploying NLP models using Flask/FastAPI, Optimizing NLP Models for Production, Cloud-based NLP Services (AWS, Google AI, Hugging Face API)	
TOTAL			**88**

Natural Language Processing Practice Sessions

Total Practice Hours for the semester = 30 (2 hours per week)

Minimum 10 Laboratory experiments based on the following-

- Implement tokenization, stemming, and lemmatization using NLTK/spaCy.
- Perform POS tagging and Named Entity Recognition (NER).
- Build word embeddings using Word2Vec and visualize embeddings.
- Train an N-gram model and evaluate it using perplexity.
- Implement Naïve Bayes and SVM for sentiment analysis.
- Perform topic modeling using LDA on a real-world dataset.

- Implement RNN, LSTM, and GRU models for text generation.
- Fine-tune BERT for text classification.
- Train a Seq2Seq model for machine translation.
- Build and deploy a chatbot using Rasa or OpenAI GPT API.
- Train a speech-to-text model using DeepSpeech.
- Deploy an NLP model as an API using Flask.

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4 * 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH	8 * 4 NCH = 32 NCH (Problem Solving, Seminar, Case Study, Discussion, Internship, Projects)

Text Books

1. *Speech and Language Processing,* Daniel Jurafsky & James H. Martin, 2nd Edition, 2008, Pearson
2. *Natural Language Processing with Python,* Steven Bird, Ewan Klein, Edward Loper, 1st Edition, 2009, O'Reily

Reference Books:

1. Nitin Indurkhya & Fred J. Damerau, *Handbook of Natural Language Processing*, 2nd Edition, 2010, Taylor & Francis

Subject Name: Big Data Analytics		**Subject Code: CAP054C205**
L-T-P-C – 4-0-0-4	**Credit Units: 04**	**Scheme of Evaluation: T**

Objective:

The objectives of the course are to enable students to understand the fundamentals of big data and its challenges, learn big data processing techniques and tools, apply machine learning techniques to big data, and develop big data solutions for real-world applications.

Prerequisites: Probability & Statistics, Database Management Systems (DBMS), Python/Java Programming, Basic Data Structures and Algorithms

Course Outcomes

On successful completion of the course, the students will be able to:		
SI No	**Course Outcome**	**Blooms Taxonomy Level**
CO 1	Understand the characteristics of big data and the challenges associated with its storage and processing.	**BT 2**
CO 2	Apply Hadoop architecture, HDFS, and MapReduce programming for distributed data processing.	**BT 3**
CO 3	Apply Hadoop architecture, HDFS, and MapReduce programming for distributed data processing.	**BT 4 & 5**
CO 4	Design and optimize NoSQL database systems (e.g., MongoDB, Cassandra) for big data applications.	**BT 6**

Detailed Syllabus:

Module	Topics	Course Content	Periods
I.	**Introduction to Big Data and Storage Systems**	Introduction to Big Data: Definition and Characteristics (3Vs: Volume, Velocity, Variety), Challenges in Big Data Analytics, Applications in Healthcare, Finance, and IoT, Big Data Storage & Management, Traditional Databases vs. Big Data Storage, NoSQL Databases (MongoDB, Cassandra, HBase), Distributed File Systems (HDFS, Amazon S3, Google Bigtable), Data Acquisition & Preprocessing, Data Ingestion: Batch vs. Stream Processing, Data Cleaning and Transformation, Schema Design for Big Data, Introduction to Distributed Computing, Basics of Parallel and Distributed Processing, CAP Theorem and BASE Properties, Google's Big Data Technologies: Bigtable, MapReduce, Spanner	**22**
II.	**Hadoop & Spark**	Hadoop Ecosystem, Hadoop Architecture and Components (HDFS, YARN, MapReduce), Hadoop Cluster Setup, Hadoop vs. Spark, MapReduce Programming Model, Understanding the MapReduce Workflow, Writing MapReduce Programs (Java/Python), Combiner and Partitioner in MapReduce, Apache Spark & Resilient Distributed Datasets (RDDs), Spark Core Concepts and Architecture Transformations and Actions in RDDs, Spark DataFrames and Datasets, Advanced Spark Concepts, Spark SQL and DataFrames, Spark MLib for Machine Learning, Performance Tuning in Spark	**22**
III.	**Machine Learning & Streaming Analytics**	Machine Learning with Big Data, Challenges of Machine Learning on Big Data, Scalable ML Algorithms (Decision Trees, Clustering, Regression), Apache Spark MLlib, Big Data Streaming Analytics, Introduction to Stream Processing, Apache Kafka, and Apache Flink Real-time Data Processing with Spark Streaming, Graph Processing with Big Data, Introduction to Graph Analytics, Apache Giraph and GraphX in Spark, PageRank Algorithm, Text & Social Media Analytics, Sentiment Analysis on Large-scale Text Data, Natural Language Processing (NLP) using Spark, Twitter and Social Media Data Analysis	**22**
IV	**Cloud-Based Big Data Analytics**	Big Data on Cloud Platforms, Google Cloud BigQuery, AWS Big Data Services (Redshift, EMR), Microsoft Azure Data Lake, Big Data Security & Privacy, Data Governance & Compliance (GDPR, CCPA), Secure Data Storage & Access Control, Ethical Considerations in Big Data Analytics, Big Data Use Cases & Applications, Fraud Detection in Banking & Finance, Healthcare Analytics for Disease Prediction Smart Cities and IoT Data Analysis, Future Trends in Big Data Analytics, AI and Big Data Integration, Quantum Computing for Big Data, Edge Computing and IoT Analytics	**22**
TOTAL			**88**

Big Data Analytics Practice Sessions

Total Practice Hours for the semester = 30 (2 hours per week)

Minimum 10 Laboratory experiments based on the following-

- Setup and configure Hadoop Distributed File System (HDFS).
- Perform CRUD operations on NoSQL databases (MongoDB, Cassandra).
- Implement batch and stream data ingestion techniques.
- Write a MapReduce program for word count and log processing.
- Implement data transformations using Spark RDDs and DataFrames.
- Perform SQL operations on Spark DataFrames
- Implement a recommendation system using Spark MLlib.
- Process real-time streaming data using Apache Kafka.
- Perform sentiment analysis on Twitter data.
- Deploy and analyze Big Data workloads on AWS/Azure.
- Perform fraud detection using Big Data techniques.
- Build a predictive model for healthcare analytics.

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4 * 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH	8 * 4 NCH = 32 NCH (Problem Solving, Seminar, Case Study, Discussion, Internship, Projects)

Textbooks

1. *Hadoop: The Definitive Guide,* Tom White, 3rd Edition, 2012, O'Reily
2. *Spark: The Definitive Guide,* Bill Chambers, Matei Zaharia, 1st Edition, 2017, O'Reily
3. *Mining of Massive Datasets,* Jure Leskovec, Anand Rajaraman, 2nd Edition, 2016, Dreamtech Press

Reference Books:

1. Nathan Marz, *Big Data: Principles and Best Practices of Scalable Real-Time Data Systems,* 1st Edition, 2015, Manning Publications
2. Mohammad Guller, *Big Data Analytics with Spark*, 1st Edition, 2015, Apress

Subject Name: Remote Sensing and GIS		**Subject Code: CAP054C206**
L-T-P-C – 4-0-0-4	**Credit Units: 04**	**Scheme of Evaluation: T**

Objective:

The objectives of the course are to make the students understand the fundamental concepts of remote sensing and its applications, learn about GIS (Geographic Information Systems) and spatial data processing, explore satellite image acquisition, preprocessing, and classification techniques etc.

Prerequisites: Basics of Digital Image Processing, Linear Algebra & Probability, Python Programming

Course Outcomes

On successful completion of the course the students will be able to:		
SI No	**Course Outcome**	**Blooms Taxonomy Level**

CO 1	**Understand** the fundamentals of remote sensing and GIS.	BT 2
CO 2	**Process** and **interpret** satellite images for spatial analysis.	BT 3
CO 3	**Analyze** and **assess** GIS solutions for urban planning and disaster management.	BT 4 & 5
CO 4	**Design** AI/ML techniques for remote sensing image classification.	BT 6

Detailed Syllabus:

Module	Topics	Course Content	Periods
I.	**Remote Sensing Fundamentals**	Fundamentals of Remote Sensing, Definition & Historical Development, Electromagnetic Spectrum & Remote Sensing Principles, Energy Interactions with Atmosphere & Earth's Surface, Remote Sensing Platforms & Sensors, Satellite & Aerial Remote Sensing Systems, Optical, Infrared, Microwave, and Hyperspectral Sensors, Types of Satellites: Landsat, Sentinel, MODIS, LIDAR, Resolution in Remote Sensing, Spatial, Spectral, Temporal & Radiometric Resolutions, Sensor Characteristics and Their Applications, Remote Sensing Data Acquisition, Passive vs. Active Remote Sensing, Satellite Data Sources and Accessibility	22
II.	**Image Processing and Interpretation**	Preprocessing of Satellite Images, Radiometric & Geometric Corrections, Image Enhancement Techniques, Image Rectification & Registration, Image Classification Techniques, Supervised & Unsupervised Classification, Machine Learning Approaches in Image Classification, Object-Based Image Analysis (OBIA), Vegetation Indices & Environmental Applications, NDVI (Normalized Difference Vegetation Index), Land Use/Land Cover (LULC) Mapping, Change Detection Techniques, Thermal & Radar Remote Sensing, Thermal Infrared Remote Sensing, Microwave & SAR (Synthetic Aperture Radar) Imaging	22
III.	**Geographic Information System (GIS)**	Fundamentals of GIS, GIS Concepts, Components & Data Models, Spatial Data Representation (Vector & Raster Data), GIS Software (ArcGIS, QGIS, Google Earth Engine), Spatial Data Acquisition & Integration, GPS (Global Positioning System) & Field Data Collection, Remote Sensing Data Integration with GIS, Spatial Analysis & Modeling, Buffering, Overlay, and Proximity Analysis, Network Analysis & Terrain Modeling, 3D GIS and DEM (Digital Elevation Model), Web GIS & Cloud-Based GIS Services, Google Earth Engine & OpenStreetMap, Cloud GIS Technologies (ArcGIS Online, Google Earth Engine)	22
IV	**Applications**	Environmental & Agricultural Applications, Deforestation & Land Degradation Monitoring, Crop Yield Estimation & Precision Agriculture, Urban & Disaster Management, Urban Growth Analysis & Smart Cities, Flood, Earthquake, and Forest Fire Mapping, Climate Change & Hydrological Applications, Glacier &	22

		Coastal Change Detection, Watershed Management & Hydrological Modeling, Artificial Intelligence & Deep Learning in Remote Sensing, AI-Based Image Segmentation, Deep Learning for Land Cover Classification, Real-Time Remote Sensing Applications	
TOTAL			**88**

Remote Sensing and GIS Practice Sessions

Total Practice Hours for the semester = 30 (2 hours per week)

Minimum 10 Laboratory experiments based on the following-

- Download and analyze Landsat/Sentinel satellite images.
- Explore spectral bands and their applications.
- Visualize remote sensing data using GIS software (QGIS/ArcGIS).
- Perform radiometric and geometric corrections on satellite imagery.
- Implement NDVI for vegetation analysis.
- Classify land use using supervised and unsupervised learning methods.
- Create and analyze spatial data using QGIS/ArcGIS.
- Perform spatial interpolation and terrain modeling.
- Develop a simple Web GIS application.
- Perform flood risk analysis using GIS.
- Use machine learning models for land cover classification.
- Develop a GIS-based disaster monitoring system.

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4 * 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH	8 * 4 NCH = 32 NCH (Problem Solving, Seminar, Case Study, Discussion, Internship, Projects)

Text Books

1. *Remote Sensing and Image Interpretation,* Thomas M. Lillesand, Ralph W. Kiefer, Jonathan Chipman, 6th Edition, 2011, Wiley
2. *Introduction to Geographic Information Systems,* Kang-Tsung Chang, 4th Edition, 2017, McGraw Hill Education
3. *Fundamentals of Remote Sensing,* George Joseph, 3rd Edition, 2018, The Orient Blackswan

Reference Books:

1. John A. Richards, *Remote Sensing Digital Image Analysis*, 4th Edition, 2005, Springer
2. Peter A. Burrough, Rachael McDonnell, *Principles of Geographic Information Systems*, 3rd Edition, 2016, Oxford University Press

Subject Name: Computer Vision		**Subject Code: CAP054C207**
L-T-P-C – 4-0-0-4	**Credit Units: 04**	**Scheme of Evaluation: T**

Objective:

To provide students with a comprehensive foundation in computer vision techniques and applications. This course emphasizes image processing, feature detection, object recognition, motion analysis, and the use of deep learning for visual understanding

Prerequisites: Basics of Digital Image Processing, Linear Algebra & Probability, Python Programming

Course Outcomes

On successful completion of the course the students will be able to:		
SI No	**Course Outcome**	**Blooms Taxonomy Level**
CO 1	Understand the fundamental principles of computer vision and digital imaging	**BT 2**
CO 2	Apply techniques for image filtering, edge detection, and segmentation	**BT 3**
CO 3	Implement object detection, tracking, and feature matching algorithms	**BT 4**
CO 4	Analyze and evaluate visual recognition systems	**BT 5**

Detailed Syllabus:

Module	Topics	Course Content	Periods
I.	**Introduction to Image Processing**	Basics of computer vision, applications, camera models, and image formation. Image filtering: Gaussian smoothing, sharpening, thresholding. Color spaces and transformations. Edge detection: Sobel, Canny.	**22**
II.	**Feature Detection**	Corner and blob detection (Harris, DoG). SIFT, SURF, and ORB features. Feature matching: Brute Force, FLANN. Homographies, RANSAC, image stitching.	**22**
III.	**Object Recognition and Motion Analysis**	Segmentation: K-means, Mean-shift, Graph Cuts. Object detection with HOG, Viola-Jones. Tracking: Kalman Filter, Optical Flow, Meanshift, Camshift. Background subtraction and motion segmentation.	**22**
IV	**ML for Computer Vision**	CNN architectures for vision: LeNet, AlexNet, VGG, ResNet. Transfer learning. Object detection using YOLO, SSD. Semantic segmentation: U-Net, FCN. Applications in face recognition, pose estimation, and scene understanding.	**22**
TOTAL			**88**

Computer Vison Practice Sessions

Total Practice Hours for the semester = 30 (2 hours per week)

Minimum 10 Laboratory experiments based on the following-

- 1. Image transformations and filtering using OpenCV
- 2. Edge detection using Sobel and Canny operators
- 3. Feature detection using Harris, SIFT, ORB
- 4. Image stitching using feature matching and homographies
- 5. Object detection using Haar cascades
- 6. Motion tracking with Kalman Filter
- 7. Background subtraction for motion detection
- 8. Image segmentation using k-means clustering
- 9. Face recognition using OpenCV and dlib
- 10. Deep learning with CNNs for image classification
- 11. Object detection using pretrained YOLOv5
- 12. Semantic segmentation with U-Net
- 13. Pose estimation using OpenPose or MediaPipe
- 14. Real-time object detection from webcam
- 15. Mini-project on visual recognition application

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4 * 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH	8 * 4 NCH = 32 NCH (Problem Solving, Seminar, Case Study, Discussion, Internship, Projects)

Textbook:

1. Computer Vision: Algorithms and Applications, Richard Szeliski, 1st Edition, 2010, Springer

Reference Books:

1. Learning OpenCV 4 Computer Vision with Python, Joseph Howse, 4th Edition, 2020, Packt
2. Deep Learning for Vision Systems, Mohamed Elgendy, 1st Edition, 2020, Manning
3. Multiple View Geometry in Computer Vision, Richard Hartley and Andrew Zisserman, 2nd Edition, 2004, Cambridge University Press

Subject Name: Cloud Computing for Big Data		**Subject Code: CAP054C208**
L-T-P-C – 4-0-0-4	**Credit Units: 04**	**Scheme of Evaluation: T**

Objective:

To introduce the concepts and technologies that enable scalable data analytics using cloud platforms. This course covers cloud architecture, storage systems, distributed computing, big data services, and deployment of analytics pipelines on cloud environments such as AWS, Azure, and Google Cloud.

Prerequisites:

Computer Networks, Operating Systems, Basics of Big Data, Python/Java Programming

SI No	Course Outcome	BT Level
CO 1	Understand cloud computing architecture, service models, and infrastructure	BT 2
CO 2	Apply distributed computing frameworks to handle big data in cloud environments	BT 3
CO 3	Design and deploy scalable big data pipelines using cloud-native services	BT 4
CO 4	Evaluate the performance and cost-efficiency of cloud-based big data systems	BT 5

Module	Topics	Course Content	Periods
I.	Introduction to Cloud Computing	Cloud computing definition, characteristics, service models (IaaS, PaaS, SaaS). Deployment models (Public, Private, Hybrid). Virtualization and containers (Docker). Cloud security basics.	22
II.	Big Data Processing Frameworks	Introduction to Hadoop ecosystem: HDFS, YARN, MapReduce. Apache Spark architecture and RDDs. Hive and Pig for data querying. NoSQL (HBase, MongoDB).	22
III.	Cloud Platforms for Big Data	Overview of AWS, Azure, GCP. Services for big data: AWS EMR, S3, Athena, Redshift; Azure HDInsight, Synapse; GCP BigQuery, Dataflow. Data lake and warehouse concepts.	22
IV.	Advanced Cloud Data Analytics	Deploying ML pipelines with cloud services (AWS SageMaker, Azure ML Studio, Google AI Platform). Streaming data with Kafka, Kinesis, and Flink. CI/CD for data engineering in cloud. Cost and scalability optimization.	22
		TOTAL	**88**

Cloud Computing Practice Sessions

Total Practice Hours = 30 (2 hours/week)
Minimum 10 lab experiments selected from:

1. Set up and deploy virtual machines and containers in AWS/GCP
2. Configure and run a Hadoop job on cloud-based cluster
3. Load and query datasets using Hive on AWS EMR
4. Develop and run PySpark programs on cloud notebooks
5. Store and retrieve big data from AWS S3 / Azure Blob Storage
6. Perform SQL queries using Google BigQuery
7. Set up data pipeline using Kafka + Spark Streaming
8. Use cloud-based NoSQL (Firebase/MongoDB Atlas) for data access
9. Deploy machine learning model using SageMaker or GCP AI Platform
10. Implement real-time dashboard using cloud analytics
11. Explore CI/CD tools for cloud-based data pipeline automation
12. Mini-project on building an end-to-end cloud big data solution

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4 * 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH	

Textbook:

1. Cloud Computing: Concepts, Technology & Architecture by Thomas Erl, Pearson, 1st Edition

Reference Books:

1. Cloud Computing for Big Data by Prasad Mukhedkar and Kalpana Sharma, Wiley
2. Hadoop: The Definitive Guide by Tom White, O'Reilly Media
3. Architecting the Cloud by Michael J. Kavis, Wiley
4. Designing Data-Intensive Applications by Martin Kleppmann, O'Reilly

Subject Name: Machine Vision		**Subject Code: CAP054C209**
L-T-P-C – 4-0-0-4	**Credit Units: 04**	**Scheme of Evaluation: T**

Objective:

To introduce the fundamental concepts, techniques, and industrial applications of machine vision. This course emphasizes illumination models, camera calibration, image acquisition, feature analysis, and vision-based inspection systems with an introduction to embedded vision and smart cameras.

Prerequisites:

Basic Linear Algebra, Digital Image Processing, Python Programming

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

SI No	Course Outcome	Blooms Taxonomy Level
CO 1	Explain the core principles of machine vision and image formation	BT 2
CO 2	Apply basic image acquisition and preprocessing techniques	BT 3
CO 3	Develop simple machine vision applications for inspection and measurement	BT 4
CO 4	Analyze and evaluate machine vision systems in industrial settings	BT 5

Detailed Syllabus:

Module	Topics	Course Content	Periods
I.	**Fundamentals of Machine Vision**	Machine vision vs. computer vision. Components: camera, lens, lighting, and frame grabbers. Image sensing and acquisition. Lighting techniques (backlighting, dome, coaxial). Color and grayscale imaging.	22
II.	**Image Processing Basics**	Thresholding, histogram equalization, filtering (Gaussian, median). Morphological operations. Noise removal. Binary image processing.	22
III.	**Feature Analysis and Measurement**	Edge and contour detection. Region properties, blob analysis. Object dimensions, shape, and orientation. Barcode and OCR reading. Geometric transformations and alignment.	22
IV	**Applications and Systems**	Machine vision in quality inspection, robotics, and automation. 2D vs. 3D vision. Stereo vision basics. Smart cameras, embedded systems, vision sensors, and industrial protocols.	22
TOTAL			**88**

Machine Vison Practice Sessions

Total Practice Hours for the semester = 30 (2 hours per week)

Minimum 10 Laboratory experiments based on the following-

Minimum 10 lab experiments from:

1. Image acquisition from USB/industrial camera using OpenCV or MATLAB

2. Illumination variation and image quality analysis
3. Grayscale and binary conversion
4. Morphological operations and region analysis
5. Edge and contour detection
6. Object dimension measurement
7. Barcode and QR code detection
8. OCR-based character recognition
9. Camera calibration using checkerboard patterns
10. Smart camera programming (NI/Matrox/OpenMV)
11. Inspection system simulation for a manufacturing line
12. Mini-project on industrial vision application

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4 * 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH	8 * 4 NCH = 32 NCH (Problem Solving, Seminar, Case Study, Discussion, Internship, Projects)

Textbook:

1. Machine Vision by Ramesh Jain, Rangachar Kasturi, Brian G. Schunck, McGraw-Hill, 1st Edition

Reference Books:

1. Fundamentals of Machine Vision by Harley R. Myler, Prentice Hall
2. Programming Computer Vision with Python by Jan Erik Solem, O'Reilly
3. Machine Vision Handbook by Bruce Batchelor, Springer

6.4 Detailed Syllabus for 3rd Semester:

Subject Name: Software Project Management		**Subject Code: CAP054C301**
L-T-P-C – 3-0-0-3	**Credit Units: 03**	**Scheme of Evaluation: T**

Objective:

To equip students with comprehensive knowledge and practical skills for managing software projects effectively, including planning, execution, monitoring, and delivery using standard project management methodologies.

Prerequisites:

- Basic understanding of software development life cycle
- Familiarity with programming and databases

Course Outcomes

On successful completion of the course, the students will be able to:		
SI No	**Course Outcome**	**Blooms Taxonomy Level**
CO 1	Understand principles of software project management and lifecycle models	**BT 2**
CO 2	Apply project planning and scheduling techniques	**BT 3**
CO 3	Analyze project risks and perform cost estimation	**BT 4**
CO 4	Evaluate quality assurance and configuration management	**BT 5 & 6**

Detailed Syllabus:

Modules	**Topics**	**Course Contents**	**Hours**
I	**Introduction**	Introduction to software project management. Importance, objectives, categories of software projects, responsibilities of a project manager. Overview of software development life cycles (Waterfall, Agile, Spiral).	**22**
II	Project Planning and Estimation	Project planning steps. Work Breakdown Structure (WBS), cost estimation techniques: COCOMO I & II, Function Point. Scheduling: Gantt charts, PERT, and CPM.	**22**
III	Risk Management and Quality Assurance	Risk identification and assessment, risk mitigation strategies. Software quality assurance, quality metrics, ISO 9126, CMMI, Six Sigma in software projects.	**22**
IV	Project Monitoring, Control and Tools	Project tracking, milestone reviews, project audits. Configuration management. Project closure. Project management tools: Microsoft Project, JIRA, GitHub Project Boards.	**22**
	TOTAL		**88**

Software Project Management Practice Sessions

Total Practice Hours for the semester = 30 (2 hours per week)

Laboratory experiments based on the following-

The student will perform a project to reflect software project cycles

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4 * 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH	8 * 4 NCH = 32 NCH (Problem Solving, Seminar, Case Study, Discussion, Internship, Projects)

Textbook:

1. Software Project Management, Bob Hughes, Mike Cotterell & Rajib Mall, 5th Edition, 2012, McGraw-Hill Education.

Reference Books:

1. Applied Software Project Management, Andrew Stellman & Jennifer Greene, 1st Edition, 2005, O'Reilly Media.
2. Managing Software Projects, Gopalaswamy Ramesh, 1st Edition, 2001, McGraw-Hill.
3. The Art of Project Management, Scott Berkun, 1st Edition, 2005, O'Reilly Media.

Subject Name: Network Security and Cryptography		**Subject Code: CAP054C302**
L-T-P-C – 4-0-0-4	**Credit Units: 04**	**Scheme of Evaluation: T**

Objective:

The objectives of the course are to make the students understand fundamental Network security and cryptography concepts

Course Outcomes

On successful completion of the course, the students will be able to:		
SI No	**Course Outcome**	**Blooms Taxonomy Level**
CO 1	**Explain the** fundamental concepts, applications, and ethical implications of Network Security & Cryptography	**BT 2**
CO 2	**Apply** Cryptography algorithms to solve problems	**BT 3**
CO 3	**Analyze** and implement network security protocols.	**BT 4**
CO 4	**Assess** and **design** Network Security & Cryptography Solutions	**BT 5 & 6**

Detailed Syllabus:

Modules	Topics	Course Contents	Hours
I	**Introduction**	Need for Security, Security Approaches, Principles of Security, Types of Attacks, Brute Force Attack, Encryption, Decryption, Cryptosystem, Cryptographic Techniques: Substitution Ciphers, Transposition Ciphers, Product Ciphers, Steganography, Block Cipher, Stream Cipher.	**22**
II	**Symmetric and Asymmetric Key Cryptography**	Overview, Algorithm Modes and Types, Data Encryption Standard: Simplified DES, The Strength of DES, Differential and Linear Cryptanalysis. Triple DES, Blowfish. Confidentiality using Conventional Encryption: Placement of Encryption Function, Traffic Confidentiality, Key Distribution, Random Number Generation. Modular Arithmetic, Public Key Cryptography and RSA: Principles of Public Key Cryptosystems, Difference with Symmetric Key Cryptography, The RSA Algorithms, Key Management, Diffie-Hellman Key Exchange.	**22**
III	**Authentication Protocols**	Message Authentication: Authentication Requirements, Authentication Functions, Message Authentication Codes, MD5 Message Digest Algorithms, Digital Signatures and Authentication Protocols: Digital Signatures, Authentication Protocols, Digital Signature Standards.	**22**
IV	**Security Protocols**	Security Applications and Protocols- Authentication Applications: Secure HTTP, HTTPS, ERT, SSH, Kerberos. Email Security: PGP, S/MIME. IP Security: Overview, IPSec architecture.	**22**
	TOTAL		**88**

Subject Name: Network Security and Cryptography Lab		**Subject Code: CAP054C312**
L-T-P-C – 0-0-2-1	**Credit Units: 01**	**Scheme of Evaluation: P**

Total Practice Hours for the semester = 30 (2 hours per week)

Laboratory experiments based on the following-

1. Implementation of Caesar Cipher and Substitution Cipher
2. Encryption and Decryption using DES and AES
3. Simulating RSA Algorithm
4. Implement Diffie-Hellman Key Exchange
5. Hashing using SHA-256 and MD5
6. Digital Signature Implementation
7. Packet Sniffing and Analysis using Wireshark
8. Simulation of Firewall and VPN
9. SSL/TLS Handshake Demonstration
10. Setup and Configuration of IDS (e.g., Snort)

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4 * 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH	8 * 4 NCH = 32 NCH (Problem Solving, Seminar, Case Study, Discussion, Internship, Projects)

Textbooks:

1. *Cryptography and Network Security*, Atul Kahate, 2nd Edition. 2003, Tata McGraw Hill.
2. *Cryptography and Network security,* Fourozan, 3rd Edition, 2007, McGraw Hill

Reference Books:

1. William Stallings, *Cryptography and Network Security*: *Principles and Practices*", 5th Edition, 2010, Prentice Hall.
2. Michael Howard, David LeBlanc, John Viega, *24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them*, 1st Edition, 2009, McGraw-Hill Osborne Media.

Subject Name: Internet of Things		**Subject Code: CAP054C303**
L-T-P-C – 3-1-0-4	**Credit Units: 04**	**Scheme of Evaluation: T**

Objective:

To provide an advanced understanding of Internet of Things (IoT) systems with a focus on system architecture, real-time analytics, device communication, cloud integration, and security. Students will gain hands-on experience in developing end-to-end IoT solutions suitable for industrial, healthcare, agricultural, and smart city applications.

Prerequisites:

Basic understanding of networking, programming (Embedded C), and microcontrollers.

Course Outcomes:

SI No	Course Outcome	BT Level
CO 1	Comprehend IoT architecture and communication protocols for system design.	BT 1 & 2
CO 2	Apply edge and fog computing techniques for distributed processing.	BT 3 & 4
CO 3	Design secure and scalable cloud-based IoT applications.	BT 5
CO 4	Develop smart IoT prototypes using microcontrollers, sensors, and actuators.	BT 5 & 6

Detailed Syllabus:

Module	Topics	Course Content	Periods
I	**IoT Architecture and Communication**	IoT definitions, evolution, applications, and ecosystems; Reference architectures (Three-layer, Five-layer, SOA); Communication models and protocols (MQTT, CoAP, XMPP, DDS); Wireless standards: BLE, Zigbee, LoRaWAN, NB-IoT; IPv6 and 6LoWPAN.	22
II	**Embedded IoT and Edge Computing**	Microcontroller programming with Arduino/ESP32; Real-time operating systems (RTOS) for IoT; Interfacing of analog and digital sensors; Actuator control; Edge analytics; Fog computing models and case studies.	22
III	**Cloud Integration and Data Analytics**	IoT data lifecycle; Cloud platforms (AWS IoT, Azure IoT Hub, Google Cloud IoT); Storage options, real-time streaming; IoT dashboards; Big data processing using Apache Kafka/Spark; Integrating ML for predictive maintenance and anomaly detection.	22
IV	**Security, Standards and IoT Applications**	IoT vulnerabilities and threat modeling; Lightweight cryptography; Authentication protocols (OAuth2, JWT); Blockchain for IoT security; Global IoT standards and interoperability (oneM2M, IEEE, ITU); Case studies: healthcare, agriculture, industry 4.0.	22

Internet of Things Practice Session

Total Practice Hours for the semester = 30 (2 hours per week)

Laboratory experiments based on the following-

1. Programming ESP32 for sensor interfacing and wireless transmission.
2. IoT device communication using MQTT with broker setup.
3. Developing real-time dashboards on ThingsBoard or Node-RED.
4. Deploying IoT solution on AWS IoT Core with shadow services.
5. Implementing edge analytics using a microcontroller with TinyML.
6. Secure communication using token-based and public key methods.
7. Controlling devices remotely using Blynk or Firebase.

8. Real-time streaming of sensor data to Apache Kafka.

9. Developing smart agriculture system using DHT11, soil moisture sensor.

10. Use of blockchain framework for recording IoT transactions.

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4* 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH(Optional)	

Textbook

1. Internet of Things: Principles and Paradigms, Rajkumar Buyya and Amir Vahid Dastjerdi, 1st Edition, 2016, Morgan Kaufmann.

Reference Books

1. Mastering Internet of Things, Peter Waher, 1st Edition, 2018, Packt.

2. Designing Connected Products: UX for the Consumer Internet of Things, Claire Rowland et al., 1st Edition, 2015, O'Reilly Media.

3. Edge Computing: A Primer, Jie Cao, Quan Zhang, Weisong Shi, 2018, Springer.

Subject Name: Embedded Systems		**Subject Code: CAP054C304**
L-T-P-C – 3-1-0-4	**Credit Units: 04**	**Scheme of Evaluation: T**

Objective:

To develop in-depth knowledge of embedded system architectures, real-time operating systems, peripheral interfacing, and firmware development. The course emphasizes hands-on development of real-time applications using modern microcontroller platforms and embedded tools.

Prerequisites:

Knowledge of digital electronics, microcontrollers, and programming in C/C++.

Course Outcomes:

SI No	Course Outcome	BT Level
CO 1	Understand the architecture and operation of embedded systems.	BT 1 & 2
CO 2	Design and interface embedded hardware components.	BT 3 & 4

CO 3	Implement real-time system functions and scheduling algorithms.	BT 5
CO 4	Develop embedded applications using microcontrollers and RTOS.	BT 5 & 6

Detailed Syllabus:

Module	Topics	Course Content	Periods
I	**Embedded System Architecture and Design**	Definition and characteristics of embedded systems. Embedded system design process and lifecycle. ARM Cortex-M architecture overview. Memory types, buses, interrupts, GPIO. System-on-Chip (SoC) and Embedded Linux overview.	22
II	**Microcontroller Programming and Peripheral Interfacing**	Introduction to 8-bit and 32-bit microcontrollers. Programming in Embedded C. Timer/counter, ADC, DAC, UART, SPI, I2C interfaces. Interfacing LEDs, LCDs, sensors, actuators, and motors with microcontrollers.	22
III	**Real-Time Operating Systems and Scheduling**	Concepts of RTOS, task scheduling, preemptive and cooperative multitasking. Inter-task communication: queues, semaphores, mutex. RTOS services and APIs (FreeRTOS/RTEMS). Real-time debugging and profiling.	22
IV	**Advanced Applications and Embedded Project Development**	Embedded communication protocols (CAN, Modbus, Zigbee). Low-power design techniques. Embedded system testing and validation. Case studies in industrial automation, automotive, and consumer electronics. Capstone project development.	22

Embedded Practice Session

Total Practice Hours for the semester = 30 (2 hours per week)

Laboratory experiments based on the following-

1. LED blinking using timers and delay functions.

2. Interfacing push-button and controlling output using interrupts.

3. Analog sensor (temperature or light) interfacing using ADC.

4. Controlling servo or DC motor with PWM signals.

5. Serial communication between two microcontrollers using UART.

6. Data transmission via SPI/I2C to an external EEPROM or RTC module.

7. RTOS-based multitasking application (FreeRTOS).

8. Implement semaphore and queue in RTOS for task coordination.

9. Embedded project: Smart home controller using sensors and actuators.

10. Design and test a low-power embedded IoT system.

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4* 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH(Optional)	

Textbook

1. Embedded Systems: Architecture, Programming and Design, Raj Kamal, 3rd Edition, 2017, McGraw-Hill Education.

Reference Books

1. Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language and C, Yifeng Zhu, 3rd Edition, 2017, E-Man Press LLC.

2. Real-Time Concepts for Embedded Systems, Qing Li and Caroline Yao, 1st Edition, 2003, CMP Books.

3. The Designer's Guide to the Cortex-M Processor Family, Trevor Martin, 2nd Edition, 2016, Newnes.

6.5 Detailed Syllabus for 4th Semester

Subject Name: Quantum Computing		**Subject Code: CAP054C401**
L-T-P-C – 3-1-0-4	**Credit Units: 04**	**Scheme of Evaluation: T**

Objective:

To introduce undergraduate students to the fundamental principles of quantum computing, quantum mechanics for computation, and the development of quantum algorithms with exposure to quantum programming frameworks.

Prerequisites:

- Linear Algebra and Basic Probability
- Discrete Mathematics and Basic Programming Knowledge

Course Outcomes:

SI No	Course Outcome	BT Level
CO 1	Understand fundamental principles of quantum mechanics relevant to computing	BT 1 & 2
CO 2	Apply quantum gates and circuits to simple computational problems	BT 3 & 4
CO 3	Analyze and simulate quantum algorithms	BT 4 & 5
CO 4	Develop basic quantum programs using frameworks like Qiskit	BT 5

Detailed Syllabus:

Module	Topics	Course Content	Periods
I	**Quantum Computation Foundations**	Introduction to quantum computing. Differences from classical computing. Quantum bits (qubits), quantum states, Dirac notation, superposition and entanglement.	22
II	**Quantum Gates and Circuits**	Single and multi-qubit gates (Pauli, Hadamard, Phase, CNOT, Toffoli). Quantum circuit representation and simplification. Quantum measurement and state collapse.	22
III	**Quantum Algorithms**	Deutsch-Jozsa algorithm, Grover's search algorithm, Shor's factoring algorithm. Quantum teleportation and superdense coding. Overview of quantum supremacy and limitations.	22

IV	**Quantum Programming and Applications**	Introduction to Qiskit and IBM Quantum Experience. Creating and simulating quantum circuits. Applications in cryptography, optimization, and machine learning.	22

Quantum Computing Practice Session

Total Practice Hours for the semester = 30 (2 hours per week)

Laboratory experiments based on the following-

1. 1. Simulate single and multiple qubit operations using Qiskit.

2. Construct and analyze quantum gates and circuits.

3. Implement Deutsch-Jozsa algorithm using IBM Q platform.

4. Run Grover’s search on a simple dataset.

5. Perform quantum teleportation using Qiskit simulator.

6. Simulate quantum entanglement and measurement outcomes.

7. Visualize Bloch sphere representations of qubit states.

8. Create a quantum coin toss experiment.

9. Explore noise and decoherence using Qiskit Aer.

10. Mini project: Quantum algorithm simulation for real-world problem.

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4* 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH(Optional)	

Textbook

1. Quantum Computation and Quantum Information, Michael A. Nielsen and Isaac L. Chuang, 10th Anniversary Edition, 2010, Cambridge University Press.

Reference Books:

1. Quantum Computing: An Applied Approach, Jack D. Hidary, 1st Edition, 2019, Springer.

2. Learn Quantum Computing with Python and Q#, Sarah Kaiser and Christopher Granade, 1st Edition, 2021, Manning Publications.

3. Programming Quantum Computers, Eric R. Johnston et al., 1st Edition, 2019, O’Reilly.

Subject Name: Blockchain Technologies		**Subject Code: CAP054C402**
L-T-P-C – 3-1-0-4	**Credit Units: 04**	**Scheme of Evaluation: T**

Objective:

To introduce students to the principles, architecture, and applications of blockchain technologies. The course will provide knowledge on decentralized systems, consensus algorithms, cryptocurrency, and smart contracts with hands-on exposure to blockchain development environments

Prerequisites:

- Basics of Computer Networks and Cryptography
- Programming fundamentals

Course Outcomes:

SI No	Course Outcome	BT Level
CO 1	Understand blockchain architecture and distributed ledger technology	BT 1 & 2
CO 2	Analyze consensus mechanisms and smart contract frameworks	BT 3 & 4
CO 3	Apply blockchain principles in real-world applications	BT 4 & 5
CO 4	Evaluate the security, scalability, and governance aspects of blockchain	BT 5

Detailed Syllabus:

Module	Topics	Course Content	Periods
I	**Introduction to Blockchain**	History and fundamentals of blockchain, structure of a block, characteristics of blockchain, public vs private vs consortium Blockchain, use cases across industries.	22
II	**Consensus Mechanisms and Cryptography**	Consensus algorithms: Proof of Work (PoW), Proof of Stake (PoS), Practical Byzantine Fault Tolerance (PBFT), Cryptographic principles: hash functions, digital signatures, Merkle trees.	22
III	**Smart Contracts and Ethereum**	Ethereum architecture, EVM, Solidity programming basics, developing and	22

		deploying smart contracts, ERC-20 tokens, Gas, Remix IDE.	
IV	**Blockchain Applications and Challenges**	Blockchain in supply chain, finance, healthcare, identity management. Scalability and interoperability issues. Security, privacy, regulatory and governance challenges.	22

Blockchain Technologies Practice Session

Total Practice Hours for the semester = 30 (2 hours per week)

Laboratory experiments based on the following-

1. Set up and explore a local Ethereum test network using Ganache.
2. Create and deploy a simple smart contract using Solidity.
3. Implement a basic cryptocurrency token using ERC-20 standard.
4. Interact with smart contracts using Web3.js or ethers.js.
5. Build a decentralized voting application on Ethereum.
6. Configure a private blockchain network using Hyperledger Fabric.
7. Evaluate blockchain performance using block explorers and metrics.
8. Simulate double-spending and analyze blockchain defense mechanisms.
9. Use Metamask for interacting with Ethereum DApps.
10. Mini Project: Develop a full-stack decentralized application (DApp).

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4* 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH(Optional)	

Textbook:

1. Mastering Blockchain, Imran Bashir, 3rd Edition, 2020, Packt Publishing.

Reference Books:

1. Blockchain Basics: A Non-Technical Introduction, Daniel Drescher, 1st Edition, 2017, Apress.
2. Blockchain Applications: A Hands-On Approach, Arshdeep Bahga and Vijay Madisetti, 1st Edition, 2017, VPT.

3. Ethereum: Blockchains, Digital Assets, Smart Contracts, Andreas M. Antonopoulos and Gavin Wood, 1st Edition, 2022, O'Reilly.

Subject Name: Soft Computing		**Subject Code: CAP054C403**
L-T-P-C – 3-1-0-4	**Credit Units: 04**	**Scheme of Evaluation: T**

Objective:

To introduce students to the fundamentals of soft computing techniques, including fuzzy logic, genetic algorithms, neural networks, and hybrid systems. The course aims to develop the ability to design and implement intelligent systems capable of learning and adaptation.

Prerequisites:

- Basic Programming Knowledge
- Fundamentals of Mathematics and Logic

Course Outcomes:

SI No	Course Outcome	BT Level
CO 1	Understand the concept of soft computing and its applications	BT 1 & 2
CO 2	Apply fuzzy logic principles for problem solving	BT 3
CO 3	Design neural networks and analyze their performance	BT 4 & 5
CO 4	Implement genetic algorithms for optimization	BT 5 & 6

Detailed Syllabus:

Module	Topics	Course Content	Periods
I	**Introduction to Soft Computing & Fuzzy Logic**	Definition and characteristics of soft computing. Introduction to fuzzy sets and fuzzy logic, fuzzy membership functions, fuzzy rules and fuzzy inference, fuzzy decision-making.	22
II	**Artificial Neural Networks**	Introduction to neural networks, biological neuron, perceptron model, multi-layer feedforward networks, backpropagation algorithm, training of neural networks, applications.	22

III	**Genetic Algorithms**	Introduction to evolutionary computation, genetic algorithm steps, selection, crossover, mutation, fitness function, convergence, applications of GA in optimization problems.	22
IV	**Hybrid Systems & Applications**	Neuro-fuzzy systems, genetic-fuzzy systems, ANN-GA integration. Case studies and real-world applications of soft computing in engineering and decision-making.	22

Soft Computing Practice Session

Total Practice Hours for the semester = 30 (2 hours per week)

Laboratory experiments based on the following-

- Implementation of Fuzzy Membership Functions: Create various membership functions: triangular, trapezoidal, Gaussian.
- Fuzzy Logic Control for Temperature System: Design a fuzzy controller for temperature regulation using fuzzy rules.
- Fuzzy Inference System using Mamdani Method: Build a Mamdani-type FIS for a real-life problem (e.g., fan speed controller).
- Fuzzy Inference System using Sugeno Method: Design a Sugeno-type FIS for mapping inputs to output decisions.
- Fuzzification and Defuzzification: Implement fuzzification and defuzzification for input-output mapping.
- Basic Perceptron Network :Implement a single-layer perceptron for logical operations (AND, OR).
- Multi-Layer Perceptron (MLP) with Backpropagation :Train a neural network for handwritten digit recognition (MNIST or subset).
- Activation Functions Visualization: Plot and compare sigmoid, tanh, and ReLU functions.
- Neural Network for XOR Problem: Solve the XOR classification using a 2-layer MLP.
- Implement Hebbian Learning Rule: Create a simple pattern recognizer using Hebbian learning.
- Implement Genetic Algorithm for Function Optimization: Use GA to maximize or minimize benchmark mathematical functions.
- Travelling Salesman Problem using GA:Apply a genetic algorithm to solve TSP for a small set of cities.
- Neuro-Fuzzy Inference System (ANFIS) :Design and train an Adaptive Neuro-Fuzzy system for a classification task.

Textbook:

1. Soft Computing and Intelligent Systems Design, F. O. Karry and C. De Silva, 1st Edition, 2004, Pearson Education.

Reference Books:

1. Neural Networks, Fuzzy Logic and Genetic Algorithms, S. Rajasekaran and G.A. Vijayalakshmi Pai, 1st Edition, 2003, PHI.

2. Principles of Soft Computing, S. N. Sivanandam and S. N. Deepa, 2nd Edition, 2011, Wiley India.

3. Introduction to Artificial Neural Systems, J. M. Zurada, 1st Edition, 1992, West Publishing Company.

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4* 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH(Optional)	

Subject Name: Edge Computing		**Subject Code: CAP054C404**
L-T-P-C – 3-1-0-4	**Credit Units: 04**	**Scheme of Evaluation: T**

Objective:

This course aims to provide students with a deep understanding of edge and fog computing paradigms, their architecture, communication models, and applications in real-time and latency-sensitive systems. Students will learn to design, deploy, and evaluate edge-based systems using platforms like Raspberry Pi, NVIDIA Jetson, or cloud-edge frameworks.

Prerequisites:

Prerequisites: Knowledge of computer networks, cloud computing, and basic IOT knowledge

Course Outcomes:

SI No	Course Outcome	BT Level
CO 1	Understand the fundamentals and architecture of edge and fog computing.	BT 1 & 2
CO 2	Analyze the role of edge computing in reducing latency and network congestion.	BT 3 & 4
CO 3	Design edge-enabled applications using lightweight computation models.	BT 4 & 5
CO 4	Evaluate different edge deployment architectures and data flow mechanisms.	BT 5

Detailed Syllabus:

Module	Topics	Course Content	Periods
I	**Introduction to Edge Computing**	Motivation, evolution from cloud to edge; comparison of cloud, fog, and edge; edge devices, characteristics, use cases in smart cities, healthcare, and manufacturing.	22
II	**Architecture and Communication**	Edge architecture models: hierarchical vs. distributed, cloud-fog-edge continuum; edge gateways; communication protocols (MQTT, CoAP, HTTP, AMQP); latency and bandwidth considerations.	22
III	**Resource Management and Security**	Resource discovery, task scheduling, containerization at edge (Docker), load balancing; security & privacy challenges in edge, trusted execution environments (TEE), lightweight encryption, authentication models.	22
IV	**Edge Platforms and Applications**	Edge orchestration (KubeEdge, Azure IoT Edge), Open Horizon, TensorFlow Lite, edge ML models; case studies: autonomous vehicles, industrial IoT (IIoT), smart surveillance, edge analytics, streaming data processing.	22

Edge Computing Practice Session

Total Practice Hours for the semester = 30 (2 hours per week)

Laboratory experiments based on the following-

1. Set up an edge computing environment using a Raspberry Pi or emulator.
2. Implement MQTT-based data transmission from an IoT device to an edge server.
3. Build a latency comparison between edge and cloud processing using Python/Node.js.
4. Deploy a containerized application using Docker on an edge node.
5. Implement a face detection model using TensorFlow Lite at the edge.
6. Collect sensor data and process using edge analytics before cloud upload.
7. Configure KubeEdge for microservice orchestration.
8. Simulate fog computing using Cisco Packet Tracer or iFogSim.
9. Perform a comparative study on different communication protocols for edge.
10. Implement a security mechanism like token-based authentication in an edge application

Credit Distribution		
Lecture/ Tutorial	**Practicum**	**Experiential Learning**
4* 22 NCH = 88 NCH	2 * 15 NCH = 30 NCH(Optional)	

Textbook

1. Fog and Edge Computing: Principles and Paradigms, Rajkumar Buyya, Satish Narayana Srirama, 1st Edition, 2019, Wiley.

Reference Books

1. Edge Computing: From Hype to Reality, Perry Lea, 1st Edition, 2020, Packt Publishing.

2. Architecting the Cloud to Edge, Victor Muntés, 2023, Springer.

3. Edge AI, Xiaofei Wang and Min Chen, 2021, Springer.