COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

PROGRAMME STRUCTURE & SYLLABUS
[2023 ADMISSIONS ONWARDS]

Five Year Integrated M.Sc. in Computer Science (Artificial Intelligence and Data Science)

1. Introduction

Artificial Intelligence (AI) and Data Science are the two most important technologies in the world today. Data science is emerging as a field that is revolutionizing science and industries alike. Work across nearly all domains is becoming more data-driven, affecting both the available jobs and the required skills. As more data and ways of analyzing them become available, more aspects of the economy, society, and daily life will depend on data. On the other hand, Artificial Intelligence aims at a broader goal of enabling machines to behave like human beings. Artificial Intelligence makes the use of algorithms to perform autonomous actions. Contemporary AI Algorithms like deep learning understand the patterns and find the goal embedded in the data. While data science builds models with statistical insights, AI is about imparting autonomy to the models that emulate cognition and human understanding.

AI & Data Science and are progressing expeditiously in the countries that want to become a knowledge economy. The proposed program aims to produce students with a sound understanding of the fundamentals of theory and practice of data science and AI. It also aims to enable students to become leaders in the industry and academia nationally and internationally and meet the nation's pressing demands. The program will enable Cochin University as the State's nexus point for collaborative efforts spanning higher education institutions, governmental agencies, industry, and nonprofits/foundations to improve the nation's capacity in data science by investing in the development of human capital and infrastructure.

2. Overview of the Course

AI & Data Science Programme prepares students to perform intelligent data analysis, which is a critical component in numerous real-world applications. Data science has emerged as one of the most high-growth, dynamic, and lucrative careers in technology during the past ten years. This program provides rigorous training in data science, machine learning, and data modeling with intensive industry-based exposure. This program aims to equip students with multidisciplinary skills across computer science, machine learning, statistics, and logic. Data scientists have career opportunities in healthcare, business, e-Commerce, social networking companies, climatology, biotechnology, genetics, and other vital areas. The central focus is to equip the students with statistical, mathematical reasoning, machine learning, knowledge discovery, and visualization skills.

3. Our Strength

The program prepares the students to grow future leaders in the AI and Data Science landscape. The program needs to cater to the interdisciplinary requirements to fulfill the designed goal. CUSAT has strong departments in the subject areas required to support this program. The faculty members' expertise from the Statistics, Mathematics, and other science and technology disciplines are an integral part of the program. The students are to take inter-departmental electives courses as part of the program. The strong base of alumni spread over the IT industry and various R&D laboratories across the country and abroad are ready to help the students in various capacities. The research culture prevailing in the postgraduate departments are also the core strength of the university.

4. Industry Collaboration

The linkages between Universities and Industries are increasingly crucial for various spheres of the economy, such as skill development, innovations, technology transfer, promotion of entrepreneurship, and start-ups. The University has always been at the forefront to realize the support from key IT Industry partners. The University's core strengths are the University's long-term associations with R & D laboratories of DRDO like NPOL, CAIR, LRDE, and VSSC. The research collaborations with International partners will add value to the programme. The proposed program amplifies students' opportunities to grab positions as Data Scientist, Data Analyst, Machine learning architect, and ML engineer, among other R & D positions in the research organizations. The following action plans help to achieve these goals:

Fostering regular interaction with the Industry: The University will invite industry and R & D experts to conduct seminars, workshops, and conferences regularly. Participation in industry-led programs by the students will also be encouraged.

Inviting Visiting faculties from industry to engage courses: The University has provided schemes to appoint industry and R&D experts as visiting faculty to engage courses that need special attention.

Continued Mentorship: The Industry and R&D experts shall monitor the students' progress from the beginning of the program enabling the students to get acquainted with industry trends and the work culture followed in the industry.

Mandatory Industrial training to the students: The students have to undergo at least two industrial training of at least two months' duration during the program's early stage and help them carry out the final year projects.

Industry Assessment of students: Opportunities are open to industry and R&D to assess the students during the program to ensure the University is achieving the program's specified objectives.

Industry Interaction and training of faculty members: The faculty members are encouraged to spend time to visit and establish constant interaction with the experts from the industry. This visit would expose the industry practices to elaborate in their classrooms to make their teachings based on a practical approach and theory application. Simultaneously, the faculty may help the industry solve their problems through their breadth of knowledge and experience.

5. Intake and Semesters

The present intake for the M.Sc. (Five Year Integrated) in Computer Science (Artificial Intelligence & Data Science) is 20. The total number of semesters for the programme will be 10. The programme offers two exit options - one at the end of the sixth semester and the other at the end of the eighth semester. The candidates who successfully complete six semesters of the programme have a provision to leave the course with a B.Sc Degree in Computer Science.

The candidates who successfully complete eight semesters of the programme have a provision to leave the course with a B.Sc (Honors) Degree in Computer Science without completing the 9^{th} and 10^{th} semesters. They also have a provision to exit the course after finishing the 8^{th} semester with a B.Sc (Honors with Research) in Computer Science Degree provided they complete a 12 credit Dissertation in lieu of the three elective courses in the 8^{th} semester. Those students who opt for B.Sc (Honors with Research) in Computer Science Degree shall audit an extra course on Research Methodology that is approved by the Department Council.

The only candidates who successfully complete the ten semesters of the programme are eligible for the M.Sc. (Five Year Integrated) Degree in Computer Science (Artificial Intelligence & Data Science).

6. Lateral Entry

Candidates with three year B.Sc Degree in Computer Science / BCA / B.Voc. Business Process and Data Analytics or equivalent awarded by any of the UGC recognized institution with not less than 60% marks, will be considered for admission to the 4th year of the M.Sc. (Five Year Integrated) in Computer Science (Artificial Intelligence & Data Science) programme provided vacant seats are available.

7. Eligibility

Candidates with a minimum of 60% marks or 6.5 CGPA in +2 level of education (Intermediate, CBSE/ICSE/HSC/ All State Boards or Equivalent) with science subjects Physics, Chemistry, and Mathematics are eligible to apply.

8. Admission

Candidates possessing the KVPY Scholarship will be directly admitted to the programme if they have the required eligibility given above. Such candidates need not have to appear for the Common Admission Test (CAT). But they have to submit the requisite application and should pay the application fee. All other eligible candidates will be called to appear for a Common Admission Test (CAT) to be held at various centers. After the entrance examination results are published, the admission will be made through a common counseling process.

9. Entrance Examination

The admission to the programme will be based on the Common Admission Test (CAT) conducted by CUSAT for the under-graduate programmes.

SYLLABUS
FOR
OUTCOME BASED EDUCATION

Five Year Integrated M.Sc. in Computer Science (Artificial Intelligence and Data Science)

For the student admissions starting from the academic year 2023-2024

Five Year Integrated M.Sc. in Computer Science (Artificial Intelligence and Data Science)

Program Outcomes (PO)

After the completion of M.Sc. programme, the students will be able to:

PO1: Apply knowledge of fundamental concepts of computing and abstract out computational models for the problems to be solved.
PO2: Elicit knowledge and methodologies from a large knowledge base including research literature, white papers, and/or prior-art documents.
PO3: Demonstrate thinking and problem-solving skills applied to the intersection of data science, AI, and managerial challenges.
PO4: Understand economic, societal, business and ethical impacts of Artificial Intelligence.
PO5: Master critical thinking tools and create successful, actionable and innovative strategies
PO6: Make clear judgments on the appropriateness of the applications of techniques and tools for given computing problems with a fair understanding of the limitations.
PO7: Commit hard to professional ethics and cyber regulations, responsibilities, and norms of professional computing practices.
PO8: Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional.
PO9: Communicate effectively with the computing community, and with society at large, about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.
PO10: Function effectively as an individual and as a member or leader in diverse teams.
PO11: Develop a disruptive entrepreneurship spirit.
PO12: Identify a timely opportunity and use innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.

PROGRAMME STRUCTURE AND SYLLABUS (2023 ADMISSIONS)

Five Year Integrated M.Sc. in Computer Science (Artificial Intelligence and Data Science)

Semester - I

Sl. No.	Course code	Course Title	C/E	Cr	Lr	L/T	M
1	23-813-0101	Mathematics for Data Science	C	4	4	1	100
2	23-813-0102	Communicative English	C	4	4	1	100
3	23-813-0103	Object Oriented Programming	C	4	4	1	100
4	23-813-0104	Computational Thinking for Problem Solving	C	4	4	1	100
5	23-813-0105	Environmental Studies	C	4	4	1	100
6	23-813-0106	Lab 1 - Python Programming Lab	C	1	0	4	100
7	23-813-0107	Lab 2 - C++ Programming Lab	C	1	0	4	100
Total for Semester I				22	20	13	700

Semester - II

Sl. No.	Course code	Course Title	C/E	Cr	Lr	L/T	M
1	23-813-0201	Linear Algebra	C	4	4	1	100
2	23-813-0202	Data Structures	C	4	4	1	100
3	23-813-0203	Introduction to Artificial Intelligence	C	4	4	1	100
4	23-813-0204	Operating Systems	C	4	4	1	100
5	23-813-0205	Java Programming	C	4	4	1	100
6	23-813-0206	Lab 3 - Data Structures Lab	C	1	0	4	100
7	23-813-0207	Lab 4 - Java Programming Lab	C	1	0	4	100
Total for Semester II				22	20	13	700

Semester - III

Sl. No.	Course code	Course Title	C/E	Cr	Lr	L/T	M
1	23-813-0301	Design & Analysis of Algorithms	C	4	4	1	100
2	23-813-0302	Probability and Statistics for Data Science	C	4	4	1	100
3	23-813-0303	Mathematics for Machine Learning	C	4	4	1	100
4	23-813-0304	Database Systems	C	4	4	1	100
5	23-813-0305	Theory of Computation	C	4	4	1	100
6	23-813-0306	Lab 5 - Algorithms Lab	C	1	0	4	100
7	23-813-0307	Lab 6 - Database Systems Lab	C	1	0	4	100
Total for Semester III				22	20	13	700

Semester - IV

Sl. No.	Course code	Course Title	C/E	Cr	Lr	L/T	M
1	23-813-0401	Foundations of Data Science	C	4	4	1	100
2	23-813-0402	Numerical Methods	C	4	4	1	100

3	23-813-0403	Digital Signal Processing	C	4	4	1	100
4	23-813-0404	Agile Software Engineering	C	4	4	1	100
5	23-813-0405	Optimization Techniques	C	4	4	1	100
6	23-813-0406	Lab 7 - Numerical Methods Lab	C	1	0	4	100
7	23-813-0407	Lab 8 - Optimization Techniques Lab	C	1	0	4	100
Total for Semester IV				22	20	13	700
Semester - V							
Sl. No.	Course code	Course Title	C/E	Cr	Lr	L/T	M
1	23-813-0501	Regression Analysis	C	4	4	1	100
2	23-813-0502	Big Data Analytics	C	4	4	1	100
3	23-813-0503	Cloud Computing	C	4	4	1	100
4	23-813-0504	R for Data Science	C	4	4	1	100
5	23-813-0505	Number Theory and Cryptography	C	4	4	1	100
6	23-813-0506	Lab 9 - R for Data Science Lab	C	1	0	4	100
7	23-813-0507	Lab 10 - Data Analytics Lab	C	1	0	4	100
Total for Semester V				22	20	13	700
Semester - VI							
Sl. No.	Course code	Course Title	C/E	Cr	Lr	L/T	M
1	23-813-0601	Inferential Statistics	C	4	4	1	100
2	23-813-0602	Machine Learning Algorithms	C	4	4	1	100
3	23-813-0603	Feature Engineering	C	4	4	1	100
4	23-813-0604	Soft Computing Techniques	C	4	4	1	100
5	23-813-0605	Parallel Computing	C	4	4	1	100
6	23-813-0606	Lab 11 - Machine Learning and Parallel Computing Lab	C	1	0	4	100
7	23-813-0607	Project	C	4	0	4	100
Total for Semester VI				25	20	13	700
Total Credit for B.Sc in Computer Science: 135				Total Marks: 4200			
Semester - VII							
Sl. No.	Course code	Course Title	C/E	Cr	Lr	L/T	M
1	23-813-0701	Computational Linguistics	C	4	4	1	100
2	23-813-0702	Digital Image and Video Processing	C	4	4	1	100
3	23-813-0703	Deep Learning	C	4	4	1	100
4	23-813-0704	Lab 12 - Computational Linguistics Lab	C	1	0	4	100
5	23-813-0705	Lab 13 - Image and Video Processing Lab	C	1	0	4	100
6		Elective - I	E	4	4	1	100
7		Elective - II	E	4	4	1	100
Total for Semester VII				22	20	13	700
Electives							

23-813-0706: Reinforcement Learning
23-813-0707: Computer Vision
23-813-0708: Virtualized Systems
23-813-0709: Advanced Optimization Techniques
23-813-0710: Bioinformatics
23-813-0711: Algorithms for Modern Data Models
23-813-0712: Complex Network Analysis

Semester - VIII

Sl. No.	Course code	Course Title	C/E	Cr	Lr	L/T	M
1	23-813-0801	Probabilistic Graphical Models	C	4	4	2	100
2	23-813-0802	Algorithms for Massive Datasets	C	4	4	2	100
3	23-813-0803	Professional Communication	C	2	2	1	100
4	23-813-0804	Mini Project	C	1	0	4	100
5		Elective - III	E	4	4	1	100
6		Elective - IV	E	4	4	1	100
7		Elective - V	E	4	4	1	100
8	23-813-0813	Dissertation	E	12*		15*	300*
Total for Semester VIII				23/23*	22/10*	12/24*	700

Total Credit for B.Sc (Honours) in Computer Science : 180 Total Credit for B.Sc (Honours with Research) in Computer Science : 180* * Out of 180 credits, 12 credits shall be earned through a research-oriented Dissertation work in lieu of Electives III, IV and V.	Total Marks : 5600

Electives
23-813-0806 : Algorithmic Game Theory
23-813-0807: Deep Learning for Computer Vision
23-813-0808: Natural Language Processing with Deep Learning
23-813-0809: Image and Video Coding
23-813-0810: Functional Programming
23-813-0811: Information Retrieval and Web Search
23-813-0812: Human Computer Interaction
23-813-0813: Cyber Physical Systems

Semester - IX

Sl. No.	Course code	Course Title	C/E	Cr	Lr	L/T	M
1	23-813-0901	Research Methodology	C	4	4	2	100
2	23-813-0902	Project & Viva Voce	C	12	0	16	300
3	23-813-0903	Elective - VI*	E	4	4	2	100
Total for Semester IX				20	8	20	500

* Students can choose any of the MOOC courses approved by the Department Council. The approved MOOC courses will be numbered as 21-813-0903-M1, 21-813-0903-M2, etc.

Semester - X

Sl. No.	Course code	Course Title	C/E	Cr	Lr	L/T	M
1	23-813-1001	Project & Viva Voce	C	20	0	20	500
Total for Semester X				20	0	20	500

Total credits for Integrated M.Sc in Computer Science (Artificial Intelligence & Data Science) Degree: 220	**Total Marks**	**6600**

23-813-0101: Mathematics for Data Science - I

Core/Elective: **Core** Semester: **1** Credits: **4**

Course Description

The aim of this course is to understand elementary mathematics that are the backbone of Computer Science. The course will introduce sets, relations, propositional and predicate calculus, probability, vector calculus and elementary graph theory, with an emphasis on applications in the Computer Science domain.

Course Outcomes (CO)

After the completion of the course, the students will be able to:
CO1: Learn basic concepts of set theory, arithmetic, logic and binary relations.
CO2: Understand and interpret graphs of various types and their applications.
CO3: Model problems in Computer Science using graphs and trees.
CO4: Represent vectors analytically and geometrically, and compute dot and cross products for presentations of lines and planes
CO5: Analyze vector functions to find derivatives, tangent lines, integrals, arc length, and curvature
CO6: Apply derivative concepts to find tangent lines to level curves and to solve optimization problems

Course Content

1. Sets, principle of inclusion and exclusion, relations, equivalence relations and partition, denumerable sets, partial order relations, Mathematical Induction, Permutations and combinations, Probability - Random variables, sum and product rules of probability – A posteriori probabilities – identities of conditional probability – independence – mutual independence – birthday paradox – random variables – indicator random variables.

2. Graphs, definition, types of graphs, paths and circuits. Eulearian and Hermitian circuits. Seven bridges machine, shortest path traveling salesman problems. Planar graph. Matrix representation of graph: adjacency matrix, incidence matrix, circuit matrix, cut set matrix, path matrix, Directed Graphs, Trees, Minimum Spanning Tree of a Graph

3. Vectors and Geometry of Space - Vectors in Space, The Dot and Cross Product of Two Vectors, Lines and Planes in Space, Distances in Space. Surfaces in Space, Cylindrical and Spherical Coordinates Vector Valued Functions - Differentiation and Integration of Vector-Valued Functions, Arc Length and Curvature

4. Functions of several variables - Limits and Continuity, Partial Derivatives, Differentials, Chain Rules for Functions of Several Variables, Directional Derivatives and Gradients, Tangent Planes and Normal Lines, Extrema of Functions of Two Variables, Lagrange Multiplier

5. Applications of graph theory in analysing a social network, Application of calculus in optimization, Solving systems of linear equations, practical aspects of knowing probability, sampling and data analysis

References

1. Kenneth H. Rosen: Discrete Mathematics and Its Applications, McGraw-Hill, 7e, 2011.
2. Joe L. Mott , Abraham Kandel & Theodore P. Baker: Discrete Mathematics for Computer Scientists & Mathematicians, PHI, 2e, 2002.
3. C.L.Liu, D. P. Mohapatra: Elements of Discrete Mathematics: A Computer Oriented Approach, McGraw Hill Education, 4e, 2017.
4. Seymour Lipschutz, John Schiller: Finite Mathematics, McGraw-Hill, 1994.Ron Larson, Bruce H. Edwards, Calculus, 11e, Cengage, 2018
5. Gilbert Strang, Calculus, 3e, Wellesley-Cambridge Press, 2017
6. George B. Thomas, Jr., Thomas' Calculus, 14e, 2017

23-813-0102: Communicative English

Core/Elective: **Core** Semester: **1** Credits: **4**

Course Description

This course is designed to enhance the learner's communication skills by giving adequate exposure in listening, speaking, reading and writing skills and the related sub-skills. This course helps to build up the learners confidence in oral and interpersonal communication by reinforcing the basics of pronunciation specially focusing on interviews / corporate meetings / international business travels

Course Outcomes (CO)

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

CO1: Demonstrate communicative competence

CO2: Use proper pronunciation, structure, style of the English Language

CO3: Understand the employment opportunities, challenges and job roles.

CO4: conduct independent surveys, collect and analyze data, prepare and present reports and projects

Course Content

1. Comprehension: Comprehension includes understanding the language by reading & listening. Passages or poems will be read out in the class and questions will be asked verbally to evaluate level of comprehension: Talks, reports, and poems.

2. Writing skills: In this section students will be exposed to various techniques of writing a paragraph, report composition, application and letters. Paragraph Writing-Objective, Introduction, the topic sentence, developing the topic.

3. Note Making/Talking: - Objective, introduction, How to read, specimen notes, reduction devices, heading and subroutine points. Report writing: - Reporting Events, Reporting interviews, Reporting Surveys.

4. Letter Writing- Personal letter, Business letters, Objectives, Introduction and format of the letter.

5. Functional Grammar: - Grammar will be taught in a functional, integrated and informal way giving more stress on the usage rather than defining.Parts of speech, agreement of the verb with the subject. Subject and predicate. Transformation of sentences – interchange of active and voice, interchange of affirmative and negative sentences.

References:

1. Wren and Martin, High School English Grammar and Composition Book, S Chand, 2018
2. Windshuttle, Keith and Elizabeth Eliot, Writing, Researching and Communicating : Communication Skills for the Information Age. 3e, Tata McGraw-Hil, 1999
3. Francois Cooren, James R. Taylor, Elizabeth J. Van, Every Communication as Organizing: Empirical and Theoretical Explorations in the Dynamic of Text and Conversation, Routledge Communication Series, 2006
4. Ramesh R Kulkarni and Mr. Rangappa Yaraddi, Business Communication: A text for UG and PG students, Notion press, 2017

23-813-0103: Object Oriented Programming

Core/Elective: **Core** Semester: **1** Credits: **4**

Course Description

This course will provide the students an exposure to the concept of object-oriented programming paradigm, focusing on the definition and use of classes along with the fundamentals of object-oriented design. The course will emphasize understanding and implementation of applications using object-oriented techniques. Topics covered include classes, overloading, data abstraction, information hiding, encapsulation, inheritance, polymorphism, templates and exception handling.

Course Outcomes (CO)

After the completion of the course, the students will be able to:
CO1: Understand the procedural and object oriented paradigm with concepts of streams, classes, functions, data and objects.
CO2: Understand dynamic memory management techniques using constructors and destructors.
CO3: Describe the concept of Inheritance, function overloading, operator overloading, virtual functions and polymorphism.
CO4: Understand the usage of templates, generic programming, exception handling and dynamic objects.
CO5: Demonstrate the use of various OOPs concepts with the help of programs.

Course Content

1. Introduction to OOP – Evolution of object oriented languages – Need of Objects – Definition of Object-Oriented Language – Classes and Objects – Creating and Using Classes and objects – Member functions and variables – Abstract data types (ADT) – Encapsulation - Typed and untyped languages - Coupling and cohesion - Constructors and Destructors.

2. Inheritance and Access Control – Member access control in classes – Friend functions and classes – Extending classes – Public Private and Protected Inheritance – Classification of Inheritance – Single – Multiple – Multilevel – Hierarchical – Hybrid.

3. Constructors, Parameterized Constructors, Copy Constructors, Dynamic Constructors, Destructors - Defining Operator Overloading, Overloading Operators, Rules for Overloading Operators, Type Conversions.

4. Polymorphism – Runtime and compile time polymorphism – overloading functions and operators – selecting friend member function for operator overloading – Virtual methods – pure virtual methods – Abstract classes – Defining and using of virtual methods, pure virtual methods and abstract classes – applications of abstract classes.

5. Advanced Concepts- Virtual Destructors – Virtual Base Classes – Template classes – Creating and using templates – Namespaces – Exception Handling - Dynamic Objects – Dynamic object allocation – Inline functions.

References

1. Bjarne Stroustrup: C++ Programming Language, 4e, Addison-Wesley, 2013.
2. Bjarne Stroustrup: Programming: Principles and Practice Using C++, 2e, Addison-Wesley, 2014.
3. Stanley Lippman, Josée Lajoie, Barbara Moo: C++ Primer, 5e, Addison-Wesley, 2012.
4. Paul Deitel, Harvey Deitel: C++ How to Program, 10e, Pearson, 2016.
5. Timothy Budd: Introduction To Object-Oriented Programming, Pearson Education, 2008.
6. Walter J. Savitch, Kenrick Mock: Problem Solving with C++, 9e, Pearson Education, 2017.
7. Ira Pohl: Object-Oriented Programming Using C++, 2e, Addison-Wesley,1996.

23-813-0104 Computational Thinking for Problem Solving

Core/Elective: **Core** Semester: **1** Credits: **4**

Course Description:

Computational Thinking (CT) refers to the thought processes involved in expressing solutions as computational steps or algorithms that can be carried out by a computer. This course introduces Computational thinking which requires understanding the capabilities of computers, formulating problems to be addressed by a computer, and designing algorithms that a computer can execute. Computational thinking is at the heart of computer science practices and is intrinsically connected with it.

Course Outcomes (CO)
After the completion of the course, the students will be able to:

CO1: Recognizing and Defining Computational Problems
CO2: Relate CT with Science and Engineering disciplines
CO3: Developing and Using Abstractions
CO4: Creating Computational Artifacts
CO5: Testing and Refining Computational Artifacts

Course Content
1. Computational thinking – Definitions, Core concepts, Use of computational thinking, examples

2. Logic and algorithmics thinking – Logical thinking, Inductive and deductive arguments, Boolean logic, Symbolic logic. Algorithmic thinking – Intuition vs precision, defining algorithms, examples. Avoiding mistakes

3. Problem-solving and decomposition – Systematic approach, Defining the problem, Devising a solution – Decomposition, heuristic approach, other effective strategies, Patterns and generalization

4. Abstraction and Modeling – Abstraction, layers of abstraction, examples. Modeling – Static and Dynamic models, Modeling data, Modeling interaction

5. Errors – Designing out the bugs, Mitigating errors, testing, debugging, fixing errors. Evaluating a solution – Solution evaluation in terms of correctness, efficiency, elegance, usability. Trade offs. Computational thinking in software development

References:

1. Karl Beecher, Computational Thinking – A beginners guide to problem solving and programming, BCS, 1e, THE CHARTERED INSTITUTE FOR IT, 2017
2. Peter J. Denning, Matti Tedre, Computational Thinking, MIT Press, 2019
3. Peter William Mcowan, Paul Curzon, Power Of Computational Thinking, World Scientific, 2017

23-813-0105: Environmental Studies

Core/Elective: **Core** Semester: **1** Credits: **4**

Course Description

The Environmental Studies facilitate students' understanding of complex environmental issues from a problem-oriented, interdisciplinary perspective

Course Outcomes (CO)

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

CO1: Understand the term "Environmental Studies" its multidisciplinary nature, scope and importance.
CO2: Describe the structure and function of ecosystem; biogeochemical cycles and processes; Ecosystem services and its restoration
CO3: Describe environmental pollution; their sources, causes and effects.
CO4: Develop critical thinking for shaping strategies (scientific, social, economic and legal) for environmental protection and conservation of biodiversity, social equity and sustainable development.
CO5: Acquire values and attitudes towards understanding complex environmental-economic social challenges, and participating actively in solving current environmental problems and preventing the future ones.
CO6: Adopt sustainability as a practice in life, society and industry

Course Content

1. Introduction to Environmental Studies: Multidisciplinary nature of environmental studies; Scope and importance; Concept of sustainability and sustainable development

2. Ecosystem: Definition and concept of Ecosystem: Structure of ecosystem (biotic and abiotic components); Functions of Ecosystem – Physical (energy flow), Biological (food chains, food web, ecological succession) and Biogeochemical (nutrient cycling) processes. Concepts of productivity, ecological pyramids and homeostasis; Types of Ecosystem – Tundra, Forest, Grassland, Desert, Aquatic (ponds, streams, lakes, rivers, oceans, estuaries) – their importance and threats on them with relevant examples from India Ecosystem services (Provisioning, Regulating, Cultural and Supporting). Basics of Ecosystem restoration

3. Environmental pollution: Environmental pollution (Air, water, soil, thermal and noise): causes, effects and controls; Air and water quality standards Nuclear hazards and human health risks Solid waste management: Control measures of urban and industrial waste Pollution case studies: Ganga Action plan (GAP), Delhi air pollution and public health issues etc.

4. Human Communities and the Environment: Human population growth: Impacts on environment, human health and welfare Resettlement and rehabilitation of project affected

persons; case studies Disaster management: floods, earthquake, cyclones and landslides Environmental movements: Chipko movement, Silent valley movement, Bishnois of Rajasthan, Narmada Bachao Andolan etc Environment justice: National Green Tribunal and its importance Environmental ethics: Role of Indian and other religions and cultures in environmental conservation Environmental communication and public awareness, case studies (e.g., CNG vehicles in Delhi, Swachh Bharat Abhiyan)

5. Global Environmental Issues and Policies: Climate change, Global warming, Ozone layer depletion, Acid rain and impacts on human communities and agriculture International agreements: Earth Summit, UNFCCC, Montreal and Kyoto protocols and Convention on Biological Diversity (CBD) 4 Sustainable Development Goals and India's National Action Plan on Climate Change Environment legislation in India: Wildlife Protection Act, 1972; Water (Prevention and Control of Pollution) Act, 1974; Forest (Conservation) Act 1980, Air (Prevention & Control of Pollution) Act, 1981; Environment Protection Act, 1986; Scheduled Tribes and other Traditional Forest Dwellers (Recognition of Forest Rights) Act, 2006

References

1. N.H. Gopal Dutt, Environmental Pollution & Control, 1e, NK, 2016
2. Daniel B. Botkin and Edward A. Keller, Earth as a living planet, 9e, John Wiley, 2014
3. Mackenzie L. Davis and Susan J. Masten, Principles of Environmental Engineering and Science, 1e, McGrawHill, 2003.
4. S.E Jorgensen and I. Johnsen, Principles of Environmental Science and Technology, 2e, Elsevier, 1989.

23-813-0106: Lab 1 - Python Programming Lab

Core/Elective: **Core** Semester: **1** Credits: **1**

Course Description

This course is designed to teach students the basic programming concepts in python.

Course Outcomes (CO)

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

CO1: Write, test and debug simple python programs
CO2: Implement python programs with conditional statements and loops
CO3: Develop Python programs stepwise by defining functions and calling them
CO4: Use Python lists, tuples, dictionaries for representing compound data
CO5: Read and write data from/to files in Python

Course Content

- Compute the GCD of two numbers using Euclidean algorithm
- Find the square root of a number
- Exponentiation (power of a number)
- Find the maximum and second maximum of a list of numbers
- Linear search
- Sorting of numbers and text
- First n prime numbers
- Multiply matrices
- Programs that take command line arguments (word count)
- Find the most frequent words in a text read from a file
- Simulate elliptical orbits in Pygame
- Simulate bouncing ball using Pygame

23-813-0107: Lab 2 - C++ Programming Lab

Core/Elective: **Core** Semester: **1** Credits: **1**

Course Description

This course is designed to teach students the basic programming concepts in C++

Course Outcomes (CO)

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

CO1: Write, test and debug simple C++ programs
CO2: Implement C++ programs with conditional statements and loops
CO3: Develop Python programs stepwise by defining functions and calling them
CO4: Use C++ Classes for representing compound data
CO5: Read and write data from/to files in C++

Course Content

- Compute the GCD of two numbers using Euclidean algorithm
- Find the square root of a number
- Exponentiation (power of a number)
- Find the maximum and second maximum of a list of numbers
- Linear search
- Sorting of numbers and text
- First n prime numbers
- Multiply matrices
- Programs that take command line arguments (word count)
- Find the most frequent words in a text read from a file
- Simulations
- Simulate bouncing ball

23-813-0201: Linear Algebra

Core/Elective: **Core** Semester: **2** Credits: **4**

Course Description

Linear Algebra is at the heart of data science and machine learning concepts. This course is intended primarily to prepare students for calculus. The aim of this course is to prepare students to demonstrate competence with the basic ideas of linear algebra including concepts of linear systems,independence, theory of matrices, linear transformations, bases and dimension, eigenvalues, eigenvectors and diagonalization

Course Outcomes (CO)

After the completion of the course, the students will be able to:
CO1: Determine if a system of equations is consistent and find its general solution
CO2: Row-reduce a matrix to reduced echelon form
CO3. Apply solution methods of linear system for various problems
CO4: Combine row reduction and cofactor expansion to compute a given determinant
CO5: Find the characteristic polynomial of a given matrix
CO6: Given a matrix and an eigenvalue, find the basis for the corresponding eigenspace
CO7: Appreciate the applications of linear algebra

Course Content

1. Systems of linear equations and matrices – Gaussian elimination, Matrices and matrix operations, Invertible matrices, Matrix transformations

Determinants – Cofactor expansion, Row Reduction, Properties of determinants, Cramer's Rule

2. Euclidean vector spaces – Vectors in n-space, Norm, Dot product, Distance in Rn, Orthogonality, Geometry of linear systems, cross product

General vector spaces – Real vector spaces, subspaces, Linear independence, Change of basis, Fundamental matrix spaces, Matrix transformations

3. Eigenvalues and eigenvectors – Diagonalization, Complex vector spaces,

Inner product spaces – Inner products, Gram-Schmidt Process, QR decomposition, Least square approximation

4. Diagonalization and quadratic forms – Orthogonal matrices, Orthogonal diagonalization, Quadratic forms, Optimization
General linear transformations – Compositions and inverse transformations, Isomorphism, Matrices for General linear transformations

5. Numerical methods – LU decompositions, Power method, Singular Value Decomposition

Applications of linear algebra – Polynomial interpolation, Geometry of matrix operations in R2, Differential equations, Dynamical systems, Mathematical modeling using least squares, Function approximation, Search engines

References:

1. Howard Anton, Chris Rorres, Elementary Linear Algebra, 11e, Wiley, 2013
2. Introduction to Linear Algebra, Gilbert Strang, 5e, Cambridge Press, 2016
3. Schaum's Outline of Linear Algebra, Seymour Lipschutz, Marc Lipson, 5e, McGraw-Hill Education, 2012

23-813-0202: Data Structures

Core/Elective: **Core** Semester: **2** Credits: **4**

Course Description

The aim of this course is to expose the students to the common data structures that are used in various computational problems. The course covers the design, analysis, and implementation of various data structures. The topics covered include elementary data structures such as arrays, stacks, queues, and lists as well as advanced data structures such as trees.

Course Outcomes (CO)

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

CO1: Understand different asymptotic notations to analyze performance of algorithms.
CO2: Use elementary and advanced data structures to solve real world problems efficiently.
CO3: Manipulate data using non-linear data structures like trees to design algorithms for various applications.
CO4: Compare and contrast different searching and sorting methods.
CO5: Understand different memory management techniques and their significance.
CO6: Analyse various hashing techniques.

Course Content

1. Introduction to programming methodologies – structured approach, stepwise refinement techniques, programming style, documentation. Elementary data organization - Data structure - Data structure operation - Analysis of algorithms: frequency count, definition of Big O notation, asymptotic analysis of simple algorithms - Recursive and iterative algorithms.

2. Array, Records and Pointers: Introduction, Linear array, Representation of linear array in memory, Traversing linear array, Inserting and Deleting, Sorting methods, Searching methods. String - representation of strings, concatenation, substring searching and deletion.

3. Linked List: Introduction, Linked list, Representation of Linked list in memory, Searching a linked list, Memory allocation, Garbage collection, Insertion and deletion in linked list, doubly linked list, Circular linked list, applications of linked list: polynomials, Memory management, memory allocation and deallocation, First-fit, best-fit and worst-fit allocation schemes.

4. Stacks, Queues, Recursion - Introduction, Stacks, Queues, Operations on stacks and Queues, Implementation of Stacks and Queues using arrays and linked list, Arithmetic expression evaluation, Recursion, DEQUEUE (double ended queue), Multiple Stacks and Queues, Applications.

5. Tree - Introduction, Terminology of Binary tree, Types of Binary tree, Traversing of binary tree, Header Nodes, Threads. Binary search tree – creation, insertion and deletion and search

operations, applications. B-Trees, B+-Trees. Hash Tables – Hashing functions – Mid square, division, folding, digit analysis, collusion resolution and Overflow handling techniques.

References

1. Samanta D.: Classic Data Structures, 2e, Prentice Hall India, 2009.
2. Richard F. Gilberg, Behrouz A. Forouzan: Data Structures: A Pseudocode Approach with C, 2e, Cengage Learning, 2005.
3. Aho A. V., J. E. Hopcroft, J. D. Ullman: Data Structures and Algorithms, Pearson Publication, 1983.
4. Tremblay J. P., P. G. Sorenson: Introduction to Data Structures with Applications, 2e, Tata McGraw Hill, 1995.
5. Peter Brass: Advanced Data Structures, Cambridge University Press, 2008.
6. Lipschutz S.: Theory and Problems of Data Structures, Schaum's Series, 1986.
7. Wirth N.: Algorithms + Data Structures = Programs, Prentice Hall, 2004.
8. Horwitz E., S. Sahni, S. Anderson: Fundamentals of Data Structures in C, University Press (India), 2008.

23-813-0203: Introduction to Artificial Intelligence

Core/Elective: **Core** Semester: **2** Credits: **4**

Course Description

This course introduces what Artificial Intelligence (AI) is, explores use cases and applications of AI, AI concepts including logic, problem solving, knowledge representation and planning. The course will be exposed to various issues and concerns surrounding AI such as ethics and bias, & jobs, and get advice from experts about learning and starting a career in AI.

Course Outcomes (CO)

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

CO1: Appreciate AI concepts.
CO2: Know pipeline of a typical AI project.
CO3: Understand knowledge based systems.
CO4: Know ethical concerns.

Course Content

1. Introduction to AI, Evolution of AI, Turing Machine, Turing test, Category of AI, Applications of AI - Industry examples of AI systems

2. Problem Solving - Solving problems by searching, Uninformed and Informed search strategies, Heuristic functions Search in complex environments, Adversarial search and games, Constraint satisfaction problems

3. Knowledge based agents, First order logic, Propositional logic, Agents based on propositional logic, Knowledge Representation - Ontological Engineering, Categories and Objects, Events, Mental Objects and Modal Logic, Reasoning Systems

4. Planning - Classical Planning, Heuristics for Planning, Hierarchical Planning, Planning and Acting in Nondeterministic Domains, Representing temporal and resource constraints

5. Philosophy, Ethics, and Safety of AI - Limits of AI, The Ethics of AI, AI Safety, Future of AI - AI Components, AI Architectures

References

1. Ethem Alpaydin, Machine Learning: The New AI, MIT Press, 2016
2. Stuart Russell and Peter Norvig, Artificial Intelligence - A Modern Approach, 3e, Pearson Education India, 2015
3. Andriy Burkov, The Hundred-Page Machine Learning Book, Andriy Burkov, 2019
4. Introduction to AI, Coursera

5. AI for everyone, Coursera
6. Jeff Heaton, Artificial Intelligence for Humans, CreateSpace, 2013
7. Mark Coeckelbergh, AI Ethics, MIT Press, 2020

23-813-0204: Operating Systems

Core/Elective: **Core** Semester: **2** Credits: 4

Course Description

The aim of this course is to provide the students general understanding of the structure of modern computer systems. The topics covered under this course includes processes and processor management, concurrency and synchronization, memory management schemes, file system and secondary storage management.

Course Outcomes (CO)

After the completion of the course, the students will be able to:
CO1: Explain the objectives and functions of modern operating systems
CO2: Describe how operating systems have evolved over time from primitive batch systems to sophisticated multi-user systems
CO3: Analyze the tradeoffs inherent in operating system design
CO4: Describe the functions of a contemporary operating system with respect to convenience, efficiency, and the ability to evolve
CO5: Discuss networked, client-server, distributed operating systems and how they differ from single user operating systems
CO6: Identify potential threats to operating systems and the security features design to guard against them
CO7: Describe how issues such as open source software and the increased use of the Internet are influencing operating system design

Course Content

1. Overview of operating systems - functionalities and characteristics of OS, Hardware concepts related to OS, CPU states, I/O channels, memory hierarchy, microprogramming, The concept of a process - operations on processes, process states, concurrent processes, process control block, process context.

2. UNIX process control and management - PCB, signals, forks and pipes, Interrupt processing, operating system organisation, OS kernel FLIH, dispatcher, Job and processor scheduling - scheduling algorithms, process hierarchies.

3. Problems of concurrent processes, critical sections, mutual exclusion, synchronisation, deadlock
   Mutual exclusion - process co-operation, producer and consumer processes, Semaphores- definition, init, wait, signal operations, Use of semaphores to implement mutex, process synchronisation etc., implementation of semaphores, Critical regions, Conditional Critical Regions, Monitors.

4. Interprocess Communication (IPC), Message Passing, Direct and Indirect, Deadlock - prevention, detection, avoidance, banker's algorithm, Memory organisation and

management, storage allocation, Virtual memory concepts, paging and segmentation, address mapping, Virtual storage management, page replacement strategies.

5. File organisation: blocking and buffering, file descriptor, directory structure, File and Directory structures, blocks and fragments, directory tree, inodes, file descriptors, UNIX file structure, Secondary Storage Management - disk components, disk scheduling, swap-space management, Distributed systems, structures, file systems, distributed coordination, Protection and Security, access rights, access matrix, Networks, Routing, Connection strategies, remote file systems.

References:

1. Silberschatz, Galvin and Gagne, Operating Systems Concepts, 10e, Wiley, 2018
2. Andrew Tanenbaum, Modern Operating Systems, 4e, Pearson, 2014
4. William Stallings, Operating Systems, 9e, Pearson, 2017
5. Harvey M. Deitel, Paul J. Deitel, David R. Choffnes, An introduction to operating systems, 3e, Pearson, 2003

23-813-0205: Java Programming

Core/Elective: **Core** Semester: **2** Credits: **4**

Course Description

The aim of this course is to provide the students a basic understanding in Java programming. The topics covered under this course includes the characteristics of Java Applets, Exception handling, Multithreading programming, Streams, Networking etc.

Course Outcomes (CO)

After the completion of the course, the students will be able to:
CO1: Implement OOPs concepts using Java.
CO2: Understand the use of Packages and Interface in java.
CO3: Develop and understand exception handling, multi-threaded applications with synchronization.
CO4: Implement I/O operations using files.
CO5: Implement programs using Java Database Connectivity.
CO6: Understand the use of applets in web-based applications.
CO7: Implement network programming using sockets.

Course Content

1. Java Basics: History of Java, Java features, data types, variables, operators, expressions, control statements, type conversion and casting, Concepts of - classes, objects, constructors, Access Specifiers, Access Modifiers, overloading methods, recursion, nested and inner classes

2. Inheritance, Inheriting data members and methods, Single and Multilevel inheritance, use of super and this keywords. Polymorphism- method overriding, dynamic method dispatch, abstract and final classes. Arrays and Strings - One dimensional arrays, Multidimensional arrays, exploring String class and methods, String Buffer class. Interface: creation and implementation of an Interface. Packages - creating and accessing a package, importing packages, creating user defined packages

3. Exception Handling: benefits of exception handling, exception hierarchy, usage of try, catch, throw, throws and finally, built-in exceptions, creating own exception sub classes. Multi-threaded Programming: thread life cycle, creating threads, thread priorities, synchronizing threads, Inter Thread Communication.

4. Managing input/output files in java, concepts of streams, stream classes, byte stream classes, character stream classes, using streams, I/O classes, file classes, I/O exceptions, creation of files, reading/writing characters, byte handling primitives, data types, random access files, JDBC (Java Database Connectivity), overview, implementation.

5. Java generics- boxing and unboxing, varargs, subtyping, wildcards, reifiable types, reflected

types; Java collections - Collection framework, Collection interfaces, Sets - HashSet, TreeSet, Queues- PriorityQueue, BlockingQueue, Lists- ArrayList, Maps - HashMap, TreeMap, ConcurrentMap, Java lambdas- functional interfaces

References

1. C. Thomas Wu, An introduction to Object-oriented programming with Java, 5e, McGraw-Hill, 2009.
2. Cay S. Horstmann, Core Java: Volume I – Fundamentals, 11e, Pearson, 2020.
3. Herbert Schildt, Java: The Complete Reference, 9e, McGraw-Hill, 2017.
4. K. Arnold, J. Gosling, David Holmes, The JAVA programming language, 4e, Addision-Wesley, 2005.
5. Paul Deitel and Harvey Deitel, Java, How to Program: Early Objects, 11e, Pearson Education, 2018.
6. Timothy Budd, Understanding Object-oriented programming with Java, 2e, Pearson Education, 2001.
7. Y. Daniel Liang, Introduction to Java programming, Comprehensive Version, 10e, Prentice Hall India, 2013.

23-813-0206: Lab 3 - Data Structures Lab

Core/Elective: **Core** Semester: **2** Credits: **1**

Course Description

The aim of this course is to equip the students with the skills to design and implement elementary and advanced data structures. The course will enable the students to identify and apply the suitable data structures for the given real world problem. It also enables them to gain knowledge in practical applications of data structures.

Course Outcomes (CO)

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

CO1: Design and analyze the time and space efficiency of algorithms.
CO2: Apply the appropriate data structure for a given real world problem.
CO3: Learn practical knowledge on the applications of data structures

Course Content

The students need to design and implement the following data structures and algorithms as part of the lab assignments.

- List data structure using arrays and singly linked list.
- Doubly linked list.
- Stack using array and singly linked list.
- Queue using array and singly linked list.
- Circular Queue using linked list.
- Sorting and Searching algorithms.
- Binary tree traversal algorithms.
- Binary Search Tree.
- BFS and DFS algorithms.
- Dijkstra's Algorithm.
- Kruskal's Algorithm.

.

23-813-0207: Lab 4 - Java Programming Lab

Core/Elective: **Core** Semester: **2** Credits: **1**

Course Description

The aim of this course is to provide a basic knowledge of java programming for students and thereby develop software development skills in java programming. As part of this course students will acquire proficiency to develop projects in java programming and help them to solve the real world problems through java programming.

Course Outcomes (CO)

After the completion of the course, the students will be able to:
CO1: Build software development skills using java programming for real world applications.
CO2: Implement front-end and back-end of an application.
CO3: Implement classical problems using java programming.

Course Content

The students shall write Java programs to implement the following:

- Object-oriented programming concepts.
- Exception handling.
- Threads.
- Multi-threading.
- Serialization concept.
- Thread synchronization.
- Producer-Consumer problem.
- File programming.
- Database programming using JDBC.
- Handling mouse & key events.
- Socket Programming.
- Applications of Java Collections.

23-813-0301: Design & Analysis of Algorithm

Core/Elective: **Core** Semester: **3** Credits: **4**

Course Description

The course covers the foundational algorithms in depth. The course helps in understanding the working and complexity of the fundamental algorithms and to develop the ability to design algorithms to attack new problems.

Course Outcomes (CO)

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

CO1: Understand the basic concepts of design and analysis of fundamental algorithms.
CO2: Develop the ability to design algorithms to attack new problems.
CO3: Prove the correctness of algorithms.
CO4: Develop the ability to analyze the complexity of algorithms.
CO5: Understand Complexity classes, concepts of P and NP problems.

Course Content

1. Introduction to design and analysis of algorithms, models of computation, correctness proofs, insertion sort, computational complexity, Master theorem , proof of Master theorem, merge sort, Quick sort, heaps, heap sort, binary search, binary search trees.

2. Graph algorithms, BFS and DFS, Dijkstra's algorithm, proof of correctness of Dijkstra's algorithm, Complexity analysis of Dijkstra's algorithm , Negative weight edges and cycles , Bellman-Ford algorithm, proof of correctness and complexity of Bellman-Ford, All pairs shortest paths, Floyd-Warshall algorithm, proof of correctness and complexity, Minimum Spanning Trees , Prim's algorithm, Cut property, Kruskal's algorithm, proof of correctness and complexity analysis of Kruskal's Algorithm, Maximum-Flow networks, Ford-Fulkerson method, proof of correctness and complexity, Edmonds-Karp algorithm.

3. Probability review, Experiments, outcomes, events, Random variables, Expectation, Linearity of Expectation, Indicator Random Variables, Hiring Problem, Quicksort , Best case and Worst case complexity, Randomized Quicksort , Average case complexity , Hashing, Chaining, Open Addressing, Universal Hashing, Perfect Hashing , Analysis of hashing operations.

4. Dynamic Programming, Rod-cutting problem, Recursive formulation, Bottom-up reformulation of recursive algorithms, Optimal Substructure Property, Matrix chain multiplication, Complexity of dynamic programming algorithms, Sequence Alignment , Longest common subsequence, Greedy algorithms, Optimal substructure and greedy-choice properties , 0-1 and fractional Knapsack problems, Huffman coding.

5. P vs NP, NP Hardness, Reductions, Travelling Salesman Problem, NP-Completeness, SAT, 2- SAT and 3-SAT, Vertex Cover.

References

1. Thomas H. Cormen et al, Introduction to Algorithms, 3e, MIT Press, 2009.
2. Jon Kleinberg, Eva Tardos, Algorithm Design, 2e, Pearson, 2015.
3. Robert Sedgewick, Kevin Wayne, Algorithms, 4e, AW Professional, 2011.
4. Steven S. Skiena, The Algorithm Design Manual, 2e, Springer, 2011.

23-813-0302: Probability and Statistics for Data Science

Core/Elective: **Core** Semester: **3** Credits: **4**

Course Description

The aim of this course is to introduce fundamental concepts in probability and statistics from a data-science perspective. The aim is to become familiarized with probabilistic models and statistical methods that are widely used in data analysis.

Course Outcomes (CO)

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

CO1: Learn the core concepts of probability theory.
CO2: Understand basic principles of statistical inference in estimation and testing.
CO3: Understand the connection between statistical theory and statistical practice.
CO4: Understand the collection, analysis, interpretation, and presentation of data.
CO5: Evaluate problems on discrete and continuous probability distributions.
CO6: Explore certain statistical concepts in practical applications of data science domain.

Course Content

1. Probability theory: probability spaces, conditional probability, independence – Random variables: discrete and continuous random variables, functions of random variables, generating random variables – Multivariate random variables: joint distributions, independence, generating multivariate random variables, rejection sampling – Expectation: Mean, variance and covariance, conditional expectation.

2. Random process: definition, mean and autocovariance functions, iid sequences, Gaussian and Poisson process , random walk – Convergence of random process: types of convergence, law of large numbers, Central limit theorem, monte carlo simulation – Markov chains: recurrence, periodicity, convergence, markov-chain monte carlo- Gibbs sampling, EM algorithm, variational inference.

3. Descriptive statistics: histogram, sample mean and variance, order statistics, sample covariance, sample covariance matrix – Frequentist statistics: sampling, mean square error, consistency, confidence intervals, parametric and non-parametric model estimation.

4. Bayesian statistics: Bayesian parametric models, conjugate prior, bayesian estimators – Hypothesis testing: testing framework, parametric testing, permutation test, multiple testing – Mixture models: Gaussian mixture models, multinomial mixture models.

5. Linear regression: linear models, least-squares estimation, interval estimation in simple linear regression, overfitting – Multiple linear regression models: Estimation of model parameters, MLE – Nonlinear regression: Non linear least squares, transformation to linear model – Generalized linear models: logistic regression models, Poisson regression.

References

1. Michael Mitzenmacher and Eli Upfal; Probability and Computing, 2e, Cambridge University Press, 2017.

2. Alan Agresti, Christine A. Franklin and Bernhard Klingenberg; Statistics: The Art and Science ofLearning from Data, 4e, Pearson, 2017.

3. Sheldon M Ross; A First Course in Probability, 10e, Pearson, 2018.

4. Robert V Hogg, Joseph W McKean and Allen T Cralg; Introduction to Mathematical Statistics, 8e, Pearson, 2018.

5. Douglas C Montgomery, Elizabeth A Peck and G Geoffrey Vining; Introduction to Linear Regression Analysis, 5e, Wiley-Blackwell, 2012.

23-813-0303: Mathematics for Machine Learning

Core/Elective: **Core** Semester: **3** Credits: **4**

Course Description

The aim of this course is to inculcate a comprehensive knowledge about mathematical formalisms required to understand machine learning concepts. The course introduces in detail linear algebra, probability concepts, optimization, and some of the applications.

Course Outcomes (CO)

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

CO1: Outline the fundamental concepts of linear algebra.
CO2: Illustrate matrix diagonalization.
CO3: Analyze the process of backpropagation.
CO4: Apply Bayes' theorem.
CO5: Analyze the gradient descent algorithm
CO6: Examine linear programming problems.
CO7: Build some of the basic machine learning applications.

Course Content

1. Linear Algebra – vectors – matrices – systems of linear equations – vector spaces – linear independence –basis and rank – linear mappings – affine spaces – Norms – lengths and distances – angles and orthogonality– orthonormal basis – inner product of functions - orthogonal projections – rotations.

2. Determinant and trace – eigenvalues and eigenvectors – cholesky decomposition – eigen decomposition and diagonalization – singular value decomposition – matrix approximation – Partial differentiation – gradients – gradients of vectors and matrices – higher order derivatives – backpropagation and automatic differentiation – multivariate Taylor series.

3. Probability review – conditioning and independence – Bayes theorem – counting – discrete and continuous random variables – discrete and continuous probability distributions – Gaussian distribution – Bayesian inference – limit theorems – estimation – conjugacy and exponential family – inverse transform – sampling from distributions.

4. Optimization – gradient descent – choosing the right step size – gradient descent with momentum – stochastic gradient descent – constrained optimization and Lagrange multipliers – convex optimization – linear programming – quadratic programming – Empirical risk minimization – probabilistic modeling and inference – directed graphical models.

5. Applications: linear regression – parameter estimation – Bayesian Linear Regression – PCA –

Maximum Variance Projections – Low-Rank Approximations – Gaussian mixture models – Parameter learning via maximum likelihood – EM Algorithm – Support Vector Machines – Separating Hyperplanes – Primal and Dual forms – The Kernel Trick.

References:

1. Gilbert Strang, Linear Algebra and Learning from Data, Wellesley-Cambridge Press, 2019.

2. Marc Peter Deisenroth et al., Mathematics for Machine Learning, 1e, Cambridge Press,2020.

3. Mehryar Mohri et al., Foundations of Machine Learning, 2e, The MIT Press, 2018.

4. Gilbert Strang, Introduction to Linear Algebra, 5e, Wellesley-Cambridge Press, 2016.

5. James Stewart, Multivariable Calculus, 7th Edition, Cengage Learning, 2011.

6. Dimitri P. Bertsekas, John N. Tsitsiklis, Introduction to Probability, 2e, Athena Scientific, 2008.

7. Morris H. DeGroot, Mark J. Schervish, Probability and Statistics, 4e, Pearson, 2011.

23-813-0304: Database Systems

Core/Elective: **Core** Semester: **3** Credits: **4**

Course Description

This course introduces fundamental principles of Database Management Systems (DBMS) with special focus on relational databases. The course also gives a glimpse of the alternative data management models such as NoSQL and Hadoop. This course helps the learners to manage data efficiently by identifying suitable structures to maintain organization data and to develop applications that utilize database technologies.

Course Outcomes (CO)

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

CO1: Explain how use of database systems evolved from programming with simple collections of data files.
CO2: Describe the major components of a modern database system.
CO3: Design solutions for querying data using languages such as SQL.
CO4: Discuss and compare the aspects of Concurrency Control and Recovery in Database systems
CO5: Explain various types of NoSQL databases
CO6: Give examples of interactions with database systems that are relevant to computer engineering.
CO7: Discuss alternative data management techniques

Course Content

1. Database Management Systems (DBMS) - Characteristics of Database system, Structured, semi-structured and unstructured data. Data Models and Schema - Three Schema architecture. Database Languages, Database architectures and classification. ER model - Basic concepts, entity set & attributes, notations, Relationships and constraints, cardinality, participation, notations, weak entities, relationships of degree 3.

2. Structure of Relational Databases - Integrity Constraints, Synthesizing ER diagram to relational Schema Introduction to Relational Algebra - select, project, cartesian product operations, join - Equi-join, natural join. query examples, Structured Query Language (SQL),Table definitions and operations- SQL queries on single and multiple tables, Nested queries (correlated and non-correlated), Aggregation and grouping, Views, assertions, Triggers, SQL data types

3. Different anomalies in designing a database, The idea of normalization, Functional dependency, Armstrong's Axioms (proofs not required), Closures and their computation,

Equivalence of Functional Dependencies (FD), Minimal Cover (proofs not required). First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF), Boyce Codd Normal Form (BCNF), Lossless join and dependency preserving decomposition, Algorithms for checking Lossless Join (LJ) and Dependency Preserving (DP) properties

4. Transaction Processing Concepts - overview of concurrency control, Transaction Model, Significance of concurrency Control & Recovery, Transaction States, System Log, Desirable Properties of transactions. Serial schedules, Concurrent and Serializable Schedules, Conflict equivalence and conflict serializability, Recoverable and cascadeless schedules, Locking, Two-phase locking and its variations. Log-based recovery, Deferred database modification, check-pointing.

5. Introduction to NOSQL Systems - The CAP Theorem - Document-Based NOSQL Systems and MongoDB - NOSQL Key-Value Stores - Column-Based or Wide Column NOSQL Systems - NOSQL Graph Databases and Neo4j - Big Data - Introduction to MapReduce and Hadoop - Hadoop Distributed File System (HDFS) - MapReduce

References

1. Avi Silberschatz, Henry F. Korth and S Sudarshan, Database System Concepts, 7e, McGraw Hill International Edition, 2019.
2. Elmasri R. and S. Navathe, Database Systems: Models, Languages, Design and Application Programming, Pearson Education, 2017.
3. Sliberschatz A., H. F. Korth and S. Sudarshan, Database System Concepts, 6e, McGraw Hill, 2013.
4. Dan Sullivan, Arsames Qajar, Hands-On NoSQL: A Practical Guide to Design and Implementation with Technical Case Studies, Wiley, 2021
5. Alex Holms, Hadoop in Practice, 2e, Dreamtech Press, 2018

23-813-0305: Theory of Computation

Core/Elective: **Core** Semester: **3** Credits: **4**

Course Description

This is an introductory course on formal languages, automata, computability, and related matters. These topics form a major part of what is known as the theory of computation. The study of the theory of computation has several purposes, most importantly to familiarize students with the foundations and principles of computer science, to teach material that is useful in subsequent courses and to strengthen students' ability to carry out formal and rigorous mathematical arguments.

Course Outcomes (CO)

After the completion of the course, the students will be able to:
CO1: Demonstrate knowledge of basic mathematical models of computation and describe how they relate to formal languages
CO2: Understand the limitations on what computers can do, and learn examples of unsolvable problems
CO3: Learn that certain problems do not admit efficient algorithms, and identify such problem

Course Content

1. Languages, Grammars, Automata and their applications - Finite automata, DFA, Regular languages, NFA, Regular expressions and Regular Grammars

2. Properties of regular languages, Closure, Identifying non regular languages - Context free languages, Context free grammars, parsing and ambiguity, Programming languages - Methods for transforming grammars - Normal forms

3. Push down automata, languages accepted by push down automata - Connection with Context free languages - Properties of context free languages, pumping lemmas

4. Turing machines, definition, Turing machines as a language acceptors, as transducers, combining Turing machines, Turing's Thesis, Other models of Turing machines

5. Limits of algorithmic computation, Problems that cannot be solved - Other models of computation - Computational complexity, Complexity classes

References

1. Peter Linz, An Introduction to Formal Languages and Automata, Jones & Bartlett Learning, 6e, 2016.

2. John E Hopcroft, Rajeev Motwani and Jeffrey D Ullman, Introduction to Automata Theory, Languages, and Computation, 3e, Pearson Education, 2007
3. John C Martin, Introduction to Languages and the Theory of Computation, TMH, 2007
4. Michael Sipser, Introduction To Theory of Computation, Cengage Publishers, 2013.

23-813-0306: Lab 5 - Algorithms Lab

Core/Elective: **Core** Semester: **3** Credits: **1**

Course Description

This course is designed to teach students how to implement algorithms in an efficient way.

Course Outcomes (CO)

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

CO1: Implement and understand well known algorithms
CO2: Know how to debug and make changes

Course Content

- Sort a given set of elements using the Quicksort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

  Compute the transitive closure of a given directed graph using Warshall's algorithm

  Implement 0/1 Knapsack problem using Dynamic Programming

  From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm

  Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm

  Print all the nodes reachable from a given starting node in a digraph using BFS method

  Check whether a given graph is connected or not using DFS method

  Find Minimum Cost Spanning Tree of a given undirected graph using Prim’s algorithm.

  Implement N Queens problem using Backtracking

23-813-0307: Lab 6 - Database Systems Lab

Core/Elective: **Core** Semester: **3** Credits: **1**

Course Description

This course is designed to teach students the basic database systems and its applications.

Course Outcomes (CO)

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

CO1: Write, test and debug simple sql programs
CO2: Implement sql programs with basic operations
CO3: Develop applications based on transactions
CO4: Use NoSQL as backend to develop a simple application
CO5: Overview of Mapreduce algorithm and use of existing implementation

Course Content

- ER Modeling of a web based reservation system
- Write the queries for Data Manipulation and Data Definition Language.
- Write SQL queries to create views, Joins.
- Write a query to understand the concepts for ROLL BACK, COMMIT & CHECK POINTS.
- Operations in MongoDB – Insert, Query, Update, Delete and Projection
-

23-813-0401: Foundations of Data Science

Core/Elective: **Core** Semester: **4** Credits: **4**

Course Description

While traditional areas of computer science remain highly important, increasingly researchers of the future will be involved with using computers to understand and extract usable information from massive data arising in applications, not just how to make computers useful on specific well-defined problems. This course introduces the statistics and computer science concepts required to master data science as a subject.

Course Outcomes (CO)

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

CO1: Learn the mathematical foundations to deal with high dimensional data.
CO2: Understand the concepts like random graphs, random walks, markov chains.
CO3: Understand the basic underpinnings of machine learning algorithms.
CO4: Learn different algorithms to deal with massive data problems.

Course Content

1. High dimensional space: Law of large numbers, geometry of high dimensions, properties of the unit ball, Gaussians in high dimension, random projection and Johnson-Lindenstrauss Lemma, separating Gaussians – Singular Value Decomposition: Power method to compute SVD, singular vectors and Eigenvectors, Applications of SVD.

2. Random Graphs: G(n,p) model, phase transitions, giant component, branching process, cycles and full connectivity – Growth models of Random Graphs: Growth models with and without preferential attachment, small world graphs.

3. Random walks and Markov chains: Stationary distribution, MCMC, Gibbs sampling, areas and volumes, convergence of random walks, random walks in Euclidean space, web as a Markov chain.

4. Learning and VC dimension: Linear Separators, the Perceptron Algorithm, and Margins, Nonlinear Separators, Support Vector Machines, and Kernels, Strong and Weak Learning – Boosting – VapnikChervonenkis dimension: Examples of Set Systems, The Shatter Function, The VC Theorem, Simple Learning.

5. Algorithms for Massive Data Problems: Locality-Sensitive Hashing - shingling of documents, min-hashing. Distance measures, nearest neighbors, frequent itemsets- LSH families for distance measures, Applications of LSH- Challenges when sampling from massive data Frequency Moments of Data Streams, Counting Frequent Elements, Matrix Algorithms Using Sampling, Sketch of a Large Matrix, Sketches of Documents.

References

1. Avrim Blum, John Hopcroft, Ravindran Kannan; Foundations of Data Science, 2018. https://www.cs.cornell.edu/jeh/book.pdf
2. Jure Leskovec, Rajaraman, A., & Ullman, J. D., Mining of Massive Datasets, Cambridge University Press, 2e, 2016.
3. Charu C. Aggarwal, Data Streams: Models and Algorithms, 1e, Springer, 2007.
4. Michael I Jordan et.al , Frontiers in Massive Data analysis, 1e, National Academies Press, 2013.
5. Nathan Marz & James Warren, Big Data: Principles and best practices of scalable realtime data systems, Manning Publications, 2015.

23-813-0402: Numerical Methods

Core/Elective: **Core** Semester: **4** Credits: **4**

Course Description

The aim of this course is to develop the basic understanding of the numerical algorithms to provide solutions to common problems formulated in science and engineering. As part of this course, students will get exposure to the basic concepts and techniques of numerical solution of algebraic equation, system of algebraic equation, numerical solution of differentiation, integration and their applications to Computer Science and Engineering. The course will also help the students to enhance their analytical skills as well as the programming skills through the implementation of various numerical algorithms.

Course Outcomes (CO)

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

CO1: Learn and implement numerical techniques to find the roots of nonlinear equations and solutions of systems of linear equations.
CO2: Understand the difference operators and the use of Interpolation.
CO3: Analyse different numerical Differentiation and Integration techniques.
CO4: Understand and implement numerical solutions of ordinary and partial differential equations.

Course Content

1. Approximation and Errors in computing: Introduction, Significant digits, Inherent error, Rounding error, Truncation error, Absolute and relative error, Error propagation.

2. Roots of Nonlinear Equations and solution of system of Linear Equations: Bisection method, False position Method, Newton-Raphson Method, fixed – point method, Muller's method for complex and multiple roots, convergence of Bisection, Newton- Raphson's and False position methods, Gauss Elimination method by pivoting, Gauss – Jordan method, Gauss – Seidel method, Relaxation method, convergence of iteration methods.

3. Difference Operators and Interpolation: Forward and Backward difference operators and table, Interpolation with equidistant point, Lagrange Interpolation Polynomial, Newton Interpolating Polynomial using divided Difference Table.

4. Numerical Differentiation and Integration: Differentiating continuous functions, differentiating tabulated functions, Higher order derivatives, Richardson's Extrapolation, Newton – cotes integration formula, Trapezoidal rule, Simpson's rule, Boole's rule and Weddle's rule, Romberg's Integration.

5. Numerical Solution of Ordinary and Partial Differential Equations :Taylor series method, Euler and modified Euler method, RungeKutta methods, Milne's method, Adams – Bashforth-

Moulton method, Finite differences approximations of partial derivatives, Solution of Laplace equation(Elliptic)by standard 5 – point formula , solution of one dimensional heat equation(Parabolic)by Bender-Schmidt method, crank – Nicolson method, Solution of one dimensional wave equation(Hyperbolic) by iterative method.

References

1. Steven C. Chapra, Raymond P. Canale: Numerical Methods for Engineers, 7e, McGrawHill, 2015.
2. Kendall E. Atkinson: An Introduction to Numerical Analysis, John Wiley and Sons, 1989.
3. C.F. Gerald, P.O. Wheatley: Applied Numerical Analysis, 7th edition, Addison-Wesley, 2004.
4. S.S. Sastry: Introductory Methods of Numerical Analysis, 5e, PHI, 2012.
5. S. D. Conte, Carl de Boor: Elementary Numerical Analysis: An Algorithmic Approach, SIAM, 2018.
6. Joe D. Hoffman: Numerical Methods for Engineers and Scientists, 2e, CRC Press, 2001.
7. E. Balagurusamy: Numerical Methods, Tata McGraw Hill, 2017.
8. M.K. Jain, S.R.K. Iyengar, R.K. Jain : Numerical Methods for Scientific and Engineering Computation, 6e, New Age International, 2007.
9. B.S. Grewal: Numerical Methods in Engineering & Science, 11e, Khanna Publishers, 2013.
10. T. Veerarajan, T. Ramachandran: Numerical methods with programs in C, Tata McGraw-Hill. 2006.
11. Nita H. Shah: Numerical methods with C++ programming, PHI, 2009.
12. Curtis F. Gerald, Patrick O. Wheatley: Applied Numerical Analysis, 7e, Pearson Education, 2004.

23-813-0403: Digital Signal Processing

Core/Elective: **Core** Semester: **4** Credits: **4**

Course Description

Digital Signal Processing course covers the essential elements of a DSP system from A/D conversion through powerful statistical modeling algorithms. This course includes both theory and practice with an emphasis on how to implement efficient algorithms in C/C++. We begin with a discussion of basic DSP concepts such as sampling and discrete-time signal representations. We then discuss traditional topics such as transforms and filter design.

Course Outcomes (CO)

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

CO1: Understand the principles of discrete-time signal analysis to perform various signal operations
CO2: Know the principles of z-transforms to finite difference equations.
CO3: Apply the principles of Fourier transform analysis to describe the frequency characteristics of discrete-time signals and systems
CO4: Know the principles of signal analysis to filtering
CO5: Use computer programming tools to process and visualize signals

Course Content

1. Defining Signals - Types of Signals, Signal properties, Example of signals, Various signals The property of periodicity, Difference between CT and DT systems, The delta function Properties and signal types, Manipulating signals - time reversal, time shift, time dilation/contraction, Composing Signals, Sampling theorem

2. Linear systems - Convolution, Direct and Graphical form, Properties of convolution, Correlation, Z-Transform, Discrete Fourier Transform - Fourier transform, Properties, DFT basis functions, Applications of DFT, Fast Fourier Transform.

3. Digital Filters: Time domain and frequency domain parameters, Low pass and high pass filters, Band pass and Band reject filters, Filter classification.

4. Moving average filters, frequency response, Relatives of the moving-average filters, recursive implementation, windowed sinc filters, custom filters, deconvolution.

5. Recursive filters, Single Pole recursive filters, Narrow-band filters, Phase response, Chebyshev filters, Chebyshev and Butterworth responses.

References:

1. Steven W. Smith: The Scientist and Engineers' Guide to Digital Signal Processing, 1e, California Technical Pub, 1998.
2. John G. Proakis, Dimitris K Manolakis, Digital Signal Processing: principles, Algorithms and Applications, 4e, Pearson, 2007.
3. Li Tan, Jean Jiang: Digital Signal Processing: Fundamentals and Applications, 2e, Academic Press, 2013.
4. Robert J. Schilling , Sandra L Harris;, Digital Signal Processing using MATLAB, 3e, CL Engineering, 2016.
5. Juan Zhang, Digital Signal Processing: Fundamentals, Techniques and Applications, 1e, Nova Science Pub Inc, 2016.

23-813-0404: Agile Software Engineering

Core/Elective: **Core** Semester: **4** Credits: **4**

Course Description

Software development is a human activity. Agile methods, whether for project management or software development, are the ideal approach for developing software products where change is a risk factor. This course discusses the important milestones in effective software development and project management in the agile way.

Course Outcomes (CO)

After the completion of the course, the students will be able to:
CO1: Understand the agile principle and methodologies and appreciate the need for iterative approaches to software development.
CO2: Develop a software product architecture using UML.
CO3: Communicate with the development team using industry standard notations, designs and documentations.
CO4: Evaluate the purpose and benefits of agile methodologies like Scrum compared to traditional methods.
CO5: Apply various techniques, metrics and strategies for testing software projects.
CO6: Analyze, Formulate, and Apply key agile project management principles to manage a practical project.
CO7: Create an ability to work as a team leader by establishing goals, planning tasks and meeting the goals.

Course Content

1. Agile product architecting using UML: Envisioning the product – product vision – desirable qualities of the vision - customer needs – techniques for creating vision – dependencies and layering.

2. Agile testing and development: Testing in agile, Refactoring development artifacts, agile patterns for user interface development.

3. Agile project management principles. Agile philosophy. APM frameworks – envision, speculate,explore, adapt and close. Configuring project life cycles. Deliverables – management, technical. Feature based delivery Agile technical team: Roles and responsibilities, team empowerment, leadership collaboration.

4. Agile practices: Facilitated workshops, MoSCoW approach to prioritization, iterative development methodologies – SCRUM and XP, modeling, timeboxing.

5. Agile project planning and estimation: Agile requirements - structure and hierarchy of requirements. The Agile approach to estimating- Agile measurements.

References

1. Gary McLean Hall, Adaptive Code: Agile coding with design patterns and SOLID principles 2e, Microsoft Press, 2017.
2. Robert C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship, 1e, PHI, 2017.
3. Marcus Ries and Diana Summers, Agile Project Management: A Complete Beginner's Guide To Agile Project Management, CreateSpace Independent Publishing Platform, 2016.
4. Robert K. Wysocki, Effective Project Management: Traditional, Agile, Extreme, 7e, Wiley India 2014.
5. John C. Goodpasture, Project Management the Agile Way: Making it Work in the Enterprise, 1e, Cengage Learning India, 2014.

23-813-0405: Optimization Techniques

Core/Elective: **Core** Semester: **4** Credits: **4**

Course Description

The aim of this course is to provide exposure to the well-known conventional and advanced optimization techniques developed during the last three decades. This course will emphasize on the advanced optimization techniques to solve large-scale problems especially with nonlinear objective functions.

Course Outcomes (CO)

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

CO1: Understand the basic concepts of optimization and its applications.
CO2: Understand the mathematical representation and classical methods for solving optimization problems.
CO3: Explain and demonstrate working principles of various population based optimization techniques.
CO4: Explain and Demonstrate working principle of various Hybrid Algorithms for optimization.

Course Content

1. Graphical solution of linear programming problems, The Simplex Method: Computational procedure, Artificial variable techniques, Two-phase simplex method. Duality in linear programming: Concept of duality. Formulation of primal dual pairs, Duality and simplex method, Dual simplex method and algorithm, Computational procedure of the revised simplex method.

2.Transportation Problems: Mathematical formulation, Vogel's method with optimality test – MODI method, Unbalanced transportation problem. Assignment problem - Mathematical formulation, Hungarian assignment method, the Travelling Salesman's problem (TSP). Sequencing problems: Problems with n jobs & 2 machines, n jobs and k machines, 2 jobs and k machines.

3. Integer Programming: Gomory's methods, Branch & Bound method. Network Scheduling: Basic terms, Critical path methods, PERT. Queuing Theory: Characteristics of queuing systems, Poisson process and exponential distribution, Steady state M/M/1, M/M/C (Models I, II, IV, V). Inventory Control: Inventory Costs, Economic order quantity, Deterministic inventory problems, EOQ problems with no shortage, With shortage, Production problem with no shortage, with shortage.

4. Genetic Algorithm-Introduction-Working principle-Representation-selection-fitness assignment-reproduction-cross over-mutation-constraint handling-advanced genetic algorithms-Applications-Artificial Immune Algorithm-Introduction-Clonal selection algorithm- Negative selection algorithm-Immune network algorithms-Dendritic cell algorithms.

5. Particle Swarm Optimization - Introduction - Working principles- Parameter selection-Neighborhoods and Topologies-Convergence-Artificial Bee Colony Algorithm-Introduction-Working principles-Applications-Cuckoo search based algorithm-Introduction- Working principles- Random walks and the step size-Modified cuckoo search. Hybrid Algorithms-Concepts- divide and conquer- decrease and conquer - HPABC - HBABC - HDABC - HGABC - Shuffled Frog Leaping Algorithm - Working principles -Parameters- Grenade Explosion Algorithm-Working principle-Applications.

References

1. Kanti Swarup, P. K. Gupta & Man Mohan: Operations Research, Sultan Chand & Sons Publishers, 2010.
2. S. D. Sharma: Operations Research: Theory, Methods & Applications, Kedar Nath Publishers, 2012.
3. Hamdy A. Taha: Operation Research -An Introduction, 9th edition, Pearson Education, 2010.
4. Dan Simon, Evolutionary Optimization Algorithms, 1e, Wiley, 2013.
5. Xin-She Yang, Engineering Optimization: An Introduction with Meta-heuristic Applications, 1e, Wiley, 2010.
6. S.S. Rao, Engineering Optimization: Theory and Practice, 4e,New Age International, 2013.
7.. R. VenkataRao, Teaching Learning Based Optimization Algorithm: And Its Engineering Applications, 1e, Springer, 2016.

23-813-0406: Lab 7 - Numerical Methods Lab

Core/Elective: **Core** Semester: **4** Credits: **1**

Course Description

As part of this course the students will get exposure to the application of the computer to solve problems using the various numerical techniques.

Course Outcomes (CO)

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

CO1: Write, test and debug programs which implement different numerical techniques.
CO2: Solve simple engineering problems using numerical methods.
CO3: Develop applications using different numerical methods.
CO4: Analyse the computational merits and demerits of different numerical techniques.

Course Content

Write programs to implement the following:

- Techniques to find roots of nonlinear equations.
- Methods to find solutions of systems of linear equations.
- Interpolation techniques.
- Numerical differentiation and integration techniques.
- Numerical solutions of ordinary and partial differential equations.

23-813-0407: Lab 8 - Optimization Techniques Lab

Core/Elective: **Core** Semester: **4** Credits: **1**

Course Description

The aim of this course is to provide an exposure to the computer-based implementation of the optimization techniques. This course will also emphasize on how evolutionary algorithms like genetic algorithms and swarm intelligence algorithms can be implemented using computers.

Course Outcomes (CO)

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

CO1: Write, test and debug programs which implement different optimization techniques.
CO2: Solve simple engineering problems using optimization algorithms.
CO3: Develop applications using different evolutionary algorithms.
CO4: Analyse the computational merits and demerits of different optimization techniques.

Course Content

The students shall implement the following algorithms:

The Simplex Method
Dual simplex method
Transportation Problems
Assignment problem
Travelling Salesman's problem
Sequencing problems
Integer Programming
Genetic Algorithm
Swarm Intelligence algorithms

23-813-0501: Regression Analysis

Core/Elective: **Core** Semester: **5** Credits: **4**

Course Description

Regression analysis is the most popularly used statistical technique with application in almost every imaginable field. The focus of this course is on a careful understanding of regression models and associated methods of statistical inference, data analysis, interpretation of results, statistical computation and model building.

Course Outcomes (CO)

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

CO1: Understand how to calculate a simple linear regression model
CO2: Analyse the model with standard error, R-squared, and slope
CO3: Understand how to review and check model assumptions
CO4: Understand how to extend the model to multiple linear regression

Course Content

1. Introduction to regression. Motivating examples, an overview of the objectives of regression analysis – Simple linear regression - Examples of simple linear regression

2. The regression model - Estimation of the regression coefficients and error variance - Inferences for the regression coefficients- Estimating the expected response - Predicting future observations - Inverse prediction and regulation

3. Multiple Linear Regression - Estimation of the regression coefficients and error variance - Inferences for the regression coefficients - Estimating the expected response - Predicting future observations. - The Analysis of Variance approach to regression and general linear F-tests

4. Model selection: Mallow's Cp, AIC, BIC, R-squared, subset selection of independent variables, transformation of dependent and independent variables, multicollinearity, principal component regression, ridge-regression, Lasso.

5. Logistic Regression: Statistical models for binary data; Interpretation of odds and odds ratios; Maximum likelihood estimation in logistic regression; Deviance, Residual analysis for logistic regression - Autocorrelation in time series data - Introduction to nonlinear regression models

References

1. Kutner, Applied Linear Statistical Models, McGraw Hill Education, 5e, 2013
2. Seber G.A.F., Linear Regression Analysis, Wiley, 2e, 2018

3. Annette J, Dobson, Adrian G. Barnett, An Introduction to Generalized Linear Models, Chapman & Hall, 4e, 2018
4. Norman R. Draper, Harry Smit, Applied Regression Analysis, 3e, Wiley India, 2011

23-813-0502: Big Data Analytics

Core/Elective: **Core** Semester: **5** Credits: **4**

Course Description

In the age of big data, data science (the knowledge of deriving meaningful outcomes from data) is an essential skill that should be equipped by software engineers. It can be used to predict useful information on new projects based on completed projects. This course provides a practitioner's approach to some of the key techniques and tools used in Big Data analytics. Knowledge of these methods will help the students to become active contributors to the field of Data Science and Big Data Analytics

Course Outcomes (CO)

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

CO1: Understand predictive modeling techniques for data analytics
CO2: Apply data preprocessing techniques for big data
CO3: Measure the performance of data classification and regression models
CO4: Understand the use of Classification Trees and Rule-Based Models in big data analytics projects

Course Content

1. Predictive Models, Process, Data Pre-processing, Data Transformations, Over-Fitting and Model Tuning, Data Splitting, Resampling Techniques.

2.Measuring Performance in Regression Models, The Variance-Bias Trade-off, Linear Regression for Solubility Data, Penalized Models, Nonlinear Regression Models, Multivariate Adaptive Regression Splines, Support Vector Machines, K-Nearest Neighbors

3.Discriminant Analysis and Other Linear Classification Models, Linear Discriminant Analysis, Partial Least Squares Discriminant Analysis, Nearest Shrunken Centroids, Nonlinear Discriminant Analysis, Flexible Discriminant Analysis

4. Measuring Performance in Classification Models, Class Predictions, Class Probabilities, Evaluating Predicted Classes, Two-Class Problems, Evaluating Class Probabilities, Receiver Operating Characteristic (ROC) Curves

5. Classification Trees and Rule-Based Models, Regression Model Trees, Bagged Trees, Random Forests, Boosting, Remedies for Severe Class Imbalance, Factors That Can Affect Model Performance

References

1. Max Kuhn and Kjell Johnson, Applied Predictive Modeling, 2e, Springer, 2018
2. Ankam Venkat, Big Data Analytics, Packt Publishing Limited, Birmingham, UK, 2016
3. EMC Education Services, Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data, 1e, Wiley, 2015
4. Hadley Wickham, Garrett Grolemund, R for Data Science: Import, Tidy, Transform, Visualize, and Model Data, 1e, Shroff/O'Reilly,, 2017
5. Joel Grus, Data Science from Scratch, Shroff, 1e, O'Reilly Media, 2015
6. James D. Miller, Statistics for Data Science, 1e, Packt Publishing Limited, 2017
7. Thomas Rahlf, Data Visualisation with R: 100 Examples, 1e, Springer, 2017

23-813-0503: Cloud Computing

Core/Elective: **Core**   Semester: **5**   Credits: **4**

Course Description

The aim of this course is to provide students an exposure into the cloud computing domain. Students will get the basic understanding about cloud computing architecture, tools and techniques used for cloud deployment and virtualization techniques.

Course Outcomes (CO)

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

CO1: Understand various basic concepts related to cloud computing technologies.
CO2: Understand architecture and concept of different cloud models: IaaS, PaaS, SaaS.
CO3: Explore cloud technologies, architectures, and standards
CO4: Familiarize/Use different cloud services.
CO5: Familiarize with design methodologies and programming for cloud applications.
CO6: Understand various cloud simulators
CO7: Familiarize with the use of VM virtualization, Docking, Orchestration

Course Content

1. Cloud Computing Overview: Origins of Cloud computing – Cloud components - Essential characteristics – On-demand self service, Broad network access, Location independent resource pooling ,Rapid elasticity , Measured service, Comparing cloud providers with traditional IT service providers, Roots of cloud computing.

2. Cloud Insights: Architectural influences – High-performance computing, Utility and Enterprise grid computing, Cloud scenarios – Benefits: scalability ,simplicity ,vendors ,security, Limitations – Sensitive information - Application development- security level of third party - security benefits, Regularity issues: Government policies.

3. Cloud Architecture- Layers and Models: Layers in cloud architecture, Software as a Service (SaaS), features of SaaS and benefits, Platform as a Service (PaaS), features of PaaS and benefits, Infrastructure as a Service (IaaS), features of IaaS and benefits, Service providers, challenges and risks in cloud adoption. Cloud deployment model: Public clouds – Private clouds – Community clouds - Hybrid clouds - Advantages of Cloud computing.

4. Virtual machine based distributed computing - bare-metal hypervisor, virtual machines, creating virtual machines, virtual machine on local host, cloning virtual machines - elastic cloud computing clustering - cold and hot migration - Case studies - KVM, Xen

5. Docker and Containers - Basics of Docker, Working with docker data, Docker networking, Kubernetes clustering- components and objects, load balancing, Interacting with APIs, Orchestration, AI platforms for machine learning serving

References

1. Toby Velte, Anthony Velte, Robert Elsenpeter: Cloud Computing, A Practical Approach, 1e, McGraw-Hill Education, 2009.

2. Rajkumar Buyya, James Broberg, Andrzej Goscinski: Cloud Computing: Principles and Paradigms, 1e, Wiley, 2013.

3. Thomas Erl: Cloud Computing: Concepts, Technology & Architecture, 1e, Pearson Education India, 2014.

4. Michelle Vine: Networking, Models and Methods of Cloud Computing, 1e, Willford Press, 2016.

5. Kai Hwang, Geoffrey C. Fox, Jack J. Dongarra: Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, 1e, Elsevier, 2012.

6. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi: Mastering Cloud Computing, 1e, McGraw Hill Education, 2017.

7. Brendan Burns, Joe Beda, Kelsey Hightower, Kubernetes: Up and Running: Dive into the Future of Infrastructure, 2e, O'Reilly Media, 2019

8. Courses offered through https://www.qwiklabs.com/

23-813-0504: R for Data Science

Core/Elective: **Core** Semester: **5** Credits: **4**

Course Description

The aim of this course is to provide knowledge to students in R programming and its use for effective data analysis. Students will learn how to install and configure the software for a statistical programming environment and discuss generic programming language concepts as they are implemented in a high-level statistical language. This course covers topics such as programming in R, accessing R packages, writing functions, debugging and organizing R code.

Course Outcomes (CO)

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

CO1: Understand the basics in R programming in terms of constructs, control statements, string functions.
CO2: Understand the use of R for data analytics.
CO3: Learn to apply R programming for Text processing.
CO4: Perform appropriate statistical tests using R.
CO5: Create and edit visualizations with R.
CO5: Able to appreciate and apply the R programming from a statistical perspective.

Course Content

1. R Programming Basics: Overview of R programming, Environment setup with R Studio, R Commands, Variables and Data Types, Control Structures, Array, Matrix, Vectors, Factors, Functions, R packages.

2. Data Visualization using R: Reading and getting data into R (External Data): Using CSV files, XML files, Web Data, JSON files, Databases, Excel files. Working with R Charts and Graphs: Histograms, Boxplots, Bar Charts, Line Graphs, Scatterplots, Pie Charts.

3. Statistics with R: Random Forest, Decision Tree, Normal and Binomial distributions,Time Series Analysis, Linear and Multiple Regression, Logistic Regression, Survival analysis, Analysis of Variance (One way ANOVA, Two way ANOVA), Time Series Analysis.

4. OOP: S3 Classes – S4 Classes – Managing your objects – Input/Output – accessing keyboard and monitor – reading and writing files – accessing the internet – String Manipulation – Graphics –Creating Graphs – Customizing Graphs – Saving graphs to files – Creating three-dimensional plots.

5. Advanced R Programming: Interfacing R to Other Languages, Text mining, Neural Networks, Monte Carlo methods, Markov chains, classification, Market Basket Analysis.

References

1. W. N. Venables, D.M. Smith and the R Development Core Team, An Introduction to R, Notes on R: A Programming Environment for Data Analysis and Graphics.
URL: https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf

2. Norman Matloff, The Art of R Programming – A Tour of Statistical Software Design, 1e, No Starch Press, 2011.

3. Jared P. Lander, R for Everyone: Advanced Analytics and Graphics, 1e, Pearson Education India, 2014.

4. Mark Gardener, Beginning R - The Statistical Programming Language, John Wiley & Sons, Inc., 2013.

5. Peter Dalgaard, Introductory Statistics with R, 2e, Springer, 2008.

23-813-0505: Number Theory and Cryptography

Core/Elective: **Core** Semester: **5** Credits: **4**

Course Description

The course provides an introduction to basic number theory, where the focus is on computational aspects with applications in cryptography. Applications to cryptography are explored including symmetric and public-key cryptosystems. Modern cryptographic methods are also discussed.

Course Outcomes (CO)

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

CO1: Understand the number theoretic foundations of modern cryptography.
CO2: Implement and analyze cryptographic and number theoretic algorithms.
CO3: Understand public key cryptosystems.
CO4: Understand modern cryptographic techniques.

Course Content

1. Divisibility, Division Algorithm, Euclidean Algorithm, Congruences, Complete Residue systems, Reduced Residue systems, Fermat’s little theorem, Euler’s Generalization, Wilson's Theorem, Chinese Remainder Theorem, Euler Phi-function, multiplicative property, Finite Fields, Primitive Roots, Quadratic Residues, Legendre Symbol, Jacobi Symbol, Gauss's lemma, Quadratic Reciprocity Law

2. Primality Tests, Pseudoprimes, Carmichael Numbers, Fermat’s pseudoprimes, Euler pseudoprimes, Factorization by Pollard’s Rho method, Simple Continued Fraction, simple infinite continued fractions, Approximation to irrational numbers using continued fractions, Continued Fraction method for factorization.

3.Traditional Cryptosystem, limitations, Public Key Cryptography Diffie-Hellman key exchange, Discrete Logarithm problem, One-way functions, Trapdoor functions, RSA cryptosystem, Digital signature schemes, Digital signature standards, RSA signature schemes, Knapsack problem, ElGamal Public Key Cryptosystem, Attacks on RSA Cryptosystem: Common modulus attack, Homomorphism attack, timing attack, Forging of digital signatures, Strong primes, Safe primes, Gordon's algorithm for generating strong primes.

4. Cubic Curves, Singular points, Discriminant, Introduction to Elliptic Curves, Geometry of elliptic curves over reals, Weierstrass normal form, point at infinity, Addition of two points, Bezout's theorem, associativity, Group structure, Points of finite order.

5. Elliptic Curves over finite fields, Discrete Log problem for Elliptic curves, Elliptic Curve Cryptography, Factorization using Elliptic Curve, Lenstra's algorithm, ElGamal Public Key Cryptosystem for elliptic curves.

References

1. James S. Kraft and Lawrence C. Washington, An Introduction to Number Theory with Cryptography, 2e, CRC Press, 2018.

2. Jeffrey Hoffstein, Jill Pipher and Joseph H. Silverman, An Introduction to Mathematical Cryptography, 2e, Springer, 2014.

3. Christof Paar and Jan Pelzl, Understanding Cryptography, 1e, Springer, 2010.

4. G.H.Hardy and E M Wright, An Introduction to the Theory of Numbers, 6e, Oxford University Press, 2008.

5. Song Y.Yan, Computational Number Theory & Modern Cryptography, 1e, Wiley, 2013.

23-813-0506: Lab 9 – R for Data Science Lab

Core/Elective: **Core**　　　　Semester: **5**　　　　Credits: **1**

Course Description

The aim of this course is to provide knowledge to install and use R for various programming tasks, using looping constructs and R mathematical functions. The course will also give exposure to extended R libraries and packages that can be used for data exploration in R.

Course Outcomes (CO)

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

CO1: Master the use of the R interactive environment.
CO2: Expand R by installing R packages.
CO3: Develop loop constructs in R.
CO4: Use R for descriptive statistics.
CO5: Use R for inferential statistics.

Course Content

Set of programs to be done in the lab:

- Programs on data types in R.
- Built-in functions in R.
- Creating and manipulating vectors.
- Creating and manipulating matrices.
- Operations on Data Frames in R.
- Operations on Lists in R.
- Operators in R.
- Programs on looping constructs.
- Customizing and Saving Graphs in R.
- Plot functions in R.
- 3D plot functions in R.
- Data Visualization.
- Probability Distributions.
- Densities of Random Variables.
- Binomial Distribution.
- Correlation.
- Estimating a linear relationship.
- Perform tests of hypotheses.

23-813-0507: Lab 10 – Data Analytics Lab

Core/Elective: **Core** Semester: **5** Credits: **1**

Course Description

The aim of this course is to provide knowledge to install and use Data Analytics tools ecosystem including Apache Hadoop, Pig, Hive and Spark for various data analytics tasks

Course Outcomes (CO)

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

CO1: Understand the importance of Data Analytics to create competitive advantage in business
CO2: Understand the architectural concepts of Hadoop and introducing map reduce paradigm
CO3: Develop programs using Map-Reduce framework
CO4: Derive business benefit from unstructured data
CO5: Know programming tools PIG & HIVE in Hadoop ecosystem
CO6: Develop Big Data applications for streaming data using Apache Spark
CO7: Build a complete business data analytic solution

Course Content

Run a basic Word Count Map Reduce program to understand Map Reduce Paradigm

Write a Map Reduce program that mines datasets like weather data, purchases data etc.

Write Pig Latin scripts to sort, group, join, project, and filter data

Use Hive to create, alter, and drop databases, tables, views, functions, and indexes

Run apache spark applications using Scala

Data analytics using Apache Spark on standard datasets

References

1. Tom White, Hadoop: The Definitive Guide: Storage and Analysis at Internet Scale, 4e, O'reilly, 2015
2. Alex Holmes, Hadoop in Practice, 2e, Dreamtech Press, 2015
3. Holden Karau, Andy Konwinski,Patrick Wendell and Matei Zaharia Learning Spark: Lightning-Fast Big Data Analysis, 1e, O'Reilly Media, 2015

23-813-0601: Inferential Statistics

Core/Elective: **Core** Semester: **6** Credits: **4**

Course Description

This course covers commonly used statistical inference methods for numerical and categorical data. You will learn how to set up and perform hypothesis tests, interpret p-values, and report the results of your analysis in a way that is interpretable for clients or the public. Using numerous data examples, you will learn to report estimates of quantities in a way that expresses the uncertainty of the quantity of interest. The course also introduces practical tools for performing data analysis and explores the fundamental concepts necessary to interpret and report results for both categorical and numerical data

.

Course Outcomes (CO)

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

CO1: Apply probabilistic and statistical reasoning to describe and analyse essential features of data sets and problems in real-life business situations.
CO2: Understand statistical techniques to estimate the population mean, proportion and variance.
CO3: Acquire techniques to test hypotheses with an assumption on the population means, proportions and variances under different circumstances.
CO4: Carry out nonparametric tests for statistical problems.
CO5:Use and extend knowledge of inferential statistics and their applications in real-life business situations.

1. Introduction to statistical inference - inference methods, confidence, intervals and hypothesis tests - statistical versus practical significance - effect of sample size, confidence, and significance levels - Central Limit Theorem and Confidence Interval - Sampling Variability and CLT, CLT (for the mean) examples, Confidence Interval (for a mean), Accuracy vs. Precision, Required Sample Size for ME, CI (for the mean) examples
2. Inference and Significance - Hypothesis Testing (for a mean), HT (for the mean) examples, Inference for Other Estimators, Decision Errors, Significance vs. Confidence Level, Statistical vs. Practical Significance, Inference for Comparing Means - t-distribution, Inference for comparing two independent means and paired samples, Power, Comparing more than two means, ANOVA, Conditions for ANOVA, Multiple comparisons, Bootstrapping
3. Inference for Proportions - Sampling Variability and CLT for Proportions, Confidence Interval for a Proportion, Hypothesis Test for a Proportion, Estimating the Difference Between Two Proportions, Hypothesis Test for Comparing Two Proportions, Small Sample Proportions, Examples
4. Non-Parametric test - Comparing Two Small Sample Proportions, Chi-Square GOF Test, The Chi-Square Independence Test, Kolmogrov–Smirnov Test, Sign Test, Two-Sample Problem

5. Statistical Decision Theory - Complete and Minimal Complete Class of Decision Rules, Optimal Decision Rule, Method of Finding a Bayes Rule, Methods for Finding Minimax Rule, Invariance

References

1. Pradip Kumar Sahu, Santi Ranjan Pal, Estimation and Inferential Statistics, 1e, Springer, 2015
2. Geoff Cumming, Robert Calin-Jageman, Introduction to the New Statistics: Estimation, Open Science, and Beyond, 1e, Routledge, 2016
3. Malcolm Asadoorian, Essentials of Inferential Statistics, 5e, University Press Of America, 2008
4. Schaum's Outline of Elements of Statistics II: Inferential Statistics, 1e, McGraw-Hill Education, 1999

23-813-0602: Machine Learning Algorithms

Core/Elective: **Core** Semester: **6** Credits: **4**

Course Description

Machine learning is programming computers to optimize a performance criterion using example data or past experience. This course is to discuss many methods that have their bases in different fields: statistics, pattern recognition, neural networks, artificial intelligence, signal processing, control, and data mining. Major focus of the course is on the algorithms of machine learning to help students to get a handle on the ideas, and to master the relevant mathematics and statistics as well as the necessary programming and experimentation.

Course Outcomes (CO)

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

CO1: Demonstrate strength and weakness of Machine Learning approaches.
CO2: Appreciate the underlying mathematical relationships within and across algorithms and different paradigms of Machine Learning.
CO3: Utilize dimensionality reduction techniques for feature selection.
CO4: Examine methods for model building and fine tuning.
CO5: Experiment with Machine learning tools.
CO6: Apply Machine Learning algorithms to many real world problems.

Course Content

1. Machine Learning – Examples of Machine Learning applications – Supervised Learning: Learning a class from examples – Learning multiple classes – Regression – Model selection – Bayesian Decision Theory: Classification – Discriminant functions – Association rules – Parametric methods: MLE – Bayes estimator – Parametric classification – Tuning model complexity.

2. . Multivariate Methods – Classification – Regression – Dimensionality reduction: LDA – PCA –Factor Analysis – ICA – Locally Linear Embedding – MDS- Probabilistic Learning: Gaussian Mixture Models- EM algorithm- Nearest Neighbor Methods – Distance Measures.

3. Support Vector Machines: Optimal separation – Kernels – SVM algorithm – Extensions to SVM – Optimization and Search: Least-squares optimization – conjugate gradients – Search: Search techniques – Exploitation and exploration – Simulated annealing.

4. .Learning with trees: Decision trees – CART – Ensemble Learning: Boosting – Bagging – Random Forests – Unsupervised Learning: K-Means algorithm – Vector quantization – SOM algorithm – Markov Chain Monte Carlo Methods.

5. Graphical Models: Bayesian Networks – Markov Random Fields – HMMS – Tracking Methods – Deep Belief Networks: Hopfield Network – Boltzmann Machine – RBM – Deep Learning.

References

1. Ethem Alpaydin, Introduction to Machine Learning, 3e, MIT Press, 2014.
2. Tom M. Mitchell, Machine Learning, McGraw Hill Education; 1e, 2017.
3. Stephen Marsland, Machine Learning, An Algorithmic Perspective, 2e, CRC Press, 2015.
4. Giuseppe Bonaccorso, Machine Learning Algorithms, 1e, Packt Publishing Limited, 2017.
5. Ethem Alpaydin, Machine Learning- The New AI, MIT Press, 1e, 2016.
6. Andrew Ng, Machine Learning Yearning, ATG AI (Draft version), 1e, 2018.
7. Rohit Singh, Tomi Jaakkola, and Ali Mohammad.6.867 Machine Learning. Fall 2006. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu
8. Andrew Ng, https://www.coursera.org/learn/machine-learning

23-813-0603: Feature Engineering

Core/Elective: **Core** Semester: **6** Credits: **4**

Course Description

This course presents Feature Engineering (FE) ideas and approaches that are as much domain independent as FE can possibly be. This course also emphasizes the different techniques as used in key domains (graph data, time series, text processing, computer vision and others) through case studies.

Course Outcomes (CO)

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

CO1: Demonstrate strength and weakness of different features used for Machine Learning.
CO2: Appreciate the significance of feature normalization, discretization and outliers.
CO3: Utilize dimensionality reduction techniques for feature selection.
CO4: Examine methods for automated feature engineering.
CO5: Experiment with different types of data.

Course Content

1. Feature Engineering - Evaluation metrics, cross validation, out-of-fold estimation, overfitting, curse of dimensionality, Machine learning life cycle - Feature engineering cycle, Analysis - Exploratory Data Analysis, Error Analysis, Domain modeling - Feature construction, Feature types

2. Normalization, Discretization and Outliers- Normalizing features, Discretization and Binning, Descriptive Features, Dealing with Outliers, Advanced Techniques, Computable Features - Imputation, Decomposing Complex Features, Kernel-Induced Feature Expansion

3. Feature Selection, Dimensionality, Reduction and Embeddings - Feature Selection, Regularization, Embedded Feature Selection, Dimensionality Reduction

4. Variable-Length Data and Automated Feature Engineering - Variable-Length Feature Vectors, Instance-Based Engineering, Deep Learning and Feature Engineering, Automated Feature Engineering

5. Case studies - Graph Data, Timestamped data, Textual data, Image data, Audio data

References

1. Pablo Duboue, The Art of Feature Engineering: Essentials for Machine Learning, 1e, Cambridge University Press, 2020

2. Max Kuhn, Kjell Johnson, Feature Engineering and Selection: A Practical Approach for Predictive Models, 1e, CRC Press, 2020
3. Alice Zheng, Amanda Casari, Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists,1e, O'Reilly Media, 2018
4. Guozhu Dong, Huan Liu, Feature Engineering for Machine Learning and Data Analytics, 1e, CRC Press, 2020

23-813-0604: Soft Computing Techniques

Core/Elective: **Core** Semester: **6** Credits: **4**

Course Description

The aim of this course is to cover fundamental concepts used in Soft computing. As part of this course the students will get exposure to Fuzzy logic, Artificial Neural Networks and optimization techniques using Genetic Algorithm. To provide hands-on practices to the students applications of Soft Computing techniques to solve a number of real life problems will be covered. This course will provide exposure to theory as well as practical systems and software used in soft computing.

Course Outcomes (CO)

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

CO1: Learn Fuzzy logic and its applications.
CO2: Understand the basic concepts of Artificial neural networks and its applications.
CO3: Solve single-objective optimization problems using GAs.
CO4: Solve multi-objective optimization problems using Evolutionary algorithms.
CO5: Apply Soft computing techniques to solve problems in various application domains.

Course Content

1. Introduction to Soft Computing: Concept of computing systems - "Soft" computing versus "Hard" computing - Characteristics of Soft computing - Some applications of Soft computing techniques.

2. Fuzzy logic: Introduction to Fuzzy logic - Fuzzy sets and membership functions - Operations on Fuzzy sets - Fuzzy relations, rules, propositions, implications and inferences - Defuzzification techniques - Fuzzy logic controller design - Some applications of Fuzzy logic.

3. Genetic Algorithms: Concept of "Genetics" and "Evolution" and its application to probabilistic search techniques - Basic GA framework and different GA architectures - GA operators: Encoding, Crossover, Selection, Mutation, etc. - Solving single-objective optimization problems using Gas.

4. Multi-objective Optimization Problem Solving: Concept of multi-objective optimization problems (MOOPs) and issues of solving them - Multi-Objective Evolutionary Algorithm (MOEA) - Non-Pareto approaches to solve MOOPs - Pareto-based approaches to solve MOOPs - Some applications with MOEAs.

5. Artificial Neural Networks: Biological neurons and its working - Simulation of biological neurons to problem solving - Different ANNs architectures - Training techniques for ANNs - Applications of ANNs to solve some real life problems.

References

1. Timothy J. Ross, Fuzzy Logic with Engineering Applications, 4e, Wiley, 2016.
2. S. Rajasekaran, and G. A. Vijayalakshmi Pai, Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis and Applications, 1e, Prentice Hall India, 2003.
3. Melanie Mitchell, An Introduction to Genetic Algorithms, 1e, MIT Press, 1998.
4. Nikola K. Kasabov, Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering, 1e, MIT Press, 1996.
5. S. N. Sivanandam and S. N. Deepa, Principles of Soft Computing, 3e, Wiley, 2018.
6. Randy L. Haupt and Sue Ellen Haupt: Practical Genetic Algorithms, 2e, Wiley, 2004.
7. Simon Haykin: Neural Networks and Learning Machines, 3e, Pearson, 2009.
8. J.-S. R. Jang, C.-T. Sun, and E. Mizutani: Neuro Fuzzy and Soft Computing, 1e, Pearson Education India, 2015.

23-813-0605: Parallel Computing

Core/Elective: **Core** Semester: **6** Credits: **4**

Course Description

This course introduces parallel computing with a strong emphasis on programming. The aim of this course is to expose the students to parallel and distributed computing with special focus on writing parallel code for processor intensive applications to be run on GPU and Cluster computing systems.

Course Outcomes (CO)

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

CO1: Understand the characteristics of parallel systems and distributed infrastructures.
CO2: Learn how to design parallel programs and how to evaluate their performance.
CO3: Enable the students to write parallel code for high performance computing applications.
CO4: Expose the students to parallel computing libraries such as OpenMP, MPI, PVM and CUDA.

Course Content:

1. Overview of Parallel computing - von Neumann architecture, Flynn's classification, limits and costs of parallel programming, Parallel Computer Memory Architectures - Shared Memory, Distributed Memory, Hybrid architecture.

2. Parallel Programming Models - Shared Memory Model, Threads model, Message Passing Model, Data Parallel Model, SPMD and MPMD, Designing parallel programs - Automatic vs manual parallelization, Partitioning, Communications, Synchronization, Data Dependencies, Load Balancing, Granularity, I/O, Debugging, Performance analysis and Tuning.

3. Basic Parallel Algorithmic Techniques - Pointer Jumping, Divide-and-Conquer, Partitioning, Pipelining, Accelerated Cascading, Symmetry Breaking, Synchronization (Locked, Lock-free), Parallel Algorithms - Data organization for shared/distributed memory, Min/Max,Sum, Searching, Merging, Sorting, Prefix operations, N-body problems, Matrix operations.

4. Overview of Cluster based distributed computing: Hardware technologies for cluster computing, Software and software architectures for cluster computing: Shared memory (OpenMP) and Message-Passing (MPI/PVM) models. Dynamic process creation, one-sided communication, Parallel I/O, Multi-Core CPU programming.

5. Overview of GPUs: architecture, features and Programming model. System issues: cache and data management, languages and compilers, stream processing, GPU-CPU load balancing. Writing Parallel Programs, GPU-Compute Architecture, CUDA, Memory organization in CUDA.

References

1. Joseph Jaja, Introduction to Parallel Algorithms, 1e, Addison-Wesley Professional, 1992.
2. Ananth Grama, George Karypis, Vipin Kumar, Anshul Gupta, Introduction to Parallel Computing", 2e, Addison-Wesley Professional, 2003.
3. Michael Quinn, Parallel Programming in C with MPI and OpenMP, 1e, McGraw-Hill, 2003.
4. Jason Sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, 1e, Addison-Wesley Professional, 2010.
5.David Culler, J.P. Singh, Anoop Gupta, Parallel Computer Architecture: A Hardware / Software Approach", 1e, Morgan Kaufmann, 1998.
6. William Gropp, Steven Huss-Lederman, Andrew Lumsdaine, Ewing L. Lusk, Bill Nitzberg, William Saphir, Marc Snir, MPI - The Complete Reference. Volume 2, The MPI Extensions, 2e, MIT Press, 1998.
7. David B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors: A Hands-on Approach", 1e, Morgan Kaufmann, 2010.
8. Rob Farber, "CUDA Application Design and Development", 1e, Morgan Kaufmann, 2011.

23-813-0606: Lab 11: Machine Learning and Parallel Computing Lab

Core/Elective: **Core** Semester: **6** Credits: **1**

Course Description

The aim of this course is to expose the students to the practical implementation of algorithms for machine learning from a variety of perspectives. This course will enable students to make use of Data sets in implementing the machine learning algorithms and implement the machine learning concepts and algorithms in any suitable language of choice. The aim of this course is also to expose the students to parallel computing using CUDA programming environment. This course will enable students to implement machine learning algorithms that utilize parallel GPUs in an efficient way.

Course Outcomes (CO)

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

CO1: Understand the implementation procedures for the machine learning algorithms.
CO2: Understand the implementation procedures for the CUDA environment.
CO3: Design Python programs for various Learning algorithms using GPU.
CO4: Apply appropriate data sets to the Machine Learning algorithms.
CO5: Identify and apply Machine Learning algorithms to solve real world problems.

Course Content

Following algorithms need to be implemented in Python using the available public datasets with GPU environment:

- Supervised Learning
- Unsupervised Learning
- Regression
- Bayesian theory
- Dimensionality Reduction Techniques
- Expectation-Maximization Algorithms
- Support Vector Machine
- Decision Trees
- Random Forest
- K-means clustering
- Vector quantization
- Self-organizing Map
- Deep Learning

23-813-0607: Project

Core/Elective: **Core** Semester: **6** Credits: **4**

Course Description

Through the project work, the student has to exhibit the knowledge in terms of engineering or technological innovation or research ability to solve the contemporary problem. On completion of the work, the student shall submit a project report. The qualitative and quantitative results of the work will be evaluated through a viva voce exam.

Course Outcomes (CO)

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

CO1: Demonstrates in depth knowledge and thoughtful application through the detailed analysis of the problem chosen for the study.
CO2: Assess the gap by acquiring knowledge about the previous works, and its interpretation and application.
CO3: Demonstrates the design of the proposed methodology and its merits.
CO4: Organize the interim project content with proper structure and sequencing.
CO5: Demonstrate the academic discussion skills to emphasize, argue with clarity of purpose using evidence for the claims.

23-813-0701: Computational Linguistics

Core/Elective: **Core** Semester: **7** Credits: **4**

Course Description

Computational Linguistics deals with statistical and rule based modelling of natural languages from a computational point of view. This course is intended to give a comprehensive coverage of language processing fundamentals like morphology, Syntax, Semantics and pragmatics. Application of various computational models in application domains like Machine translation, information retrieval etc. is also dealt with.

Course Outcomes (CO)

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

CO1 : Understand the fundamentals of written language processing.
CO2: Applying the fundamentals in real world problems like POS tagging, Corpus development, WordNet, Dialogue processing, document retrieval, Machine translation etc.
CO3: Creating resources for less resource languages.
CO4: Case study of various typical Language processing tools.

Course Content

1. Words- Regular Expressions and Finite Automata-Morphology and Finite State Transducers - Probabilistic Models of Pronunciation and Spelling -N grams.

2. Word Classes and Part-of-Speech Tagging-MM Taggers- probabilistic Context Free Grammars for English Syntax-Parsing with Context Free Grammars- probabilistic parsing-Features and Unification-Language and Complexity.

3. Semantics-Representing Meaning-canonical forms-FOPC-ambiguity resolution-scoping phenomena-Semantic Analysis-syntax driven semantic analysis-Lexical Semantics-Word Sense Disambiguation and Information Retrieval.

4. Discourse-Reference Resolution -Text Coherence -Dialog and Conversational Agents - Dialogue acts-dialogue structure.

5. Statistical alignment and machine translation-clustering- text categorization.

References

1. James Pustejovsky, Amber Stubbs, Natural language annotation for machine learning, 1e, O'Reilly, 2013.
2. Alexander Clark and Chris Fox, The handbook of Computational linguistics and natural language processing, 1e, Willey-Blackwell, 2012.

3. Grant S Ingersoll, Thomas Morton, Andrew L Farris, Taming Text: How to Find, Organize, and Manipulate It, 1e, Manning Publications 2013.
4. Daniel Jurafsky and James Martin , Speech and Language Processing, 2e, Pearson, 2013.
5. Christopher D. Manning and HinRich Schütze, Foundations of statistical natural language processing, 1e, MIT press, 1999.

23-813-0702: Digital Image and Video Processing

Core/Elective: **Core** Semester: **7** Credits: **4**

Course Description

The aim of this course is to inculcate a comprehensive knowledge about various Digital Image and Video Processing techniques. The objectives are to give an in-depth knowledge about the basic theory and algorithms related to Digital Image and Video Processing, provide awareness about the current technologies and issues, provide hands-on experience in using computers to process digital images and Videos using Python and OpenCV library.

Course Outcomes (CO)

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

CO1: Understand the fundamental concepts of signal and image processing systems.
CO2: Evaluate the different spatial filters for image enhancement and restoration.
CO3: Analyze images in the frequency domain using various transforms.
CO4: Apply the different spatial and frequency domain filters on color images.
CO5: Evaluate the performance of different image segmentation algorithms.
CO6: Understand the fundamental concepts of video processing systems.
CO7: Evaluate the different algorithms for motion estimation from videos.
CO8: Understand the techniques for depth estimation from stereo images or videos.
CO9: Develop any image or video processing application.

Course Content

1. Signals: Impulse Sequence - Exponential Sequence - Periodic Sequence. Linear Systems -Shift Invariant systems - Linear Shift Invariant (LSI) systems – Convolution - Correlation. Image Transforms: Fourier Transform - Discrete Fourier Transform - Z- transform – KL Transform. Causal Systems - Random Signals - Stationary Process - Markov Process.

2. Intensity Transformation and Spatial Filtering: Intensity Transformation Functions. Histogram Processing: Histogram Equalization - Histogram Matching. Image enhancement: Arithmetic/Logic operations - Image Subtraction - Image Averaging. Spatial Filtering: Smoothening Spatial Filters - Sharpening Spatial Filters - Laplacian Filter - Unsharp masking - High Boost Filter. Gradient operators: Edge detection filters. Frequency Domain Smoothening - Frequency Domain Sharpening Filters - Laplacian in Frequency domain - Homomorphic Filtering.

3. Image degradation/Restoration process model - Noise probability density functions – Spatial Filtering: Mean Filters - Order-statistics filter - Adaptive Filters - Periodic Noise Reduction – Band-reject filters - Band-pass filters - Notch filters. Inverse filtering - Wiener filtering – Performance measures. Color image processing: Color fundamentals - Color models – RGB,

CMYK – HSI - Color image smoothening and sharpening – Color image histogram - Color edge detection.
4. Point and line detection - Hough Transform. Image Segmentation: Fundamentals – Thresholding – Otsu's optimum global thresholding - Region-based segmentation: Region growing – Region Splitting and Merging - Segmentation using Morphological Watersheds.

5. Color video processing: Video display - Composite versus component video - Progressive and interlaced scan. Motion estimation: Optical flow - pixel based motion estimation - block matching algorithm - deformable block matching algorithm - Global and region based motion estimation - multiresolution motion estimation - Feature based motion estimation. Stereo and multi-view sequence processing: Depth perception - Stereo imaging principle - Disparity estimation.

References

1. Rafael C. Gonzalez and Richard E. Woods: Digital Image Processing, 4e, Pearson, 2017.
2. Anil K. Jain: Fundamentals of Digital Image Processing, 1e, Pearson, 1988.
3. William K. Pratt: Digital Image Processing, 4e, John Wiley & Sons, 2007.
4. Azriel Rosenfield and Avinash C. Kak: Digital Picture Processing, 2e, Morgan Kaufmann, 1982.
5. Bernd Jahne: Digital Image Processing, 6e, Springer, 2005.
6. Yao Wang, Jorn Ostermann and Ya-Qin Zhang: Video Processing and Communications, 1e, Pearson, 2001.
7. Alan C. Bovik: The Essential Guide to Video Processing, 2e, Academic Press, 2009.
8. A. Murat Tekalp: Digital Video Processing, 2e, Prentice Hall, 2015.

23-813-0703: Deep Learning

Core/Elective: **Core** Semester: **7** Credits: **4**

Course Description

Deep learning is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. This course describes deep learning techniques used by practitioners in industry, including deep feedforward networks, regularization, optimization algorithms, convolutional networks, sequence modeling, and practical methodology. This course is useful to students planning careers in either industry or research, and for software engineers who want to begin using deep learning in their products or platforms.

Course Outcomes (CO)

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

CO1:Understand the need for Deep learning, Feed forward networks, Learning XOR, Gradient based Learning, Hidden units.
CO2: Differentiate between training error and generalization error, Underfitting and Overfitting.
CO3: Identify Regularization strategies, Dataset Augmentation, Adversarial Training.
CO4: Demonstrate the working of Convolution Operation, Sparse interactions, Parameter sharing, Equivariant representations, Pooling.
CO5: Describe Recurrent Neural Networks, Recurrent networks with a single output
CO6: Understand different types of Autoencoders, Undercomplete Autoencoders, Regularized Autoencoders, Dimensionality Reduction.
CO7: Explain Deep generative models like Boltzmann Machines, Restricted Boltzmann Machines.

Course Content

1. Deep Networks: Feed forward networks – Learning XOR- Gradient based Learning – Hidden units – Architecture design- Back propagation – Differentiation algorithms.

2. Regularization for Deep Learning: Penalties-Constrained optimization-Under constrained problems - Dataset augmentation-Semi Supervised learning- Sparse representation- Adversarial training - Optimization for training deep models: Basic algorithms-Algorithms with adaptive learning rates.

3. Convolutional Networks: Convolution-Pooling-Variants of pooling- Efficient convolutional algorithms – Recurrent and Recursive Nets: Recurrent Neural Networks-Deep Recurrent Networks - Recursive Neural Networks- Explicit memory.

4. Linear Factor Models: Probabilistic PCA- ICA – Slow feature analysis – Sparse coding – Autoencoders: Undercomplete Autoencoders – Regularized Autoencoders- Learning Manifolds - Applications of Autoencoders – Representation learning.

5. Deep generative models: Boltzmann Machines – RBM - Deep Belief Networks - Deep Boltzmann Machines-Convolutional Boltzmann Machines- Directed generative Nets.

References

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, 1e, MIT Press, 2017.
2. Nikhil Buduma and Nicholas Locascio, Fundamentals of Deep Learning: Designing Next Generation Machine Intelligence Algorithms, 1e, Shroff/O'Reilly, 2017.
3. Josh Patterson and Adam Gibson, Deep Learning: A Practitioner's Approach, 1e, Shroff/O'Reilly, 2017.

23-813-0704: Lab 12: Computational Linguistics Lab

Core/Elective: **Core** Semester: **7** Credits: **1**

Course Description

The objective of Natural Language Processing lab is to introduce the students with the basics of NLP which will empower them for developing advanced NLP tools and solving practical problems in the field. The experiments in this lab are arranged in a logical sequence to inculcate a new concept at every step, starting from very basic ones to advanced ones.

Course Outcomes (CO)

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

CO1: Understand morphological features of a word by analysing it.
CO2: Calculate bigrams from a given corpus and calculate probability of a sentence
CO3: Calculate emission and transition matrix which will be helpful for tagging Parts of Speech using Hidden Markov Model.
CO4: know the importance of context and size of training corpus in learning Parts of Speech

Course Content

Following algorithms need to be implemented by any programming language

- Word Analysis
- Word Generation
- Morphology
- N-Grams
- N-Grams Smoothing
- POS Tagging: Hidden Markov Model
- POS Tagging: Viterbi Decoding
- Building POS Tagger
- Chunking
- Building Chunker

References

1. Jurafsky and Martin, Speech and Language Processing, Prentice Hall, 1e, 2000
2. Akshar Bharati, Rajeev Sangal and Vineet Chaitanya: "Natural Language Processing: A Paninian Perspective", Prentice-Hall of India , 1e, 1995
3. https://nlp-iiith.vlabs.ac.in/

23-813-0705: Lab 13: Image and Video Processing Lab

Core/Elective: **Core** Semester: **7** Credits: **1**

Course Description

The aim of this course is to provide practical knowledge to students in implementing image and video processing algorithms using any programming language. The students will be able to analyse the performance of various image and video processing algorithms both qualitatively and quantitatively. As part of this course students will get an opportunity to do a research-oriented mini project thereby getting an exposure to the recent research developments in the domain of Digital Image and Video Processing.

Course Outcomes (CO)

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

CO1: Implement basic image processing algorithms in spatial and frequency domain.
CO2: Understand the convolution and correlation operations on images.
CO3: Apply different image transforms.
CO4: Analyse the performance of different image restoration and enhancement algorithms.
CO5: Apply different color image processing techniques.
CO6: Analyse various segmentation algorithms.
CO7: Implements different video processing techniques.
CO5: Solve real life problems using image processing and machine learning algorithms.

Course Content

Following experiments need to be done in the lab

- Convolution and Correlation.
- Image transforms.
- Intensity transformation functions.
- Histogram equalization and matching.
- Image enhancement in spatial and frequency domain.
- Image restoration in spatial and frequency domain.
- Color image processing.
- Hough transform.
- Image segmentation algorithms.
- Video processing.

23-813-0706: Reinforcement Learning

Core/Elective: **Elective** Semester: **7** Credits: **4**

Course Description

The course aims to introduce the concepts of reinforcement learning and to impart an understanding of how reinforcement learning -- along with supervised and unsupervised learning -- form a building block of modern artificial intelligence. The course will provide a solid introduction to the field of reinforcement learning and students will learn about the core challenges and approaches, including generalization and exploration.

Course Outcomes (CO)

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

CO1: Define the key features of reinforcement learning that distinguishes it from AI and non-interactive machine learning
CO2: Demonstrate the ability to formulate a given problem as a reinforcement problem with all ingredients.
CO3: Implement in code common RL algorithms
CO4: Describe the exploration vs exploitation challenge
CO5: Compare and contrast at least two approaches for addressing the above challenge.

Course Content

1. The Reinforcement Learning problem: evaluative feedback, non-associative learning, Rewards and returns, Markov Decision Processes, Value functions, optimality and approximation

2. Bandit Problems: Explore-exploit dilemma, Binary Bandits, Learning automata, exploration schemes Dynamic programming: value iteration, policy iteration, asynchronous DP, generalized policy iteration

3. Monte-Carlo methods: policy evaluation, roll outs, on policy and off policy learning, importance sampling Temporal Difference learning: TD prediction, Optimality of TD(0), SARSA, Q-learning, Rlearning, Games and after states

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

5. Function Approximation: Value prediction, gradient descent methods, linear function approximation, Control algorithms, Fitted Iterative Methods Policy Gradient methods: nonassociative learning - REINFORCE algorithm, exact gradient methods, estimating gradients, approximate policy gradient algorithms, actor-critic methods Hierarchical RL: MAXQ framework, Options framework, HAM framework, Option discovery algorithms

References

1. R. S. Sutton and A. G. Barto, Reinforcement Learning - An Introduction. 2e, MIT Press, 2018 eBook: http://incompleteideas.net/book/the-book-2nd.html
2. Marco Wiering and Martijn van Otterlo (Editors), Reinforcement Learning: State-of-the Art, Springer, 2012
3. Csaba Szepesvari, Algorithms for Reinforcement Learning, Morgan and Claypool Publishers, 2010
4. David Silver: https://www.davidsilver.uk/teaching/

23-813-0707: Computer Vision

Core/Elective: **Elective** Semester: **7** Credits: **4**

Course Description

This course introduces the basic concepts and applications in computer vision. The course covers different feature extraction techniques as well as different segmentation and classification techniques. It includes vision tasks like object detection, recognition and motion detection. The content of the course also includes practical exercises to help the students formulating and solving computer vision problems.

Course Outcomes (CO)

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

CO1: Understand the fundamental theories and techniques of human vision with computer vision.
CO2: Learn the process of image formation in the camera.
CO3: Apply different types of morphological operations to an image.
CO4: Apply different region properties in an image.
CO5: Summarize different texture, color-based feature extraction methods used for computer vision.
CO6: Understand the working of the Camera calibration system.
CO7: Learn different methods to compute the motion of an object from 2D image sequences.
CO8: Understand the process of the depth information from stereo images.
CO9: Develop a computer-based system with vision capabilities.

Course Content

1. Imaging and Image Representation: Imaging Devices, 3D structure from 2D images, Five frames of reference. Binary Image Analysis: Pixels and Neighborhoods, Applying masks to images, Counting the objects in an image, Connected components labeling. Binary image morphology, Region properties, Region adjacency graphs.

2. Feature detection and matching: Points and patches, SIFT, Edges-Edge detection and linking, Lines-Hough transforms. Color and Shading: Color bases, Color histograms, Color segmentation, Shading. Texture: Texture, Texels and Statistics, Texel based Texture Descriptions, Quantitative Texture Measures, Texture Segmentation.

3. Content based image retrieval: Image distance measures: Color similarity,Texture similarity, Shape similarity, Database organization. Motion from 2D image sequences: Computing Motion Vectors, Computing paths of moving points, Detecting significant changes in video.

4. Matching in 2D: Registration of 2D data, Representation of points, Affine mapping functions, 2D object recognition via Affine Mapping: Local Feature Focus method, Pose clustering, Geometric hashing, 2D object recognition via Relational Matching.

5. Perceiving 3D from 2D images: Labeling of line drawings from blocks world, 3D cues available in 2D images, Perspective imaging model, Depth perception from Stereo- Establishing correspondences - 3D sensing and Object pose Computation - 3D Affine transformations, Camera Model, Affine calibration matrix, Improved Camera calibration method, Pose estimation, 3D object reconstruction.

References

1. Linda G. Shapiro, George C. Stockman, Computer Vision, 1e, Prentice Hall, 2001.
2. Richard Szeliski, Computer Vision: Algorithms and Applications, 1e, Springer, 2010.
3. David A. Forsyth and Jean Ponce, Computer Vision: A Modern Approach, 2e, Pearson Education, 2015.
4. Simon J. D. Prince, Computer Vision: Models, Learning, and Inference, 1e, Cambridge University Press, 2012.
5. Ramesh Jain, RangacharKasturi, Brian G. Schunck, Machine Vision, 1e, McGraw-Hill, 1995.
6. E.R. Davies, Computer Vision: Principles, Algorithms, Applications, Learning, 5e, Academic Press, 2017.
7. J. R. Parker, Algorithms for Image Processing and Computer Vision, 2e, Wiley, 2010.

23-813-0708: Virtualized Systems

Core/Elective: **Elective** Semester: **7** Credits: **4**

Course Description

Virtualization provides the benefit of reducing the total cost of ownership and improving the business agility. This course systematically introduces the concepts and techniques used to implement the major components of virtual servers behind the scene. It discusses the details on hypervisor, CPU scheduling, memory management, virtual I/O devices, mobility, and etc.

Course Outcomes (CO)

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

CO1: Explain virtualization concepts.
CO2: Illustrate the merits of server virtualization.
CO3: Explain CPU virtualization.
CO4: Identify the roles of type 1 and type 2 hypervisors.
CO5: Explain the memory management techniques for virtualized systems.
CO6: Explain the methods of virtualization at I/O levels.
CO7: Outline the virtualization support for different cloud computing models.

Course Content

1. Overview: Why server virtualization –History and re-emergence – General structures – Architectures comparison - Commercial solutions – VMWare, Xen.

2. Virtual machines: CPU virtualization -Privileged instructions handling -Hypervisor – Paravirtualization - Hardware-assisted virtualization - Booting up - Time keeping – CPU scheduling- Commercial examples.

3. Memory management in virtualization: partitioning –reclamation –ballooning. Memory sharing. OS-level virtualization –VMWare –Red Hat Enterprise Virtualization.

4. I/O virtualization: Virtualizing I/O devices -monolithic model -virtual I/O server – Virtual networking –tunneling –overlay networks - Commercial examples. Virtual storage: Granularity - file system level–blocks level.

5. Virtualized computing: Virtual machine based distributed computing - elastic cloud computing clustering - cold and hot migration - Commercial examples - Challenges and future trends.

References

1. Jim Smith and Ravi Nair: Virtual Machines: Versatile Platforms for Systems and Processes, 1e, Morgan Kaufmann, 2005.

2. Sean Campbell: Applied Virtualization Technology -Usage models for IT professionals and Software Developers, 1e, Intel Press, 2006.
3. Matthew Portnoy: Virtualization Essentials, 1e, JW, 2012.
4. George Trujillo, Charles Kim, Steve Jones, Rommel Gracia and Justin Murray: Virtualizing Hadoop, VM Press, 2015.

23-813-0709: Advanced Optimization Techniques

Core/Elective: **Elective** Semester: **7** Credits: **4**

Course Description

This course is about the well-known population-based optimization techniques developed during the last three decades. This course emphasizes on the advanced optimization techniques to solve large-scale problems especially with nonlinear objective functions.

Course Outcomes (CO)

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

CO1: Understand the basic concepts of optimization and its applications.
CO2: Understand the mathematical representation and classical methods for solving optimization problems.
CO3: Explain and demonstrate working principles of various population based optimization techniques.
CO4: Explain and Demonstrate working principle of various Hybrid Algorithms for optimization.

Course Content

1. Introduction to optimization- formulation of optimization problems-Review of classical methods-Linear programming-Nonlinear programming-Constraint optimality criteria-constrained optimization-Population based optimization techniques.

2. Genetic Algorithm - Introduction - Working principle - Representation - selection - fitness assignment - reproduction - crossover - mutation - constraint handling -advanced genetic algorithms - Applications - Artificial Immune Algorithm - Introduction- Clonal selection algorithm- Negative selection algorithm - Immune network algorithms - Dendritic cell algorithms.

3. Differential Evolution - Introduction - Working principles - parameter selection - advanced algorithms in Differential evolution - Biogeography-based Optimization - Introduction - Working Principles - Algorithmic variations.

4. Particle Swarm Optimization-Introduction- Working principles- Parameter selection-Neighborhoods and Topologies-Convergence-Artificial Bee Colony Algorithm-Introduction-Working principles- Applications-Cuckoo search based algorithm-Introduction- Working principles- Random walks and the step size-Modified cuckoo search.

5. Hybrid Algorithms-Concepts- divide and conquer- decrease and conquer-HPABC-HBABC- HDABC-HGABC-Shuffled Frog Leaping Algorithm - Working principles - Parameters- Grenade Explosion Algorithm-Working principle-Applications

References

1. Dan Simon, Evolutionary Optimization Algorithms, 1e, Wiley, 2013
2. Xin-She Yang, Engineering Optimization: An Introduction with Meta-heuristic Applications, 1e, Wiley, 2010
3. S.S. Rao, Engineering Optimization: Theory and Practice, 4e,New Age International, 2013
4. R. VenkataRao, Teaching Learning Based Optimization Algorithm: And Its Engineering Applications, 1e, Springer, 2016

23-813-0710: Bioinformatics

Core/Elective: **Elective** Semester: **7** Credits: **4**

Course Description

Present fundamental concepts from molecular biology, computational problems in molecular biology and some efficient algorithms that have been proposed to solve them.

Course Outcomes (CO)

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

CO1: Understand and appreciate basic concepts of molecular Biology and Human genome project.
CO2: Illustrate and explain various sequence alignment algorithms.
CO3: Understand the basic concepts of Fragment Assembly of DNA and demonstrate various algorithms for the same.
CO4: Demonstrate and evaluate different algorithms for identifying optimal phylogenetic trees.
CO5: Understand the concepts of structure prediction in molecular biology.
CO6: Understand and demonstrate an algorithm in the literature for the domain.

Course Content

1. Basic concepts of molecular Biology-Proteins-Nucleic acids– genes and genetic synthesis – translation- transcription- protein Synthesis- Chromosomes- Maps and sequences- human genome project- sequence databases.

2. Strings-Graphs-Algorithms- Comparing 2 sequences- Global & Local comparison-General Gap Penalty Function-Affix gap penalty function-comparing multiple sequences-Star alignments-Tree alignments-Database Search-PAM matrices BLAST-FAST –Issues.

3. Fragment Assembly of DNA-Biological Background –Models-Algorithms-Heuristics-Physical Mapping of DNA-Restriction site Mapping-site models-Internal Graph Models –Hybridization Mapping-Heuristics.

4. Phylogenetic Trees –Binary Character States-Parsimony and Compatibility in Phylogenies-Algorithm for Distance Matrices-Additive Trees- Genome rearrangements-Oriented Blocks-unoriented Blocks.

5. Molecular Structure Prediction- RNA secondary structure prediction-Protein Folding problems-Protein threading-Computing with DNA-Hamilton Path Problems.

References

1. Neil James and Pavel A Pevzner, An introduction to Bioinformatics Algorithms, 4e, OUPress, 2014
2. ZhumurGhosh, BibekanandMallick , Bioinformatics : Principles and Applications, OUPress, 2015
3. Concord Bessant, Darren Oakley, Ian Shadforth, Building Bioinformatics Solutions, OUPress, 2014
4. Peter Clote and Rolf Backofen, Computational Molecular Biology-An introduction, 1e, Wiley Series, 2000

23-813-0711: Algorithms for Modern Data Models

Core/Elective: **Elective** Semester: **7** Credits: **4**

Course Description:

There exist both algorithmic and statistical challenges in modern large-scale applications and data analysis. This course describes the randomization and probabilistic techniques for modern computer science, with applications ranging from combinatorial optimization and machine learning to communication networks. The course covers the core material to advanced concepts. Also the emphasis is on methods useful in practice.

Course Outcomes (CO)

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

CO1: Relate the advanced concepts of probability theory and modern applications.
CO2: Explain the uncertainty in prediction due to intervention of random variables..
CO3: Examine random graphs and their properties.
CO4: Analyze evolutionary algorithms.
CO5: Interpret algorithms for evolving data streams
CO6: Build algorithms for new problems with volume of data

Course Content:

1. Probability: Expectations - Tail Bounds - Chernoff Bound – Balls and Bins – Probabilistic Method – Markov chains and Random walks.

2. Entropy, Randomness, and Information: Measure of randomness – Monte Carlo Method – Markov Chain Monte Carlo Method.

3. Graph models and algorithms– Random graph Models- Algorithms for graph generation - Random graphs as models of networks, Power laws, Small world Phenomena.

4. Components of evolutionary algorithms – Example applications – Genetic algorithms – Evolution strategies – Evolutionary programming.

5. Sampling, sketching, data stream models, read-write streams, stream-sort, map-reduce - Algorithms in evolving data streams

References

1. Michael Mitzenmacher, Eli Upfal, Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis, 2e, Cambridge

University Press, 2017.
2. Rajeev Motwani and PrabhakarRaghavan, Randomized Algorithms, Cambridge University Press; Reprint edition, 2010.
3. S. Muthukrishnan, Data Streams: Algorithms and Applications, 1e, Now Publishers, 2005.
4. Charu C. Aggarwal, Data Streams: Models and Algorithms, 1e, Springer, 2006.
5. Agoston E. Eiben, J.E. Smith , Introduction to evolutionary computing, 1e, Springer, 2010.

23-813-0712: Complex Network Analysis

Core/Elective: **Elective** Semester: **7** Credits: **4**

Course Description

Complex networks provide a powerful abstraction of the structure and dynamics of diverse kinds of interaction viz people or people-to-technology, as it is encountered in today's inter-linked world. This course provides the necessary theory for understanding complex networks and applications built on such backgrounds.

Course Outcomes (CO)

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

CO1: Explain and appreciate complex networks and complex network systems as different from other network systems viz. computer networks, transportation networks etc.
CO2: Explain the mathematical representation of complex networks in computer programs.
CO3: Explain and compute the centrality measures in network analysis.
CO4: Demonstrate random graph generation processes and associated properties.
CO5: Discriminate various algorithms for community detection in complex networks.
CO6: Evaluate different models for complex networks.
CO7: Illustrate and explain the flow models used in complex networks for modelling social, economic and biological systems.
CO8: Identify the performance requirements related to social media systems.
CO9: Explain the techniques for predictive modelling and analytics of social media data.

Course Content

1. Networks of information – Mathematics of networks – Measures and metrics – Large scale structure of networks – Matrix algorithms and graph partitioning.

2. Network models – Random graphs – walks on graphs - Community discovery – Models of network formation – Small world model - Evolution in social networks – Assortative mixing- Real networks - Evolution of random network - Watts-Strogatz model – Clustering coefficient - Power Laws and Scale-Free Networks – Hubs - Barabasi-Albert model – measuring preferential attachment- Degree dynamics – nonlinear preferential attachment.

3. Processes on networks – Percolation and network resilience – Epidemics on networks – Epidemic modelling - Cascading failures – building robustness- Dynamical systems on networks – The Bianconi-Barabási model – fitness measurement – Bose-Einstein condensation.

4. Models for social influence analysis – Systems for expert location – Link prediction – privacy analysis – visualization – Data and text mining in social networks - Social tagging.

5. Social media - Analytics and predictive models – Information flow – Modelling and prediction of flow - Missing data - Social media datasets – patterns of information attention – linear influence model – Rich interactions.

References

1. Mark J. Newman, Networks: An introduction, 1e, Oxford University Press, 2010.
2. Charu C Aggarwal (ed.), Social Network Data Analytics, 1e, Springer, 2011.
3. David Easley and Jon Kleinberg, Networks, Crowds, and Markets: Reasoning about a highly connected World, 1e, Cambridge University Press, 2010.
4. Albert-Laszlo Barabasi, Network Science, 1e, Cambridge University Press, 2016.

23-813-0801: Probabilistic Graphical Models

Core/Elective: **Core** Semester: **8** Credits: **4**

Course Description

Probabilistic Graphical models (PGM) are a foundation for understanding many methods of artificial intelligence, machine learning and estimation. Machine learning provides algorithms for solving problems by using training data. This course will give insight into how to formulate problems so that machine learning can be used effectively. Building good models can help learn with less data by constraining the learning space. Bayesian models are at the heart of most estimation methods. Formulation of these models is the first step in developing an estimation algorithm. The estimation itself is in many cases just inference on the model given some evidence. Approximate inference techniques such as those covered in this course are important in solving many very hard estimation problems in science and engineering. Data scientists, machine learning enthusiasts, engineers, and those who are curious about the latest advances in machine learning will find PGM interesting.

Course Outcomes (CO)

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

CO1: Demonstrate application of Probability and Graph Theory in reasoning.
CO2: Discuss how different graphs represent both factorization and independent relations.
CO3: Utilize message passing algorithms for inference.
CO4: Examine methods for learning uncertainties in a model's parameters.
CO5: Experiment with graph building tools.
CO6: Apply Bayesian networks and Markov networks to many real world problems.

Course Content:

1. Probabilistic reasoning: Representing uncertainty with probabilities – Random variables and joint distributions – Independence – Querying a distribution - Graphs.

2. Representation: Bayesian Network (BN) representation – Independencies in BN – Factorizing a distribution – D-separation- Algorithm for D-separation – From distributions to Graphs.

3. Undirected Graphical Models: Factor products – Gibbs distribution and Markov networks – Markov network independencies – Factor graphs – Learning parameters – Conditional Random Fields.

4. Gaussian Network Models: Multivariate Gaussians – Gaussian Bayesian networks – Gaussian Markov Random Fields – Exact Inference: variable elimination- Sum-product and belief updates – The Junction tree algorithm.

5. Learning: Learning Graphical Models – Learning as optimization – Learning tasks – Parameter estimation – Structure learning in BN – Learning undirected models – Actions and decisions.

References:

1. Daphne Koller and Nir Friedman: Probabilistic Graphical Models- Principles and Techniques, 1e, MIT Press, 2009.
2. Richard E. Neapolitan: Learning Bayesian Networks, 1e, Pearson, 2019.
3. Christian Borgelt, Rudolf Kruse and Matthias Steinbrecher: Graphical Models- Methods for data analysis and Mining, 2e, Wiley, 2009.
4. David Bellot: Learning Probabilistic Graphical Models in R, Packt Publishing, 1e, 2016.
5. Luis Enrique Sucar: Probabilistic Graphical Models, 1e, Springer Nature, 2015.
6. Coursera: https://www.coursera.org/specializations/probabilistic-graphical-models

23-813-0802: Algorithms for Massive Datasets

Core/Elective: **Core** Semester: **8** Credits: **4**

Course Description

Big Data concerns large-volume, complex, growing data sets with multiple, autonomous sources. With the fast development of networking, data storage, and the data collection capacity, Big Data is now rapidly expanding in all science and engineering domains. The traditional data mining algorithms also need to be adapted for dealing with the ever-expanding datasets of tremendous volume.

Course Outcomes (CO)

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

CO1: Build practical skills on developing MapReduce jobs.
CO2: Explain the search algorithms that perform efficiently on massive datasets.
CO3: Explain the algorithms for data stream processing.
CO4: Explain the link analysis methods in the context of social networks and Page rank algorithms.
CO5: Demonstrate the power of some of the online algorithms for massive data.
CO6: Explain the randomized, approximate and one-pass algorithms for mining from massive datasets.

Course Content:

1. Introduction to MapReduce – the map and reduce tasks, MapReduce workflow, fault tolerance. - Algorithms for MapReduce – matrix multiplication, relational algebra operations- Complexity theory for MapReduce.

2. Locality-Sensitive Hashing - shingling of documents, min-hashing. Distance measures, nearest neighbors, frequent itemsets- LSH families for distance measures, Applications of LSH- Challenges when sampling from massive data.

3. Mining data streams – stream model, stream data sampling, filtering streams – bloom filters, counting distinct elements in a stream - Flajolet-Martin algorithm. Moment estimates - Alon-Matias-Szegedy algorithm, counting problems for streams, decaying windows.

4. MapReduce and link analysis- PageRank iteration using MapReduce, topic-sensitive PageRank - On-line algorithms – Greedy algorithms, matching problem, the adwords problem – the balance algorithm.

5. Computational model for data mining – storage, cost model, and main memory bottleneck. Hash based algorithm for mining association rule – improvements to a-priori, park-chen-yu

algorithm, multistage algorithm, approximate algorithm, limited-pass algorithms – simple randomized algorithm, Savasere, Omiecinski, and Navathe algorithm, Toivonen algorithm.

References:

1. Jure Leskovec, Rajaraman, A. and Ullman, J. D., Mining of Massive Datasets, Cambridge University Press, 2nd edition, 2016.
2. Charu C. Aggarwal, Data Streams: Models and Algorithms, 1e, Springer, 2007.
3. Michael I Jordan, Frontiers in Massive Data analysis, 1e, National Academies Press, 2013.
4. Nathan Marz and James Warren, Big Data: Principles and best practices of scalable realtime data systems, Manning Publications, 2015.

23-813-0803: Professional Communication

Core/Elective: **Core** Semester: **8** Credits: **2**

Course Description

This course is aimed at equipping students with communication skills that are relevant for professional and advanced academic contexts

Course Outcomes (CO)

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

CO1: Understand the nature and objective of Technical Communication relevant for the workplace
CO2: Utilize the technical writing for the purposes of Technical Communication and its exposure in various dimensions
CO3: Enhance confidence in face of diverse audience
CO4: Evaluate their efficacy as fluent & efficient communicators by learning the voice-dynamics
CO5: Understand the Soft Skills required in a professional environment

Course Content

1. Technical Communication: features: Distinction between General And Technical Communication; Language as a tool of communications; Levels of communication: Interpersonal, Organizational, Mass communication; The flow of communication: Downward, Upward, Lateral/Horizontal (Peer group) : Importance of technical communication; Barriers to Communication
2. Words and Phrases: Word formation, Synonyms and Antonyms; Homophones; Select vocabulary of about 500-1000 New words; correct Usage: all Parts of Speech; Modals; Concord; Articles; Infinitives; Transformation of sentences; Requisites of Sentence Construction: Paragraph Development: Techniques and Methods Inductive, Deductive, Spatial , Linear, and Chronological.
3. Principles, Sales & Credit letters; Claim and Adjustment Letters; Job Application and Resumes. Reports: Types; Significance; Structure, Style & Writing of Reports. Technical Proposal; Parts; Types; Writing of Proposal; Significance; Negotiation skills
4. Nuances and Modes of Delivery; Body Language; Dimensions of Speech: Syllable; Accent; Pitch; Rhythm; Intonation; Paralinguistic features of voice; Interpersonal communication: Definition; Types; Team work; Attitude; Way to improve Attitude Listening Skills : Types; Methods for improving Listening Skills
5. Following essays from the prescribed text book with emphasis on Mechanics of writing. (i) Humanistic and Scientific Approaches to Human Activity by Moody E. Prior (ii) The Language of Literature and Science by A. Huxley (iii) Man and Nature by J. Bronowski (iv) Science and Survival by Barry Commoner (v) The Mother of the Sciences by A.J. Bahm

References

1. John M. Lannon, Laura J. Gurak, Technical Communication, 14e, Pearson, 2016
2. Mike Markel, Stuart A. Selber, Technical Communication, 12e, Bedford, 2017
3. Paul MacRae, Business and Professional Writing: A Basic Guide, 2e, Broadview Press, 2019
4. Goodheart-Willcox, Soft Skills for the workplace, 1e, Goodheart-Willcox publishing, 2016
5. V.N. Arora and Laxmi Chandra, Improve your Writing, 1e, Oxford Univ. Press, 2013.
6. Meenakshi Raman & Sangeeta Sharma, Technical Communication- Principles and Practices by, 3e, Oxford Univ. Press, 2015
7. Neelam Saxena, Functional skills in Language and Literature, 1e, ABD Publishers, 2018.

23-813-0804: Mini Project

Core/Elective: **Core** Semester: **8** Credits: **1**

Course Description

Through the project work, the student has to exhibit the knowledge in terms of engineering or technological innovation or research ability to solve the contemporary problem. On completion of the work, the student shall submit a project report. The qualitative and quantitative results of the work will be evaluated through a viva voce exam.

Course Outcomes (CO)

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

CO1: Demonstrates in depth knowledge and thoughtful application through the detailed analysis of the problem chosen for the study.
CO2: Assess the gap by acquiring knowledge about the previous works, and its interpretation and application.
CO3: Demonstrates the design of the proposed methodology and its merits.
CO4: Organize the interim project content with proper structure and sequencing.
CO5: Demonstrate the academic discussion skills to emphasize, argue with clarity of purpose using evidence for the claims.

23-813-0813: Dissertation

Core/Elective: **Elective** Semester: **8** Credits: **12**

Course Description :

This course is only for those students who have opted to take B.Sc (Honours with Research) in Computer Science in their eighth semester and is in lieu of the Elective courses III, IV and V. The aim of the research-oriented dissertation is to provide the students with a comprehensive understanding of the research process and equip them with the necessary skills to conduct independent research in their chosen area of computer science. By the end of the course, the student should have completed a substantial research project that showcases their ability to conduct rigorous research and to effectively communicate the research outcome through a research report and dissertation.

On completion of the work, they shall communicate their research to a reputed peer-reviewed Journal/Conference and also submit a Final Dissertation report. The qualitative and quantitative results of the work will be evaluated through a viva-voce exam conducted by an expert panel.

Course Outcomes (CO)

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

CO1: Demonstrates in-depth knowledge and thoughtful application through the detailed analysis of the problem chosen for the study
CO2: Assess the gap by acquiring knowledge about the previous works, and its interpretation and application
CO3: Demonstrates the design of the proposed methodology and its merits.
CO4: Interpret research results, evaluate significance and limitations, and analyze findings within existing knowledge.
CO5: Demonstrate the academic discussion skills to emphasize, and argue with clarity of purpose using evidence for the claims.
CO6: Present research findings through clear written and oral communication to technical and non-technical audiences.

23-813-0806: Algorithmic Game Theory

Core/Elective: **Elective** Semester: **8** Credits: **4**

Course Description

Game theory is a branch of mathematics and economics which models interactions of agents as games. Algorithmic game theory is the intersection of game theory and computer science. This course introduces algorithmic game theory in an application-oriented manner.

Course Outcomes (CO)

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

CO1: Explain the fundamental concepts of non-co-operative and cooperative game theory
CO2: Distinguish between standard game models and solution concepts.
CO3: Illustrate a variety of advanced algorithmic techniques and complexity results for computing game theoretical solution concepts
CO4: Identify rationale of decision making in games.
CO5:Apply solution concepts, algorithms, and complexity results to unseen games that are variants of known examples.
CO6: Compare the state of the art in some areas of algorithmic research, including new developments and open problems

Course Content

1. Introduction to game theory – strategies, costs, payoffs – solution concepts – finding equilibria – games with sequential moves – games with simultaneous moves – discrete strategies, continuous strategies – mixed strategies – games with incomplete information – expected payoffs – Prisoner's dilemma and repeated games – Nash equilibrium – Computational complexity of Nash equilibrium
2. Games on networks – congestion games – selfish routing – Nash and wardrop equilibria for networks – price of anarchy – pricing network edges – network design with selfish agents – economic aspects of internet routing
3. Epistemic game theory – Modeling knowledge – rationality and belief – common belief in rationality – game strategies and perfect recall – cryptography and game theory – modeling cryptographic algorithms as games – multi-party computations – MPC and games
4. Mechanism design – general principles – social choice – incentives – algorithms mechanism design – distributed aspects – cost-sharing mechanisms – mechanism design without money – house allocation problem – stable matchings
5. Voting – evaluation of voting systems – strategic manipulation of votes – auctions – types of auctions – winner's curse – bidding strategies – fairness in auctions

References

1. Avinash K. Dixit et al., Games of Strategy, 4e, W. W. Norton & Company, 2014
2. Noam Nisan et al., Algorithmic Game Theory, 1e, Cambridge University Press, 2007
3. Steven Tadelis, Game Theory: An Introduction, 1e, Princeton University Press, 2013.
4. Michael Maschler, et al., Game Theory, 1e, Cambridge University Press, 2013.
5. Andres Perea, Epistemic Game Theory: Reasoning and Choice, 1e, Cambridge University Press, 2012

23-813-0807: Deep Learning for Computer Vision

Core/Elective: **Elective** Semester: **8** Credits: **4**

Course Description

Deep Learning has fundamentally changed the landscapes of a number of areas in artificial intelligence, including Computer Vision (CV), Natural Language Processing (NLP) and game playing. Intersection of Deep Learning and CV have emerged with interesting results serving a benchmark for the advances in one of the most important tasks in artificial intelligence. This course introduces the state of the art of deep learning research and its successful applications to major CV tasks, including image classification, object detection, image captioning, video classification, and image generation. The course will help the students to gain practical insights into tools and techniques to implement CV projects effectively and also to outline and analyze various research frontiers of CV in the deep learning era.

Course Outcomes (CO)

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

CO1: Understand the foundations of Computer Vision.
CO2: Evaluate the different feature extraction techniques.
CO3: Develop an object recognition system using handcrafted features and classifiers.
CO4: Understand the concepts of neural networks and logistic regression.
CO5: Develop an image classification system using minimal neural network.
CO6: Understand the fundamentals of Convolutional Neural Network (CNN).
CO7: Develop an image classification system using CNN.
CO8: Develop an image classification system using pre-trained models or auto-encoders.
CO9: Understand the fundamentals of adversarial networks.
CO10: Develop a video processing application using CNN.

Course Content

1. Introduction to Computer Vision - Image Filtering - Interest Point Detection - Feature Extraction - Geometric features - SIFT, SURF, HOG, WLD, LBP. Recognition: Geometry-based – Appearance based. Applications: Object recognition - Face recognition. Implementation: Object Recognition using hand-crafted features and classifiers.

2. Neural Networks – Stochastic Gradient Descent – Backpropagation – Logistic Regression – Softmax. Implementation:Image Classification using minimal neural network.

3. Convolutional Neural Networks: Building Blocks – Hyperparameter Tuning – Learning – Visualizing CNNs – Batch Normalization and Dropout – Deconvnets. Implementation: Simple Image classification using CNN.

4. Transfer Learning – Pre-trained Models – Autoencoders. Implementation: Image Classification using Pre-trained models/Autoencoders.

5. Generative Adversarial Network (GAN) – Attention Mechanism – YOLO. Application: Video Classification-Streaming CNN for action recognition - 3D convolution for temporal learning – Segmenting and captioning videos. Implementation: Video Classification / Summarization / Anomaly Detection using CNN.

References

1. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, The MIT Press, 2017.
2. Charu C Aggarwal, Neural Networks and Deep Learning, Springer, 2018.
3. Eugene Charniak, Introduction to Deep Learning, The MIT Press, 2019.
4. Linda G. Shapiro and George C. Stockman, Computer Vision, 1e, Prentice Hall, 2001.
5. Richard Szeliski, Computer Vision: Algorithms and Applications, 1e, Springer, 2010.
6. David A. Forsyth and Jean Ponce, Computer Vision: A Modern Approach, 2e,Pearson Education, 2011.
7. Simon J. D. Prince: Computer Vision: Models, Learning, and Inference, 1e, Cambridge University Press, 2012.
8. Ramesh Jain, Rangachar Kasturi and Brian G. Schunck, Machine Vision, 1e, McGraw-Hill, 1995.

23-813-0808: Natural Language Processing with Deep Learning

Core/Elective: **Elective** Semester: **8** Credits: **4**

Course Description

Deep learning has fundamentally changed the landscapes of a number of areas in artificial intelligence, including speech and natural language, vision, robotics, and game playing. Intersection of deep learning and NLP have emerged with interesting results serving a benchmark for the advances in one of the most important tasks in artificial intelligence. This course introduces the state of the art of deep learning research and its successful applications to major NLP tasks, including speech recognition and understanding, dialogue systems, lexical analysis, parsing, knowledge graphs, machine translation, question answering, sentiment analysis, social computing, and natural language generation.

Course Outcomes (CO)

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

CO1: Outline the foundations of speech and written language processing.
CO2: Analyse various NLP tasks through the lens of deep learning.
CO3: Implement an NLP project effectively.
CO4: Analyze various research frontiers of NLP in the deep learning era.

Course Content

1. Word embeddings: SVD based methods – Iteration based methods – Word2vec – optimization Language models: Unigram, Bigram – CBOW – Skip-Gram Model - GloVe – Evaluation of Word Vectors: Intrinsic and Extrinsic tasks – Word Window Classification Implementations: Gensim Word2vec, Doc2Vec, FastText -GloVe using gradient descent.

2. Neural Networks Architectures: feed-forward computation – representational power – back propagation and computation graphs Implementations: Neural network and Deep neural network in ML framework.

3. Linguistic structure: Dependency parsing – N-gram language models – Recurrent Neural Networks and language models – Sequence modeling: Recurrent and Recursive Neural Nets Implementations: Character level RNN, Chatbot using recurrent sequence to sequence models.

4. Machine translation – Seq2Seq learning - Attention models– ConvNets for NLP.

5. Natural language generation – Coreference resolution – Constituency parsing.

References

1. Dan Jurafsky and James H. Martin: Speech and Language Processing, 3e, Prentice Hall, 2019.

2. Jacob Eisenstein: Natural Language Processing, The MIT Press, 2019.
3. Yoav Goldberg and Graeme Hirst: Neural Network Models for Natural Language Processing, Morgan and Claypool Life Sciences, 2017.
4. Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep Learning, The MIT Press, 2017.
5. Charu C Aggarwal: Neural Networks and Deep Learning, Springer, 2018.
6. Eugene Charniak: Introduction to Deep Learning, The MIT Press, 2019.
7. Natural Language Processing in Deep Learning by Stanford (Coursera)
8. Deep Learning Specialization by deeplearning.ai
9. Applied AI with Deep Learning by IBM (Coursera)
10. Introduction to pyTorch and Machine Learning (Udemy)
11. Practical Deep Learning with pyTorch (Udemy)

23-813-0809: Image and Video Coding

Core/Elective: **Elective** Semester: **8** Credits: **4**

Course Description

The aim of this course is to give a rigorous introduction into the fundamental concepts of data compression with strong emphasis on the mathematical techniques and its applications to image and video coding. The main objectives of the course are to understand how digital data can be compressed using either lossless or lossy techniques, to provide a strong mathematical background in the field of coding theory, to expose the students to the standard compression techniques used in various coding standards and to expose the students to the latest image and video coding standards.

Course Outcomes (CO)

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

CO1: Understand the mathematical preliminaries of Lossless compression techniques.
CO2: Analyse the significance of uniquely decodable codes and prefix codes.
CO3: Evaluate the performance of Huffman code and its variants.
CO4: Evaluate the performance of Arithmetic and Integer Arithmetic coding.
CO5: Understand the fundamentals of dictionary and prediction based compression techniques.
CO6: Evaluate the performance of various dictionary-based compression methods.
CO7: Evaluate the performance of various context-based predictive compression methods.
CO8: Understand the mathematical preliminaries of Lossy compression techniques.
CO9: Evaluate the different quantization techniques.
CO10: Understand the basic concepts of Differential and Transform based coding.
CO11: Compare and contrast different content-dependent video coding techniques.
CO12: Compare and contrast different image and video compression standards.
CO13: Develop an image or video compression technique.

Course Content

1. Introduction: Compression Techniques - Modeling and Coding. Mathematical Preliminaries for Lossless compression: Information Theory – Models - Coding: Uniquely decodable codes - Prefix codes - Kraft-McMillan Inequality. Huffman Coding: Minimum Variance Huffman Codes - Length of Huffman Codes - Adaptive Image Compression Standards: JPEG - JPEG 2000 - JPEG XR - JPEG-LS - JPEG XT - JPEG Pleno. Video Compression Standards: MPEG-4 - H.263 - H.264/AVC - H.265/HEVC - AVS China - Dirac.ffman Coding - Golomb codes - Rice codes - Tunstall codes. Arithmetic Coding: Integer Arithmetic Coding.

2. Dictionary Techniques: Static Dictionary - Digram coding - Adaptive Dictionary - LZ77 - LZ78 - LZW. Context-based Compression: Prediction with partial match - Burrows-Wheeler Transform – CALIC - Run-Length Coding – JBIG – JBIG2.

3. Mathematical Preliminaries for Lossy Coding: Distortion Criteria - Rate Distortion Theory. Scalar Quantization: Quantization problem - Uniform Quantizer - Lloyd-Max Quantizer - Adaptive Quantization - Non-uniform Quantization - Entropy-Coded Quantization. Vector Quantization: LBG Algorithm - Tree Structured and Structured Vector Quantizers. Differential Coding: Basic algorithm – DPCM. Transform Coding.

4. Content dependent video coding: Temporal prediction and Transform coding - Two dimensional shape coding - Joint shape and texture coding - Region based and object based video coding - Knowledge based video coding - Semantic video coding - Layered coding system - Scalable video coding.

5. Image Compression Standards: JPEG - JPEG 2000 - JPEG XR - JPEG-LS - JPEG XT - JPEG Pleno. Video Compression Standards: MPEG-4 - H.263 - H.264/AVC - H.265/HEVC - AVS China - Dirac.

References

1. Khalid Sayood, Introduction to Data Compression, 4e, Morgan Kaufmann Publishers, 2012.
2. David Salomon, Data Compression – The Complete Reference, 4e, Springer, 2006.
3. Alistair Moffat and Andrew Turpin, Compression and Coding Algorithms, 1e, Kluwer Academic Publishers, 2002.
4. Vasudev Bhaskaran and Konstantinos Konstantinides, Image and Video Compression Standards, 2e, Kluwer Academic Publishers, 2003.
5. Mark Nelson and Jean-Loup Gailly, The Data Compression Book, 2e, John Wiley & Sons, 1995.
6. John Miano, Compressed Image File Formats, 1e, Addison Wesley Professional, 1999.
7. Peter Wayner, Compression Algorithms for Real Programmers, Morgan Kaufmann, 1e, 1999.
8. Yao Wang, Jorn Ostermann and Ya-Qin Zhang, Video Processing and Communications, Pearson, 1e, 2001.
9. Alan C. Bovik, The Essential Guide to Video Processing, Academic Press, 2e, 2009.
10. A. Murat Tekalp, Digital Video Processing, Prentice Hall, 2e, 2015.

23-813-0810: Functional Programming

Core/Elective: **Elective** Semester: **8** Credits: **4**

Course Description

As big data and multiple cores become ubiquitous, functional programming has become relevant as never before. The latest standards for popular programming languages like C++ and Java have included support for a large number of functional programming features. This course aims to provide a thorough introduction to functional programming. It covers both the theoretical underpinnings and practical, programming aspects.

Course Outcomes (CO)

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

CO1: Explain and appreciate the functional programming paradigm.
CO2: Identify the various methods in functional programming as different from imperative programming.
CO3: Analyze the proofs of correctness of functional programming codes.
CO4: Develop programming skills on any one frontline functional languages (e.g. Haskell, Clojure, Kotlin etc.)
CO5: Appreciate the need for imperative constructs and explain efficient methods and workarounds for the cases in functional programming languages.
CO6: Measure and appraise the recent adaptations of functional concepts into non-functional languages.

Course Content

1. Introduction to Functional Programming – Motivation – Defining features of the functional Paradigm – First Class Functions – Referential Transparency – Introduction to Haskell – Data Types and Pattern Matching– Laziness – Program Correctness.

2. Lambda Calculus – Alpha, beta conversions – Normal forms – Applicative order – Reductions - Church Rosser Theorems – Y combinator – Recursion – Proofs of correctness.

3. Classes for Numbers – Lists in Haskell – Basic List operations – Higher order list functions – List comprehension – Strings and Tuples – User defined data types: lists, queues, trees.

4. Proving correctness of programs – Induction – Proofs using higher order functions – Infinite Lists – Lazy Evaluation – Efficiency – Controlling Space and Time complexity – Polymorphism - Conditional Polymorphism – Type classes.

5. Programming imperatively in Haskell – The IO Monad – Why Monads are Necessary – The State Monad– ST Monad – Mutable and Immutable Arrays – Parsing using Monads –

Applications – Fault-tolerant systems – Financial analysis – Comparison to other functional languages.

References

1. Richard Bird, Thinking Functionally with Haskell, 1e, Cambridge University Press, 2014.
2. Graham Hutton, Programming in Haskell, 1e, Cambridge University Press, 2007.
3. KeesDoets and Jan van Eijck, The Haskell Road to Logic, Maths and Programming, 2e, College Publications, 2004.
4. Greg Michaelson, an Introduction to Functional Programming through Lambda Calculus, 1e, Dover Publications, 2011.
5. Chris Okasaki, Purely Functional Data Structures, 1e, Cambridge University Press, 1999.

23-813-0811: Information Retrieval and Web Search

Core/Elective: **Elective** Semester: **8** Credits: **4**

Course Description

A coherent treatment of classical and web based information retrieval that includes web search, text classification, text clustering, gathering, indexing and searching documents and methods of evaluating systems.

Course Outcomes (CO)

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

CO1: Understand advanced techniques for text-based information retrieval .
CO2: Understand Boolean and vector space retrieval models.
CO3: Evaluate various text classification techniques.
CO4: Understand Web search characteristics, web crawling and link analysis.
CO5: Build working systems that assist users in finding useful information on the Web.

Course Content

1. Taxonomy of IR Models – Classic models- Set theoretic model- Algebraic models-Probabilistic modelStructured text retrieval models- Models for browsing- Retrieval evaluations-Reference collections.

2. Query languages-query operations-text and multimedia languages-Text operations-document preprocessing- matrix decompositions and latent semantic indexing-text compression –indexing and searching-inverted files-suffix trees- Boolean queries-sequential searching-pattern matching.

3. Text Classification, and Naïve bayes-vector space classification-support vector machines and machine learning on documents-flat clustering –hierarchical clustering.

4. Web search basics-web characteristics-index size and estimation- near duplicates and shingling-web crawling-distributing indexes- connectivity servers-link analysis-web as a graph-PageRank-Hubs and authorities- question answering.

5. Online IR systems- online public access catalogs-digital libraries-architectural issues - document models - representations and access- protocols.

References

1. Ricardo Baezce Yates and BerthierRibeiro-Neto, Modern Information Retrieval: The Concepts and Technology behind Search, 3e, ACM Press, 2017.

2. Christopher D. Manning, PrabhakarRaghavan and HinrichSchütze, Introduction to Information Retrieval, 1e, Cambridge University Press, 2008.
3. Bruce Croft, Donald Metzler and Trevor Strohman, Search Engines: Information Retrieval in Practice, 1e, AW, 2009.

23-813-0812: Human Computer Interaction

Core/Elective: **Elective** Semester: **8** Credits: **4**

Course Description

Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and the major phenomena surrounding them. It is often regarded as the intersection of Computer Science and behavioural science. HCI is also sometimes referred to as man–machine interaction (MMI) or computer–human interaction (CHI).

Course Outcomes (CO)

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

CO1: Explain the capabilities of both humans and computers from the viewpoint of human information processing.
CO2: Describe typical human–computer interaction (HCI) models and styles, as well as various historic HCI paradigms.
CO3: Apply an interactive design process and universal design principles to designing HCI systems.
CO4: Describe and use HCI design principles, standards and guidelines.
CO5: Analyze and identify user models, user support, socio-organizational issues, and stakeholder requirements of HCI systems.

Course Content

1. Overview of HCI – Mental models – Cognitive architecture – task loading and stress in HCI – Human error identification.

2.Input technologies – sensor and recognition based input – visual displays – Haptic interfaces – Non speech auditory output – network based interactions.

3.Designing human computer interaction – Visual design principles – intercultural user interface designs – Conversational speech interface – multimodal interface – adaptive interfaces and agents – Tangible user interfaces – Information visualization – Human centered designs of DSS – Online communities – Visual environment.

4.Domain specific design – HCI in healthcare – games – older adults – kids – Physical disabilities – Perpetual Impairments – Deaf and Hard of Learning users.
5.Developments process – requirement specification – User experiences and HCI – Usability Engineering life cycle – Task analysis – prototyping tools and techniques – scenario based design – Participatory design – Testing and evaluation – Usability testing – Inspection based evaluation – Model based evaluation.

References

1.Andrew sears, Julie A Jacko and Lawrence, The Human Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, 1e, Erlbaum Associates, 2008.
2.Alan Dix, Janet Finlay, Gregory D Abowd and Russell Beale, Human - Computer Interaction, 3e, Pearson, 2012.
3.Helen Sharp, Yvanno Rogers and Jenny Preece, Interaction Design: Beyond human Computer Interaction, 1e, John Wiley, 2011.
4.Jan Noyes and Chris Baber, User-Centred Design of Systems, 1e, Springer, 2013.

23-813-0813: Cyber Physical Systems

Core/Elective: **Elective** Semester: **8** Credits: **4**

Course Description

Cyber-Physical Systems (CPS) is a new frontier for computer systems that is transforming the way people interact with engineered systems. CPS applications include systems such as aircraft, automotive, medical devices, process control, and critical infrastructure. Unlike the traditional computer systems, the interplay between the cyber and the physical systems in CPS brings significant challenges in the modeling, design, analysis and verification of such systems. The complex, interdisciplinary nature of CPS requires a unique approach for the education of CPS. This course introduces Modeling formalism of Cyber-Physical Systems (CPS), Modeling of physical and cyber systems, and software synthesis from these modeling formalisms.

Course Outcomes (CO)

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

CO1: Categorize the essential modeling formalism of Cyber-Physical Systems (CPS).
CO2: Analyze the functional behavior of CPS based on standard modeling formalism.
CO3: Improve specific software CPS using existing synthesis tools.
CO4: Contrast CPS requirements based on operating system and hardware architecture constraints.
CO5: Analyze and verify the correctness of CPS implementations against system requirements and timing constraints.

Course Content

1. Introduction to Cyber Physical System: Cyber physical system: Definition Applications, Design Process for Cyber Physical System: Modeling, Design, Analysis: Modelling continuous dynamics, Newtonian Mechanics, Actor models, Properties that actors and the systems: Causal Systems, Memoryless Systems, Linearity and Time Invariance, Stability. Feedback control

2. Modeling Discrete Systems :Discrete Systems ,State, Finite-State Machines: Transitions, The occurrence of reaction, Update functions, Determinacy and Receptiveness, Extended State Machines, Nondeterministic Finite State Machines , Behaviors and Traces

3. Hybrid Systems: Actor Model for State Machines, Continuous Inputs, State Refinements, Classes of Hybrid Systems: Timed Automata, Higher-Order Dynamics, Supervisory control

4. Composition of State Machines: Concurrent Composition: Side-by-Side Synchronous Composition, Side-by-Side Asynchronous Composition, Shared Variables, Cascade Composition, General Composition, Hierarchical state machines

5. Concurrent Models of Computation : Structure of Models, Synchronous-Reactive Models: Feedback Models, Well-Formed and ill-Formed Models, Constructing a Fixed Point, Dataflow Models of Computation: Dataflow Principles, Synchronous Dataflow ,Dynamic Dataflow, Structured Dataflow, Process Networks, Timed Models of Computation: Time-Triggered Models, Discrete Event Systems, Continuous-Time Systems

References:

1. Edward Ashford Lee, Sanjit Arunkumar Seshia, Introduction to Embedded Systems - A Cyber-Physical Systems Approach, 2e, MIT Press, 2017
2. Walid M. Taha · Abd-Elhamid M. Taha Johan Thunberg, Cyber-Physical Systems: A Model-Based Approach, 1e, OpenAccess, https://doi.org/10.1007/978-3-030-36071-9, Springer
3. Rajeev Alur , Principles of Cyber-Physical Systems, 1e, MIT Press, 2015
4. Raj Rajkumar, Dionisio de Niz, Mark Klein, Cyber-Physical Systems, 1e, AW Professional, 2017
5. Peter Marwedel, Embedded System Design: Embedded Systems Foundations of CyberPhysical Systems, and the Internet of Things, 3e, Springer, 2017

Online courses:

Coursera: https://www.coursera.org/learn/cyber-physical-systems-1

23-813-0901: Research Methodology

Core/Elective: **Core** Semester: **9** Credits: **4**

Course Description

This is a research methodology course that focuses on developing researching and writing skills in the Computer Science & Engineering Domain. As software development requires a multidisciplinary approach, many of the concepts are borrowed from social science, Psychology, Statistics, and other domains. Here we investigate the empirical research methods for their applicability and suitability to a research problem. As each of them comes with their strengths and weaknesses, perhaps a feasible, judicial mix of such methods should provide a greater insight and understanding in order to derive useful contributions.

Course Outcomes

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

CO1: Understand research and research methods in Computer Science.
CO2: Assess the gap by acquiring knowledge about the previous works, and its interpretation and application.
CO3: Demonstrates the design of the proposed methodology and its merits.
CO4: Organize the research content with proper structure and sequencing.
CO5: Demonstrate the academic discussion skills to emphasize, argue with clarity of purpose using evidence for the claims

Course Content

1.Introduction: Meaning, Objectives, Types of Research, Research Approaches, Significance of Research, Research Methods versus Methodology, Research and Scientific Method, Research Process, Criteria of Good Research. Defining the research problem, Selecting the Problem, Necessity of Defining the Problem, Technique Involved in Defining a Problem. Research design: Meaning, Need for Research Design, Features of a Good Design, Important Concepts relating to Research Design, Different Research Designs.

2. Data Collection: Introduction, Experiments and surveys, Collection of Primary and Secondary Data, selection of appropriate method for data collection. Data Preparation process, Some problems in preparation process, Missing values and Outliers, types of Analysis, Statistics in research. Testing of hypothesis: Hypothesis, Testing the Hypotheses, Test Statistic and Critical region, critical value and Decision Rule, Procedure for Hypothesis Testing, Hypothesis Testing for – Means, Proportions, variance, difference of two mean, difference of two proportions, difference of two variances, P-Value approach, power of test, Limitations of the Tests of Hypotheses. ChiSquare Tests.

3. Publication Ethics: definition, introduction and importance.Publication misconduct, Predatory publishers and journals. Open-access publishing - Open access publications and initiatives. Software tools: Use of plagiarism software like Turnitin, Urkund and other open source software tools.

4. Writing research papers, purpose, nature and evaluation, content and format, Research Presentations, The Art of Scientific and Technical Writing.Types of Report: research papers, thesis, Research Project Reports, Precautions for writing research reports, Pictures and Graphs, Oral presentation. Technical writing using LaTeX, MikTeX, Creating reports and articles, Text environment, Math environment, Figures, Tables, BibTeX, Camera Ready Preparation.

5. Database and Research Metrics: Indexing and citation databases, Web of Science, Scopus etc. Research Metrics: Journal level and author level metrics, Impact Factor, JCR, SNIP, SJR, Cite Score, Eigen factor, Article Influence Score, h-index, g-index, i10-index, e-index, hi-index, h5-index, h5-median, altmetrics etc.

References

1. C.R. Kothari and Gaurav Garg, Research Methodology : Methods And Techniques, 4e, New Age International Publishers, 2019.

2. Bhanwar Lal Garg, Renu Kavdia, Sulochana Agarwal and Umesh Kumar Agarwal: Introduction To Research Methodology, 1e, RBSA Publishers, 2015.

3. MLA Handbook for Writers of Research Papers, 7e, Modern Language Association of America, 2009.

4. Publication Manual of the American Psychological Association, 6e, American Psychological Association, 2009.

5. Jude Carroll, A Handbook for Deterring Plagiarism in Higher Education, 2e, OCSLD, 2007.

6. Robert A. Day and Barbara Gastel, How to Write and Publish a Scientific Paper, 7e, Greenwood Press, 2011.

7. T.W. Anderson, An Introduction to Multivariate Statistical Analysis, 3e, Wiley, 2009.

23-813-0902: Project & Viva Voce

Core/Elective: **Core** Semester: **9** Credits: **12**

Course Description

The project work spans two semesters. Through the project work, the student has to exhibit the knowledge in terms of engineering or technological innovation or research ability to solve the contemporary problem. On completion of the first part of the work, the student shall submit an interim project report. The qualitative and quantitative results of the work will be evaluated through a viva voce exam.

Course Outcomes (CO)

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

CO1: Demonstrates in depth knowledge and thoughtful application through the detailed analysis of the problem chosen for the study.
CO2: Assess the gap by acquiring knowledge about the previous works, and its interpretation and application.
CO3: Demonstrates the design of the proposed methodology and its merits.
CO4: Organize the interim project content with proper structure and sequencing.
CO5: Demonstrate the academic discussion skills to emphasize, argue with clarity of purpose using evidence for the claims

23-813-0903: Elective V

Core/Elective: **Elective** Semester: **9** Credits: **4**

Course Description

A credit-based MOOC course of minimum 12 weeks duration or three non-credit based MOOC courses of 4-weeks duration from SWAYAM/NPTEL/any other platforms approved by the University.

23-813-1001: Project & Viva Voce

Core/Elective: **Core** Semester: **10** Credits: **20**

Course Description

The project work spans two semesters. Through the project work, the student has to exhibit the knowledge in terms of engineering or technological innovation or research ability to solve the contemporary problem. On completion of the work, the student shall submit a final project report. The qualitative and quantitative results of the work will be evaluated through a viva-voce exam.

Course Outcomes (CO)

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

CO1: Demonstrates in depth knowledge and thoughtful application through the detailed analysis of the problem chosen for the study.
CO2: Assess the gap by acquiring knowledge about the previous works, and its interpretation and application.
CO3: Demonstrates the design of the proposed methodology and its merits.
CO4: Organize the final project report content with proper structure and sequencing.
CO5: Demonstrate the academic discussion skills to emphasize, argue with clarity of purpose using evidence for the claims.
CO6: Show ability to evaluate and reflect on critical questions.