Master of Technology

in

Computer Science & Engineering

Course Structure & Syllabus

Department of Computer Science & Engineering

National Institute of Technology Hamirpur

Hamirpur (HP) – 177005, India

1st Semester							
Sr. No.	Subject Code	Subject Name	Teaching Schedule			Hours/Week	Credit
			L	T	P		
1	CS-611	Advance Topics in Software Engineering	4	0	0	4	4
2	CS-612	Theoretical Computer Science	4	0	0	4	4
3	CS-613	Computer Systems	4	0	0	4	4
4	CS-7MN	Programme Elective-I	4	0	0	4	4
5	CS-7MN	Programme Elective-II	4	0	0	4	4
6	CS-614	Computational Lab–I	1	0	2	3	2
Total			**21**	**0**	**2**	**23**	**22**

2nd Semester							
Sr. No.	Subject Code	Subject Name	Teaching Schedule			Hours/Week	Credit
			L	T	P		
1	CS-621	Data Structures & Algorithms	4	0	0	4	4
2	CS-622	Network & Distributed Systems	4	0	0	4	4
3	CS-623	Advanced Database Management System	4	0	0	4	4
4	CS-7MN	Programme Elective-III	4	0	0	4	4
5	CS-70X	Institute Elective	4	0	0	4	4
6	CS-624	Computational Lab–II	1	0	2	3	2
Total			**21**	**0**	**2**	**23**	**22**

3rd Semester							
Sr. No.	Subject Code	Subject Name	Teaching Schedule			Hours/Week	Credit
			L	T	P		
1	CS-798	M.Tech. Dissertation	-	-	-	-	18
Total			**-**	**-**	**-**	**-**	**18**

4th Semester							
Sr. No.	Subject Code	Subject Name	Teaching Schedule			Hours/Week	Credit
			L	T	P		
1	CS-799	M.Tech. Dissertation	-	-	-	-	18
Total			**-**	**-**	**-**	**-**	**18**

Semester	1st	2nd	3rd	4th	Total
Credits	22	22	18	18	80

Annexure
List of Programme Electives

CS-711 Mathematics for Computer Science
CS-712 Artificial Intelligence
CS-713 Soft Computing
CS-714 Speech and Natural Language Processing
CS-715 Bioinformatics
CS-716 Cryptography and Computer Security
CS-717 Parallel Algorithms
CS-718 Information Theory and Coding
CS-719 Game Theory
CS-720 Combinatorial Optimization
CS-721 Intrusion Detection System
CS-722 Computer Vision
CS-723 Fault Tolerant Computing
CS-724 Biometric Systems and Security
CS-725 Cluster and Grid Computing
CS-726 Cloud Computing
CS-727 Blockchain Technology
CS-728 Advance Mobile Communication
CS-729 Networked Wireless System
CS-730 Quantum Computing
CS-731 Machine Learning
CS-732 Advances in Machine Learning
CS-733 Computer Graphics
CS-734 Digital Image Processing
CS-735 Data Mining
CS-736 Architecture of Large Systems
CS-737 Embedded Systems
CS-738 Graph Mining and Algorithms
CS-739 Internet of Things
CS-740 Advance Computer Networks

In addition to these electives, any other core/elective of M.Tech. Artificial Intelligence may also be floated as elective for M.Tech. Computer Science & Engineering.

List of Institute Electives

CS-701: Artificial Intelligence
CS-702: Machine Learning for Engineers
CS-703: Data Structures & Algorithms
CS-704: Computer Networks
CS-705: Programming for Problem Solving

Course Name: **Advance Topics in Software Engineering**

Course Code: **CS-611**

Course Type: **Programme Core**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To impart knowledge about the principles of software engineering.
- To enable the students to understand the developmental models.
- To introduce the concepts relevant to software development and project management.

Course Content

Introduction to Software Engineering: Evolution, Characteristics of Software, Principles, Software Projects being undertaken, Changes in software developmental practices. Software Process models: Waterfall, Prototype, Iterative, Evolutionary, Spiral, Unified Process Model, Comparison of different models and selection of appropriate model for project. Software project management: Project planning, Metrics and measurement, Project estimation techniques, Staffing and scheduling, Organization of teams, Risk management, Configuration management, Requirements Analysis and specifications, Software Design, User Interface Design, Coding and Testing, Verification and Validation, Software reliability and quality management, Software Maintenance and Reuse. CASE tools and technology: UML, Extreme Programming, Aspect Oriented Programming, Secure software engineering principles. Emerging Trends: Service oriented Architecture, Software as a service, recent trends in software development across different platforms like cloud etc.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Effectively apply software engineering practice over the entire system lifecycle. This includes requirements engineering, analysis, prototyping, design, implementation, testing, maintenance activities and management of risks involved in software and embedded systems.

CO2: Know classical and evolving software engineering methods, can select and tailor appropriate methods for projects, and can apply them as both team members and managers to achieve project goals.

CO3: Prepare and publish the necessary documents required throughout the project lifecycle.

Text Books:

1. Software Engineering: A Practitioner's Approach by Roger Pressman, McGraw-Hill Publishing.
2. Software Engineering by Sommerville, Pearson Education.
3. Fundamentals of Software Engineering by Rajib Mall, PHI Learning Pvt. Ltd.

Reference Books:

1. An Integrated Approach to Software Engineering by Pankaj Jalote, Narosa Publishing House.
2. Software Engineering by S. L. Pfleeger, MacMillan Publishing Company.

Course Name: **Theoretical Computer Science**

Course Code: **CS-612**

Course Type: **Programme Core**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To impart knowledge about disjunctive normal form and determine their validity.
- To introduce the fundamental concepts of common language to formal logic, formal languages.
- To enable the students to understand the application of Turing Machine and push down automata.

Course Content

Introduction to Preliminaries, Set Theory and Functions: Basic concepts of discrete mathematics and related problems, tautologies and contradiction, laws of equivalence, rules of substitution and transitivity, normal forms.
Venn Diagrams, set operations, Relations and ordering, Types of relations, properties of a relation, Functions: identity and inverse, related results.
Formal Languages Turing Machines and Pushdown Automata: Deterministic and non-deterministic FSMS, Equivalence of DFA and NDFA, Mealy and Moore machine, Chomsky hierarchy, Regular grammars, context free & context sensitive grammars, Chomsky normal forms, binary operations on languages. Simplification of CFG, Grayback Normal form, TM model, representation and languages acceptability of TM Design of TM, Universal TM and Other modification, composite and iterated TM, Pushdown automata, Acceptance by PDA.
Computability and Undecidability: Basic concepts, primitive and partial recursive function, Recursive function, Decidability, Kleene's theorem. Properties of recursive & recursively enumerable languages, Universal Turing machine and undecidable problem, Rice"s theorem, undecidable problems.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Interpret statements presented in disjunctive normal form and determine their validity.
CO2: Reformulate statements from common language to formal logic.
CO3: Demonstrate understating of context free grammars and ability to construct grammars for specific tasks.
CO4: Demonstrate understanding of Turing Machine and push down automata.

Text Books:

1. Discrete Mathematical structures with applications to Computer Science by J. P. Tremblay and R Manohar, McGraw Hill.
2. Elements of Discrete Mathematics by C.L. Liu, McGraw Hill.
3. Introduction to Automata theory, Languages and Computation by John E. Hopcroft and Jeffery Ullman, Narosa Publishers.
4. Introductory Theory of Computer Science by E.V. Krishnamurthy, East West Press.

Reference Books:

1. Discrete Mathematics by John A. Dossey, Albert D. Otto, Lawrence E. Spence and Vanden Eynden, Charles Addison-Wesley.
2. Introduction to Languages and the Theory of Computation by John C. Martin, Tata McGraw Hill.

Course Name: **Computer Systems**

Course Code: **CS-613**

Course Type: **Programme Core**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To impart knowledge about the various approaches of Operating System.
- To introduce the fundamental concepts relevant to understand the concepts of Computer Architecture.
- To enable the students to understand the basics structure of computer hardware and its operating systems.
- To impart knowledge about the organization of any computing system.
- To introduce the fundamental concepts relevant to design instruction set architectures and develop their micro architectures.

Course Content

Evolution of Operating Systems and its responsibilities: Evolution of operating systems, Types of operating systems. The process concept, system programmer's view of processes, operating system's views of processes, operating system services for process management. Scheduling concepts, scheduling algorithms, algorithm evaluation, Memory Management, memory allocation, demand paging, page replacement, segmentation. File Systems: A Simple file system, General model of a file system, Symbolic file system, Access control verification, Logical file system, Concurrent Programming and Deadlocks: Critical regions, Conditional critical regions, Deadlocks and its related issues.

Computer System Architecture: Classification of computers, Micro-operation, Instruction Set Architecture, Addressing Modes, Operation instruction set, Instruction set format, Processor Design: Arithmetic and logic unit, Stack organization, CPU Architecture types, Accumulator Based- Register, Stack-Memory, Register, Detailed data path of a typical register based CPU, Fetch, Decode, and Execute Cycle. Addition and Subtraction, Multiplication Algorithms. Control Design: Microprogrammed and Hard-wired control options, Hard-wired design methods, State table method, Multiplier control, CPU control unit. Microprogrammed, Basic concepts, control Memory, Address Sequencing. I/O Organization & Memory Hierarchy: Programmed, Interrupt driven & Direct Memory Access, Synchronous & synchronous data transfer, The need for Memory Hierarchy, locality of reference principle, Memory Hierarchy, cache, main and secondary, Memory parameters, access cycle time, cost per unit.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understand and analyze the concepts of operating system and its management

CO2: Illustrate the scheduling of processes for a given problem instance

CO3: Appreciate macro-organization of any computing system

CO4: Design instruction set architectures and develop their micro architectures

CO5: Understand various digital arithmetic algorithms.

Text Books:

1. Operating System Concepts by J.L. Peterson and A. Silberchatz, Addison Wesley.
2. Computer Organization and Architecture, Designing for Performance by William Stallings, Pearson Education.
3. Computer System Architecture by M. Morris Mano, Prentice Hall of India.

Reference Books:

1. Operating systems by W. Stallings, Prentice Hall.
2. Structured Computer Organization by Andrew S. Tanenbaum and Todd Austin, Prentice Hall of India

Course Name: **Computational Lab–I**

Course Code: **CS-614**

Course Type: **Programme Core (Lab)**

Contact Hours/Week: **1 L & 2P** Course Credits: **02**

Course Objectives

- To provide skills for designing and analyzing algorithms.
- To enable students to work on various simulators.
- To provide skills to work towards solution of real-life problems.

List of Experiments

1. Installation and working on various simulators viz. ETHEREAL, OMNET++, NS2,NS3, MATLAB, etc.
2. Installation and working on various AI tools viz. Python, scipy, R tool, NLTK, MATLAB, etc.
3. Implementation of distributed file systems with Big Data.
4. Implementation of searching techniques over big data.
5. Implementation of various encryption and decryption algorithms and comparing their performance.
6. Content based searching in image data.
7. Application development for any management information system.
8. Hands-on experience — develop, scalable, deploy websites and apps with free Cloud Simulator.
9. Creation and usage of open platform for developing & deployment applications Web Services
10. Working on parallel algorithms.
11. Creation of GUI.
12. Creating documents and presentations in LaTex.

Note: *The concerned Course Coordinator will prepare the actual list of experiments/problems at the start of semester based on above generic list.*

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Elicit, analyze and specify software requirements.

CO2: Simulate given problem scenario and analyze its performance.

CO3: Develop programming solutions for given problem scenario.

Course Name: **Data Structures & Algorithms**

Course Code: **CS-621**

Course Type: **Programme Core**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To impart knowledge about basic data structures
- To impart knowledge about the various approaches to design an algorithm.
- To introduce the fundamental concepts relevant to understand the concepts of time and space complexity, worst case, average case and best-case complexities.
- To enable the students to understand the basics of algorithms.

Course Content

Introductions to Data Structures and Algorithms: Algorithm complexity, concept of algorithmic efficiency, run time analysis of algorithms, Asymptotic Notations, Arrays, Linked List, Stack, Queues, Minimum and maximum priority queue, Trees and Graphs (Memory Representation of each and a few basic operations)

Algorithm Design Approaches: Divide and Conquer approach, examples of some sorting techniques like Merge and Quick Sort; Greedy Algorithms; Graph Algorithms: Representation of graphs, BFS, DFS, single source shortest path, all pair shortest path; Dynamic programming: Overview, difference between dynamic programming and divide and conquer, Traveling salesman Problem, longest Common sequence, 0/1 knapsack., Backtracking: 8-Queen Problem, Sum of subsets, graph coloring, Hamiltonian cycles. Computational Complexity: Complexity measures, Polynomial vs non-polynomial time complexity; NP-hard and NP-complete classes and examples.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understand asymptotic notations to analyze the performance of algorithms.

CO2: Understand and apply various problem-solving techniques such as divide and conquer, greedy algorithm, dynamic programming, etc.

CO3: Solve given problem by selecting the appropriate algorithm design technique and justify the selection.

CO4: Know the concepts of P, NP, NP-hard and NP-complete problems.

Text Books:

1. Fundamentals of Computer Algorithms by E. Horowitz and S. Sahni, Galgotia.

2. Introduction to Algorithms by T.H. Cormen, C.E.Leiserson, R.L. Rivest, MIT Press, Cambridge.

3. Data Structure Using C by Aaron M. Tenenbaum, Y. Langsam, M. Augensten, Pearson

Reference Books:

1. The Design and Analysis of Computer Algorithms by A.V. Aho, J.E. Hopcroft and J.D. Ullman, Addison Wesley.

2. Algorithm Design by J. Kleinberg and É. Tardos, Addison-Wesley.

Course Name: **Network & Distributed Systems**

Course Code: **CS-622**

Course Type: **Programme Core**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To impart knowledge about the network models and architectures.
- To introduce the fundamental concepts relevant to performance of various routing protocols and design of new routing protocol.
- To impart knowledge about the concepts of various architectural and fundamental models in distributed systems.
- To introduce the fundamental concepts relevant to client server and peer to peer interactions between clients and sever.

Course Content

Computer Networks: Goals and Applications of Networks, LAN, WAN, MAN, Network software: Protocol hierarchies, design issues of layers, Interfaces and services. Reference Model: The OSI reference model, TCP/IP reference model, Example networks. Introduction and basics of Physical Layer, Data Link Layer, Medium access sublayer, network layer, session layer, presentation, and application layer.

Introduction to Wireless Networks, basic concepts, and issues of wireless networks like hidden layer problem etc. wireless technologies like GSM, CDMA, and evolution from 1G to 6G. Type of wireless networks like Ad-hoc networks, MANETS, sensor networks etc.

Introduction to Distributed systems: Characterization of Distributed System, Examples of distributed systems, Resource sharing, challenges. Various Architectural Models, Clock Synchronization, logical and vector clocks, Clock Synchronization Algorithms, Distributed Mutual Exclusion, Distributed Deadlock Detection, Agreement protocols, Distributed resource management, Distributed shared memory-Architecture, Load distribution Scheduling.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understand network models and architectures

CO2: Identify the pros and cons of choosing a suitable MAC layer protocol

CO3: Analyze the performance of various routing protocols and design of new routing protocol

CO4: Understand concepts of various architectural and fundamental models in distributed systems

CO5: Implement client server and peer to peer interactions between clients and sever

CO6: Analyze time synchronization protocols and select a suitable time synchronization protocol

Text Books:

1. Computer Networks by A.S. Tanenbaum, Prentice Hall of India.
2. Computer Networking: A Top-Down Approach Featuring the Internet by J. Kurose and K.W. Ross, Addison-Wesley.
3. Distributed Systems: Concepts and Design by G. Coulouris, J. Dollimore, and T. Kindberg, Pearson Education.
4. Distributed Systems: Principles and Paradigms by A.Tanenbaum and Maarten van Steen, Prentice Hall of India.

Reference Books:

1. Data and Computer Communication by W. Stallings, Prentice Hall of India.
2. Advanced Concepts in Operating Systems by M. Singhal and N. Shivaratri,Tata McGraw Hill.

Course Name: **Advanced Database Management System**

Course Code: **CS-623**

Course Type: **Programme Core**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- List and explain the fundamental concepts of a database system.
- Utilize a wide range of features available in a DBMS package.
- Analyze database requirements and determine the entities involved in the system and their relationship to one another.
- Develop the logical design of the database using data modeling concepts such as entity-relationship diagrams.
- Create a relational database using a relational database package.
- Manipulate a database using SQL.
- Assess the quality and ease of use of data modeling and diagramming tools.

Course Content

Database Management systems concepts, Conceptual Database Design, Logical Database Design Physical Database Design, Query Processing, Transaction processing, Crash recovery, Concurrency control, Distributed Database, client/server database, Integrity security and repositories. Emerging Database trends, Design and database administration skills based on near-real life applications.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Learn the basic concepts and appreciate the applications of database systems.

CO2: Learn the basics of SQL and construct queries using SQL.

CO3: Be familiar with a commercial relational database system (Oracle) by writing SQL using the system.

CO4: Be familiar with the relational database theory, and be able to write relational algebra expressions for queries.

Text Books:

1. Fundamentals of Database Systems by R. Elmasri and S. Navathe, Benjamin Cummings.

2. Database System Concepts by Silberschatz, Korth and Sudarshan, Tata McGraw Hill.

Reference Books:

1. An introduction to Data Base systems by C. J. Date, Addison Wesley.

2. Client/Server Strategies by Vaakevitch David, Galgotia Publications.

Course Name: **Computational Lab–II**

Course Code: **CS-624**

Course Type: **Programme Core (Lab)**

Contact Hours/Week: **1 L & 2P** Course Credits: **02**

Course Objectives

- To provide exposure of working on hardware software platforms.
- To design and implement solutions for real life problems.

List of Experiments

1. Hands-on experience on Bluetooth, GSM, GPS, Wifi, WAP and Zigbee kits.
2. Simulation of routing protocols for wired and wireless networks.
3. Simulation of MAC protocols for wired and wireless LAN.
4. Hands-on experience with WSN Memsic/Crossbow Classroom Kit for study of various parameters/protocols viz. effect of transmission range, number of nodes, routing, etc.
5. Hand-on experience with IoT Simulators like OpenRemote, ThingsBoard, and Tinkercad etc.
6. Installation and working on various tools viz. Hadoop, Python, Spark, NoSQL, ANACONDA, Tensorflow, Keras, AWS, etc.
7. Implementation/simulation of time synchronization protocols in wired, wireless and heterogeneous environments.
8. Development of mobile applications for collection of users data based on Bluetooth, Wifi and other traffics.
9. Introduction to Version control and build tools like Git.
10. Knowledge extraction & exploratory data analysis (EDA) from given data.
11. Use of machine learning techniques for solving problems related to wired/wireless networks.
12. Comparative evaluation of intrusion detection techniques using various latest tools.

Note: *The concerned Course Coordinator will prepare the actual list of experiments/problems at the start of semester based on above generic list.*

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Simulate/implement given problem scenario and analyze its performance.

CO2: Design solutions for real life problems.

Course Name: **Mathematics for Computer Science**

Course Code: **CS-711**

Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To impart knowledge about the various mathematical concepts used in computer science.
- To introduce the fundamental mathematical concepts relevant to understand the machine learning theory.
- To enable the students to understand the concepts of probability, linear algebra & number theory and their applications in various algorithms.

Course Content

Basics of Linear Algebra: Vector spaces and subspaces, basis and dimensions, linear transformation, four fundamental subspaces. Matrix Theory: Norms and spaces, eigenvalues and eigenvectors, Special Matrices and their properties, least squared and minimum normed solutions. Calculus: Basic concepts of calculus: partial derivatives, gradient, directional derivatives, Jacobian, hessian, convex sets, convex functions and its properties.

Optimization: Unconstrained and Constrained optimization, Numerical optimization techniques for constrained and unconstrained optimization: Newton's method, Steepest descent method, Penalty function method. Probability: Basic concepts of probability: conditional probability, Bayes' theorem, independence, theorem of total probability, expectation and variance, few discrete and continuous distributions, joint distributions and covariance. Number Theory: Divisibility, primes, Euclid's GCD, extended Euclid algorithm, Modular arithmetic, Chinese remainder theorem, Fermat's theorem

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understand the topics from matrix algebra, calculus, optimization, and probability theory

CO2: Understand and apply the techniques related to probability, probability distributions and Matrix theory.

CO3: Understand and apply the concepts of optimization to problem solving.

CO4: Solve a given problem by selecting appropriate mathematical technique.

CO5: Understand the concepts of number theory and devise solutions using them.

Text Books:

1. Mathematics for Machine Learning by Marc Peter, Aldo Faisal, Cheng Soon, Cambridge University Press.
2. Discrete mathematics and its applications by Kenneth Rosen.
3. Linear Algebra and Optimization for Machine Learning by Charu C Aggarwal, Springer

Reference Books:

1. Elementary number theory by David Burton
2. Introduction to combinatorial mathematics by Chung Liu,

Course Name: **Artificial Intelligence**

Course Code: **CS-712**

Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To provide a strong foundation of fundamental concepts in Artificial Intelligence.
- To provide a basic exposition to the goals and methods of Artificial Intelligence.
- To enable the student to apply these techniques in applications which involve perception, reasoning and learning.

Course Content

Introduction: Overview of AI problems, AI problems as NP, NP-Complete and NP Hard problems. Strong and weak, neat and scruffy, symbolic and sub-symbolic, knowledge-based and data-driven AI. Search Strategies: Problem solving by search, Heuristics and informed search, Minmax Search, Alpha-beta pruning. Constraint satisfaction (backtracking and local search methods). A* and AO* algorithms, BFS, DFS algorithms. Knowledge representation and reasoning: propositional and predicate logic, Resolution and theorem proving, Temporal and spatial reasoning. Probabilistic reasoning, Bayes theorem. Totally-ordered and partially-ordered Planning. Goal stack planning, Nonlinear planning. Learning: Learning from example, Learning by advice, Explanation based learning, Learning in problem solving, Classification, Inductive learning, Naive Bayesian Classifier, decision trees. Case study of one or more examples from natural language processing, question answering, speech, expert systems, etc.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understand the various searching techniques and constraint satisfaction problem.

CO2: Apply these techniques in applications which involve perception, reasoning, and learning.

CO3: Acquire the knowledge of real-world Knowledge representation.

CO4: Use different machine learning techniques to design AI machine and enveloping applications for real world problems.

Text Books:

1. Artificial Intelligence by Elaine Rich, Kevin Knight and Shivashankar B Nair, Tata McGraw Hill.
2. Introduction to Artificial Intelligence and Expert Systems by Dan W. Patterson, Pearson Education.
3. Artificial Intelligence: A Modern Approach by S. Russell and P. Norvig, Prentice Hall.

Reference Books:

1. Human Compatible: Artificial Intelligence and the Problem of Control by Stuart Russel, Viking.
2. A First Course in Artificial Intelligence by Deepak Khemani, Tata McGraw Hill.
3. Life 3.0 Being Human in the Age of Artificial Intelligence by Max Tegmark, Penguin UK.

Course Name: **Soft Computing**
Course Code: **CS-713**
Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To impart knowledge about the basic principles, techniques, and applications of soft computing.
- Provide the mathematical background for carrying out the optimization associated with neural network learning.
- Develop the skills to gain basic understanding of the areas of Soft Computing including Artificial Neural Networks, Fuzzy Logic and Genetic Algorithms.

Course Content

Introduction to Neuro fuzzy and Soft Computing: Fuzzy set theory, Fuzzy Rules, Fuzzy Reasoning, Fuzzy inference System, Neural Networks; Radial basis and recurrent neural networks, Hopfield Networks, Comparison of RBF and MLP Network, Running Algorithms, Neuro Fuzzy Modeling, Applications of Soft Computing, Image Processing, Forecasting, XOR Problem traveling salesman problem, Image compression suing MLPs character retrieval using Hopfield networks, Introduction to Genetic Algorithm hybrid systems etc. Recent advances in soft computing applications.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Comprehend the fuzzy logic and the concept of fuzziness involved in various systems and fuzzy set theory.

CO2: Understand the concepts of fuzzy sets, knowledge representation using fuzzy rules, approximate reasoning, fuzzy inference systems, and fuzzy logic.

CO3: Understand the fundamental theory and concepts of neural networks, Identify different neural network architectures, algorithms, applications and their limitations.

CO4: Understand appropriate learning rules for each of the architectures and learns various neural network paradigms and its applications.

Text Books:

1. Learning and Soft Computing by V. Kecman, Pearson.

2. Genetic Algorithms in Search Optimization and Machine Learning by D. E. Goldberg, Addison Wesley.

Reference Books:

1. Neural Network and fuzzy systems by B. Kosko, Prentice Hall of India.

2. Intelligent Hybrid Systems by S. Goonatilake and S. Khebbal, Wiley.

Course Name: **Speech and Natural Language Processing**
Course Code: **CS-714**
Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To develop an in-depth understanding of both the algorithms available for the processing of linguistic information and the underlying computational properties of natural languages.
- To make students focus on the computational properties of natural languages and of the algorithms used to process them.
- To conceive basics of knowledge representation, inference, and relations to the artificial intelligence.

Course Content

Introduction Brief Review of Regular Expressions and Automata, Finite State Transducers, Word level Morphology and Computational Phonology, Basic Text to Speech; Introduction to HMMs and Speech Recognition. Indian language case studies, Part of Speech Tagging; Parsing with CFGs, Probabilistic Parsing. Representation of Meaning, Semantic Analysis, Lexical Semantics, Word Sense, Disambiguation, Discourse understanding, Natural Language Generation, Techniques of Machine Translation, Indian Language case studies.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Get acquainted with natural language processing and learn how to apply basic algorithms in this field.

CO2: Grasp basics of knowledge representation, inference, and relations to the artificial intelligence.

CO3: Improve their programming skills and their knowledge of development tools.

Text Books:

1. Speech and Language Processing by Daniel Jurafsky and James H. Martin, Prentice Hall.
2. NLP: A Paninian Perspective by Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal, Prentice Hall, New Delhi.

Reference Books:

3. Language as a Cognitive Process by T. Winograd, Addison-Wesley.

Course Name: **Bioinformatics**

Course Code: **CS-715**

Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To introduce the fundamental concepts of Bioinformatics.
- To develop skills in designing biological database and retrieving.
- To explore the different paradigms in Data mining, Pattern Recognition, Soft computing.
- To apply appropriate sequence analysis methods for analyzing bio-molecular sequences.

Course Content

Introduction; Databases mapping, sequence, structure, non-redundant, Sequence alignment pair wise and multiple, phylogenetic, Structure prediction methods, homology, threading, abinitio, Sequence analysis class and secondary structure prediction, motifs PROSITE, detecting functional sites in DNA, OR Finder, Computer science perspective pattern recognition, hidden Markov models, Data Mining using Soft Computing Techniques. Brief intro to genomics, Body as a switch, Biostatistics, Computer aided drug design

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understand the basic terminologies used in the field of Bioinformatics.

CO2: Understand the databases related to Bioinformatics and will be able to comprehend data in these databases.

CO3: Perform sequence alignment and analysis.

CO4: Apply computational techniques and prediction algorithms to solve problems related to the domain of Bioinformatics.

Text Books:

1. Bioinformatics by D. Baxevanis and B. F. F. Ouellette, Wiley Interscience.
2. Introduction to bioinformatics by M. Lesk, Oxford University Press.

Reference Books:

1. Computational methods in molecular biology by S. L. Salzberg, D. B. Searls and S.Kasif (eds.), Elsevier.
2. Computer methods for macromolecular sequence analysis by R. F. Doolittle, Academic Press.
3. Guide to human genome computing by M. Bishop, Academic Press.

Course Name: **Cryptography and Computer Security**
Course Code: **CS-716**
Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To understand basics of Cryptography and Network Security
- To learn about how to maintain the Confidentiality, Integrity and Availability of a data.
- Encrypt and decrypt messages using block ciphers, sign and verify messages using well known signature generation and verification algorithms.
- To understand various protocols for network security to protect against the threats in the networks.

Course Content

Introduction: Need and basic goals for computer security, security threats etc. Cryptography, elementary number theory, finite fields, arithmetic and algebraic algorithms, secret key and public key cryptography, Pseudo random bit generators, Block and stream ciphers, Hash functions and message digests, Public key encryption, Probabilistic encryption, Authentication, Digital signatures, Zero knowledge interactive protocols, Elliptic curve cryptosystems, homomorphic encryption, Formal verification, Cryptanalyses, Hard problems. Network Security: problems in network security; kinds of attacks, PKI, key exchange protocols, example protocols such as PGP, Kerberos, IPSEC/VPN, SSL, S/MIME, etc. Protocol vulnerabilities: examples of protocol vulnerabilities such as in TCP/IP, denial of service attacks etc. Tools for network security such as firewalls and intrusion detection systems.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understand the most common type of cryptography algorithms and network security concepts and application.

CO2: Understand security protocols for protecting data on networks.

CO3: Develop an understanding of security policies as well as protocols to implement such policies in the form of message exchanges.

CO4: Understand vulnerability assessments and the weakness of using passwords for authentication.

Text Books:

1. Cryptography, Theory and Practice by Douglas R. Stinson, CRC Press.
2. Cryptography and Network Security: Principles and Practices by W. Stallings, Prentice Hall.
3. Applied cryptography by B. Schneier, John Wiley & Sons.

Reference Books:

1. Handbook of Applied Cryptography by A. Menezes, P. Van Oorschot, S. Vanstone, CRC Press.
2. Network Security by C. Kaufman, R. Perlman and M. Speciner, Prentice Hall.
3. Introduction to Cryptography with coding theory by Wade Trappe and Lawrence C. Washington, Pearson Education.

Course Name: **Parallel Algorithms**
Course Code: **CS-717**
Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To introduce principles and design techniques of parallel algorithms and data structures for various parallel architectures.
- To emphasize on theoretical aspect as well as empirical development of algorithms.

Course Content

Introduction to parallel algorithms: EREW, CREW, CRCW PRAMs and interconnection network models, Need for Parallel Processing, Data and Temporal Parallelism, Models of Computation, RAM and PRAM Model, Shared Memory and Message Passing Models, Processor Organizations, PRAM Algorithms, Analysis of PRAM Algorithms Parallelization of Algorithm: Different Parallel Programming Models, Brent's Theorem, Message passing algorithm, Load balancing and termination detection, programming with shared memory, programming for Distributed shared memory. Basic Parallel Algorithmic Techniques: Divide-and-Conquer, Partitioning, pipelining, Accelerated Cascading, Symmetry Breaking, Synchronization (Locked, Lock-free) Parallel Algorithms and Data organization for shared/distributed memory, Min/Max, Sum Searching, Merging, Sorting, Various Parallel Sorting and Sorting Networks, Introduction to Graphics Processing Units (GPUs), CUDA Programming Model, Various operation in CUDA, Optimization Techniques: Understanding thread and blocks execution, Memory Bank Conflicts, Parallel Thread Execution, Control Flow, Precision, Optimizing CPU-GPU usage.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Design efficient parallel algorithms and applications.

CO2: To use large scale parallel machines to solve problems.

Text Books:

1. Parallel Programming Techniques and Applications by B. Wilkinson and M. Allen, Pearson.
2. An Introduction to Parallel Algorithms by Joseph Jaja, Addison-Wesley Professional.

Reference Books:

1. Parallel Computers- Architecture and Programming by V. Rajaraman and C.S.R Murthy, PHI.
2. Programming Massively Parallel Processors: A Hands-on Approach by David Kirk and Wen-mei Hwu, Elsevier.
3. Parallel Programming in C with MPI and OpenMP by Michael J Quinn, McGraw Hill.
4. CUDA by Example: An Introduction to General-Purpose GPU Programming by Jason Sanders and Edward Kandrot, Nvidia.

Course Name: **Information Theory and Coding**
Course Code: **CS-718**
Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To learn concepts in information theory, and the performance characteristics of an ideal communications system.
- To know about the fundamentals in Information coding and its applications.

Course Content

Introduction to information Theory: Information and entropy, properties of entropy of a binary memory less source, Measure of Information, Source Coding, Shannon Fano coding, Huffman coding, Lempel Ziv coding, channel coding, Channel capacity, noisy channel. Coding theorem for DMC. Linear block codes, generator matrices, parity check matrices, encoder syndrome and error detection minimum distance, error correction and error detection capabilities, cyclic codes, coding and decoding. Coding convolutional codes, encoder, generator matrix, transform domain representation state diagram, distance properties, maximum likelihood decoding, Viterbi decoding, sequential decoding, interleaved convolutional codes.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understood that how the quantity of information could be measure.
CO2: Understood the concept and properties of entropy and mutual information as applied to information.
CO3: Construct compact and non-compact codes for a given data ensemble.

Text Books:

1. Elements of Information Theory by T. M. Cover and J. A. Thomas, Wiley.
2. Information Theory Coding and Cryptography by R. Bose, Tata McGraw Hill.
3. The Theory of Error Correcting Codes by F. J. MacWilliams and N. J. A. Sloane, Elsevier.

Reference Books:

1. Coding and Information Theory by S. Roman, Springer.
2. The Theory of Information and Coding by R. J. McEliece, Cambridge University Press.

Course Name: **Game Theory**
Course Code: **CS-719**
Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To learn basic concepts of game theory.
- To learn to apply game theory on real world applications

Course Content

Introduction: games and decisions, Games Strategies, Costs and Payoff, Basic Solution Concepts, Finding equilibria and Learning in Games. Zero-sum games: secure strategy, Maximin, Maximax, and Minimax Regret Solvability, value of a game. Normal form games: dominance, iterated dominance, Nash equilibrium. N-player games, mixed strategy nash equilibria. Graphical Games: Computing Nash equilibria in Tree Graphical Games, Graphical Games and correlated Equilibria. Extensive form games: subgame perfection, sequential equilibrium, Stackelberg Model of Duopoly, Buying Votes, Committee Decision-Making. Bargaining: Rubinstein bargaining, Nash bargaining. Repeated games: Folk theorem and repeated prisoner's dilemma. Tacit collusion. Incomplete information games: Bayesian equilibrium, higher order beliefs. Auctions and mechanism design: Basic auctions, voting, Vickrey-Clarke-Groves Auction. Cryptography and Game theory: cryptographic influence on game theory and Game theoretic influence on cryptography.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Solve strategic games between two and more agents in non-cooperative scenario.

CO2: Analyze and solve both simultaneous-moves and sequential-moves games.

CO3: Learn different methods to solve games.

Text Books:

1. A Course in Game Theory by M. J. Osborne and A. Rubinstein, MIT Press.
2. An Introduction to Game Theory by M. J. Osborne, Oxford University Press.

Reference Books:

1. Algorithmic Game Theory by N. Nisan, T. Rougharden, E. Tardos and V. V. Vazirani, Cambridge University Press.
2. Fun and Games: A Text on Game theory by K. Binmore, AIBS publisher.

Course Name: **Combinatorial Optimization**
Course Code: **CS-720**
Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- Introducing the fundamentals of combinatorial optimization techniques.
- Learning how to model problems using mathematical programs.
- Learn to formulate and solve traditional problems in combinatorial optimization using combinatorial algorithm.
- Introducing the concept of linear programming and matching algorithm.

Course Content

Optimization Problem: Global and Local Optima; Convex sets and functions; Convex Programming Problem; Simplex Algorithm: Forms of linear programming problem; Geometry of linear program; Duality: Dual of a linear program in general form; shortest path problem and its dual; Dual simplex algorithm; Primal dual algorithm: Shortest Path Problem, Max Flow; Algorithms and complexity: Computability; time bound; analysis of algorithm; polynomial time algorithm; Algorithm for matching; weighted matching. Special topics in Combinatorial Optimization.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Identify and classify combinatorial optimization problems with real-world problems.
CO2: Identify, classify and implement algorithm to solve combinatorial optimization problems.
CO3: Model problems using linear programming.
CO4: An understanding of the inherent complexity of problems: Polynomial time, NP-completeness.

Text Books:

1. Combinatorial optimization: algorithm and Complexity by C.H. Papadimitriou and K. Steiglitz, Prentice Hall of India.
2. Art of Computer Programming by D. Knuth, Vol. IV, Addison Wesley.

Reference Books:

1. Computational Complexity by C.H. Papadimitriou, Addison Wesley.

Course Name: **Intrusion Detection System**
Course Code: **CS-721**
Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To understand when, where, how, and why to apply Intrusion Detection tools and techniques in order to improve the security posture of an enterprise.
- To apply knowledge of the fundamentals and history of Intrusion Detection in order to avoid common pitfalls in the creation and evaluation of new Intrusion Detection Systems.
- To analyze intrusion detection alerts and logs to distinguish attack types from false alarms.

Course Content

Introduction: IDS, Types of IDS (host based, Network based, Stack based, signature Based and anomaly based) TCP/IP and security concerns, DNS and security concerns, Mail server and security concerns, Web Server and security concern, firewall, Types of Intrusion, Symptoms that help in intrusion detection, statistical pattern recognition for detection and classification of attacks, vulnerabilities and Threats; Trojan Remote Access Trojan RAT, Virus, Worms and Malwares. Data Collection Mechanism: Data Sampling, Packet Sampling, Flow Sampling, techniques for visualizing network data, Packet Sampling tools, Tcpdump windump, Wireshark tool, Writing Tcpdump/Windump Filters, libcap/winpcap libraries, pcap file, sniffing and spoofing tools, data and methodologies of computer intrusion detection, statistical & machine approaches to detection of attacks on computers. Attacks and Packet analysis: Network based attacks such as probes & denial of service attacks, host-based attacks such as buffer overflows and race conditions, malicious codes, Examining Packet Header Fields, normal and abnormal values in IP, TCP, UDP, and ICMP header fields, Fragmentation theory, packet capture examples, fragmentation-based attacks, ICMP protocol, ICMP based attacks,
Network Traffic Analysis: malicious, normal and application traffic; discern malicious traffic from false positives. IDS Patterns, DoS attacks, network mapping, and coordinated attacks, Indications & Warnings and Traffic Correlation, Network correlation, Network Situational Awareness, anomaly detection, signature-based analysis, Semantic aware signature, policy-based analysis, and host-based analysis. IDS infrastructure: IDS Architecture, IDS/IPS Management and Architecture Issues with regard to deploying IDS/IPS systems, end point approach to security, system approach to security, IDS Interoperability models: CIDF (Common Intrusion Detection Framework), IDMEF (Intrusion Detection Message Exchange Format), IODEF (Incident Object Description Exchange Format), CVE (Common Vulnerabilities and Exposures), OVAL (Open Vulnerability and Assessment Language). IDS tools: Introduction to various IDS tools. Snort and Bro IDS tools, NIDS Evasion, Insertion, and Checksums to confuse NID systems, Snort Fundamentals and Configuration.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Explain the fundamental concepts of Network Protocol Analysis and demonstrate the skill to capture and analyze network packets.

CO2: Use various protocol analyzers and Network Intrusion Detection Systems as security tools to detect network attacks and troubleshoot network problems.

Books and References

1. Network Intrusion Detection by Stephen Northcutt and Judy Novak, Sams Publishing.
2. Extrusion Detection: Security Monitoring for Internal Intrusions by Bejtlich, Pearson Education.
3. Guide to Intrusion Detection and Prevention Systems (IDPS) by Karen Scarfone and Peter Mell, National Institute of Standards and Technology (NIST).

Reference Books:

1. CCNP Security: Intrusion Prevention and Intrusion Detection Systems by David Burns, Odunayo Adesina and Keith Barker, Cisco Press.
2. Intrusion Detection and Correlation: Challenges and Solutions by Christopher Kruegel, Fredrik Valeur and

Course Name: **Computer Vision**

Course Code: **CS-722**

Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To study the development of algorithms and techniques to analyze and interpret the visible world around us.
- To understand the basic concepts of Computer Vision.
- To Understand the geometric relationships between 2D images and the 3D world.
- To enable students to apply the various concepts of Computer Vision in other application areas.

Course Content

Introduction: Computer vision and its applications, Digital image formation and low-level processing: Fundamentals of Image Formation, Image Enhancement in Spatial Domain; Gray Level Transformation, Histogram Processing, Spatial Filters. Image Transforms: Fourier Transform and their properties. Feature Extraction: Edges - Canny, Laplacian of Gaussian (LoG), Difference of Gaussians (DOG); Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Orientation Histogram, Histogram of Oriented Gradients (HOG), Scale Invariant Feature Transform (SIFT), Speeded up Robust Features (SURF), Gabor Filters. Image Segmentation: Region Growing, Edge Based approaches to segmentation, Graph-Cut, Mean-Shift, Markov Random Fields (MRFs), Texture Segmentation; Object detection. Pattern Analysis: Basics of Probability and Statistics, Clustering: K-Means, K-Medoids, Mixture of Gaussians, Classification: Discriminant Function, Supervised, Un-supervised, Semi-supervised; Classifiers: Bayes, k-nearest neighbours (KNN), Artificial Neural Network models; Dimensionality Reduction: Principal Component Analysis, Linear Discriminant Analysis, Independent Component Analysis; Non-parametric methods. Applications of Computer Vision: Artificial Neural Network for Pattern Classification, Convolutional Neural Networks, Gesture Recognition, Motion Estimation and Object Tracking

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understand the fundamental problems of computer vision.

CO2: Implement various techniques and algorithms used in computer vision.

CO3: Analyze and evaluate critically the building and integration of computer vision algorithms

CO4: Demonstrate awareness of the current key research issues in computer vision.

Text Books:

1. Computer Vision Algorithms and Applications, Richard Szeliski, Springer.
2. Computer Vision: A Modern Approach Hardcover, David Forsyth and Jean Ponce, Pearson.
3. Digital Image Processing by R. Gonzalez and r. E. Wood, Prentice Hall of India.
4. Introductory Computer Vision and Image Procession by Andrian low, McGraw Hill CO.

References Books:

1. Digital Image Processing by W.K. Pratt, McGraw Hill.
2. Computer Vision: Models, Learning, and Inference by Simon J. D. Prince, Cambridge University Press.

Course Name: **Fault Tolerant Computing**

Course Code: **CS-723**

Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To know the different advantages and limits of fault avoidance and fault tolerance techniques.
- To impart the knowledge about different types of redundancy and its application for the design of computer system being able to function correctly even under presence of faults and data errors.
- To understand the relevant factors in evaluating alternative system designs for a specific set of requirement.
- To understand the subtle failure modes of "fault-tolerant" distributed systems.

Course Content

Introduction to Fault Tolerant Computing: Faults and their manifestations, Fault/error modeling, Reliability, availability and maintainability analysis, System evaluation, performance reliability tradeoffs. System level fault diagnosis, Hardware and software redundancy techniques. Fault tolerant system design methods, Mobile computing and Mobile communication environment, Fault injection methods, Software fault tolerance, Design and test of defect free integrated circuits, fault modeling, built in self-test, data compression, error correcting codes, simulation software/hardware, fault tolerant system design, CAD tools for design for testability. Information Redundancy and Error Correcting Codes, Software Problem. Software Reliability Models and Robust Coding Techniques, Reliability in Computer Networks Time redundancy. Re execution in SMT, CMP Architectures, Fault Tolerant Distributed Systems, Data replication. Case Studies in FTC: ROC, HP Non Stop Server. Case studies of fault tolerant systems and current research issues.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Become familiar with general and state of the art techniques used in design and analysis of fault- tolerant digital systems.

CO2: Be familiar with making system fault tolerant, modeling and testing, and benchmarking to evaluate and compare systems.

Text Books:

1. Fault Tolerant Computer System Design by D. K. Pradhan, Prentice Hall.
2. Fault Tolerant Systems by I. Koren, Morgan Kauffman Publishers.

Reference Books:

1. Software Fault Tolerance Techniques and Implementation by L. L. Pullum, Artech House Computer Security Series.
2. Reliability of Computer Systems and Networks: Fault Tolerance Analysis and Design by M. L. Shooman, Wiley.

Course Name: **Biometric Systems and Security**

Course Code: **CS-724**

Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To describe principles of the selected physical and behavioral biometric methods and know how to deploy them in authentication scenarios.
- To organize and conduct biometric data collection processes and understand how to use biometric databases in system evaluation.
- To calculate distributions of between- and between-class matching scores and calculate various error estimates based on these distributions.
- To understand the biometrics security issues and know how to deploy selected liveness detection techniques to make a system spoof resistant.

Course Content

Notion of biometrics: Development of biometric authentication. Basic terms, biometric data, biometric characteristics, biometric features, biometric templates and references. Expected properties of biometric identifiers. Basics in biometric errors estimation. Enrolment, Verification, and Identification. Statistical evaluation of biometrics: Technology, Scenario, and operational evaluations. Errors of biometric systems, false non-match vs. false rejection, false match vs. false acceptance. Error curves, ROC, DET, CMC. Statistical error estimation, hypothesis testing, Principles of biometric database collection and usage. Fingerprint recognition: Fingerprint capture, sensor types, latent fingerprints. Fingerprint image preprocessing, segmentation, binary and skeletal images, Fingerprint singularities, detection of loops, deltas, whirls, and cores, using singularities in fingerprints classification. Galton's details, base and complex minutiae, detection of minutiae. Fingerprint recognition, minutiae- and correlation-based methods. Fingerprint quality and analysis, Fingerprints in forensics and biometrics, similarities, and differences.
Iris recognition: Eye and iris morphogenesis, genetic penetrance. Principles of iris image capture, iris sensors. Iris image preprocessing, segmentation, formatting, and filtering. Daugman's method, iris code, statistical properties of the iris code, other iris coding methods, wavelet analysis. Face recognition: Face detection in still images and sequences. Face features. Face space, principal component analysis and its application, eigenfaces, linear discriminant analysis and its application, Fisher faces. Face recognition methods. Security of biometrics: presentation attack detection, Static and dynamic liveness features. subversive actions vs. suspicious actions, Liveness detection in finger- and eye-based biometrics. Selected liveness detection techniques, frequency analysis for paper printouts detection, pupil dynamics and blood pulse analyses for detection of sophisticated eye and fingerprint spoof trials.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understand the fundamental concepts of biometric systems and their characteristics.

CO2: Understand the evaluation measures of different biometric modalities.

CO3: Analyze pros and cons of different biometric traits used in real world applications.

CO4: Evaluate the security aspects of different biometric systems.

Text Books:

1. Jain, A.K., Ross, A., Nandakumar, K. Introduction to Biometrics.
2. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S., Handbook of Fingerprint Recognition.
3. Burge, M.J., Bowyer, K., Handbook of Iris Recognition.
4. H. Wechsler, Reliable Face Recognition Methods: System Design, Implementation and Evaluation, Springer.

Reference Books:

1. Marcel, S., Nixon, M.S., Li, S.Z., Handbook of Biometric Anti-Spong: Trusted Biometrics under Spoong Attacks (Advances in Computer Vision and Pattern Recognition).

Course Name: **Cluster and Grid Computing**

Course Code: **CS-725**

Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To provide an insight for achieving cost-efficient high performance system.
- To provide understanding of the design and architecture of grid and cluster computing.

Course Content

Introduction: High Performance Computing (HPC), Grand Challenge Problems Computational and communication intensive, Parallel Architectures Classifications SMP, MPP, NUMA, Clusters and Components of a Parallel Machine, Conventional Supercomputers and its limitations, Multi-processor and Multi Computer based Distributed Systems. Cluster and Grids: Cluster Components Processor/machine, High Speed Interconnections goals, topology, latency, bandwidth, Example Interconnect: Myrinet, Inifiniband, QsNet, Fast Ethernet, Gigabit Ethernet, Light weight Messaging system/Light weight communication Protocols, Cluster Middleware Job/Resource Management System, Load balancing, Scheduling of parallel processes, Enforcing policies, GUI,

Introduction to programming tools such as PVM, MPI, Cluster Operating Systems Examples: Linux, MOSIX, CONDOR, Characteristics of Grid, Computational services, Computational Grids, Data grids/Storage grids, management and applications, Different components of Grid fabric, Grid middleware, Grid applications and portal, Globus toolkit Ver.2.4, web services, MDS,GRAM, Grid Security Cryptography, Authentication, Integrity, Digital Signature, Digital Certificates, Certificate Authority, RSA, GSI,GSSAPI, Directory Service, LDAP,GRID FTP,GASS, SHA1 & SHA2. Fault Tolerance: Fault detection and diagnosis of Clusters and Grids. Recent advances in cluster and grid computing.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Have knowledge of Grid Computing, Web Services, and Service-oriented architecture, Architecture for grid computing, Cluster Computing, Process scheduling and load balancing.

CO2: Identify the resource selection for Grid environment.

CO3: Understand the data management and transfer in Grid environments.

Text Books:

1. Grid Computing by D. Janakiram, Tata Mcgraw Hill.
2. High Performance Cluster Computing, Volume 1 and 2 by R. K. Buyya, Prentice Hall.
3. Fault Tolerance in Distributed Systems by P. Jalote, Prentice Hall.

Reference Books:

1. Cluster Computing by R. K. Buyya and C. Szyperski, Nova Science, New York, USA.
2. Market oriented Grid and Utility Computing by R. K. Buyya and K. Bubendorfer, Wiley.
3. Grid Computing by J. Jaseph and C. Fellenstein, Pearson.

Course Name: **Cloud Computing**

Course Code: **CS-726**

Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To provide the understanding of basic concepts of cloud computing along with their service models and deployment models.
- To give the fundamental idea of virtualization, load balancing, and cloud management.
- To shed light on research trends in cloud computing like fog computing, edge computing.

Course Content

Introduction: Overview of cloud computing, Benefits, Cloud computing architecture, Services: Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS), Anything as a Service (XaaS), Big data as a service (BDaaS), Cloud deployment models (public, private, hybrid, virtual private, community, managed), Cloud applications, Case Study on open source and commercial clouds: Eucalyptus, Microsoft Azure, Amazon EC2.
Virtualization: Overview, Benefits, Types of virtualization, Virtual machine (background, creation, management), Virtualization of datacenters and Issues with multi-tenancy, Virtual machines provisioning and migration services, Fault tolerance mechanisms.
Resource management and load balancing: Distributed management of virtual infrastructure, Server consolidation, Dynamic provisioning and resource management, Resource optimization, Resource dynamic reconfiguration, Scheduling techniques for VM reservation, and various load balancing techniques, Service management in cloud Computing, Data management in cloud computing, SLA management in cloud computing, Cloud security.
Hypervisor/Virtual Machine Monitor (VMM): Introduction to hypervisor, Internals of VMM/hypervisor, Open source and commercial hypervisors, Study of various hypervisors, Cloud simulator. Research trend in cloud computing: Virtual machine migration in WAN, Energy efficient cloud computing, Fog Computing and Edge computing: Concepts, Principles and Paradigms.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Articulate the main concepts, key technologies, strengths, and limitations of cloud computing.

CO2: Identify the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud etc.

CO3: Explain the core issues of cloud computing such as security, vm migration, load balancing and cloud management.

CO4: Attempt to generate new ideas and innovations in cloud computing.

Text Books:

2. Cloud Computing principles and paradigms by Rajkumar Buyya, James Broberg, and Andrzej Goscinski, Wiley.
3. Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate by Michael Miller, Que Publishing.
4. Cloud Computing by M.N. Rao, PHI Learning.

Reference Books:

1. Mastering Cloud Computing by Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi, Tata McGraw Hill.
2. Fog and Edge Computing: Principles and Paradigms by Rajkumar Buyya and Satish Narayana Srirama, Wiley-Blackwell.

Course Name: **Blockchain Technology**

Course Code: **CS-727**

Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To impart knowledge of underlying mechanism of blockchain and provides the necessary knowledge needed for starting blockchain programming.
- To introduce blockchain technologies created to realize cryptocurrency like bitcoin and Ethereum.

Course Content

Introduction: Blockchain basics, The consensus problem - Asynchronous Byzantine Agreement - AAP protocol and its analysis - Nakamoto Consensus on permission-less, nameless, peer-to-peer network. Models: Abstract Models for Blockchain - GARAY model - RLA Model - Proof of Work (PoW) as random oracle - formal treatment of consistency, liveness and fairness - Proof of Stake (PoS) based Chains - Hybrid models (PoW + PoS) and other models.

Cryptographic Basics for Cryptocurrency: A short overview of Hashing, signature schemes, encryption schemes and elliptic curve cryptography. Bitcoin - Wallet - Blocks - Merkley Tree - hardness of mining - transaction verifiability - anonymity - forks - double spending - mathematical analysis of properties of Bitcoin. Ethereum - Ethereum Virtual Machine (EVM) - Wallets for Ethereum - Solidity - Smart Contracts - some attacks on smart contracts.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Familiarize the functional/operational aspects of cryptocurrency ECOSYSTEM.

CO2: Understand emerging abstract models for Blockchain Technology.

CO3: Identify major research challenges and technical gaps existing between theory and practice in cryptocurrency domain.

Text Books:

1. Mastering Bitcoin, 2nd Edition by Andreas M. Antonopoulos, Reilly Media.
2. Bitcoin and Cryptocurrency Technologies Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder.
3. Blockchain Principles and Applications in IoT, Rajdeep Chakraborty, Anupam Ghosh, Valentina Emilia Balas and Ahmed A. Elngar, Chapman & Hall.

Reference Books:

1. Blockchain Technology: Exploring Opportunities, Challenges, and Applications, Sonali Vyas, Vinod Kumar Shukla, Shaurya Gupta, Ajay Prasad, CRC Press.
2. Blockchain and Web3: Building the Cryptocurrency, Privacy, and Security Foundations of the Metaverse, Winston Ma and Ken Huang, Wiley.
3. Enterprise Strategy for Blockchain: Lessons in Disruption from Fintech, Supply Chains, and Consumer Industries (Management on the Cutting Edge), Ravi Sarathy, MIT Press.

Course Name: **Advance Mobile Communication**
Course Code: **CS-728**
Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To understand the basic cellular system concepts.
- To have an insight into the interference, frequency management and handoff management in cellular mobile system.
- To go in depth for understanding the popular GSM cellular mobile standard and wireless standards.

Course Content

Evolution from 1G to 5G: Analog voice systems in 1G; digital radio systems in 2G, voice and messaging services, TDMA based GSM, CDMA, 2.5G (GPRS), 2.75G (EDGE). IMT2000: 3G UMTS, W-CDMA, HSPA, HSPA+, 3G services and data rates; IMT Advanced: 4G, LTE, VOLTE, OFDM, MIMO, LTE Advanced Pro (3GPP Release 13+); IMT2020: 5G, enhancements in comparison to IMT Advanced.

Basics of 5G: 5G potential and applications, Usage scenarios: enhanced mobile broadband (eMBB), ultra-reliable low latency communications (URLLC), massive machine type communications (MMTC), D2D communications, V2X communications. spectrum access/sharing: millimeter Wave communication, channels and signals/waveforms in 5G, carrier aggregation, small cells, dual connectivity. New Radio (NR), Standalone and non-standalone mode; non-orthogonal multiple access (NOMA); massive MIMO, beam formation, FAPI: PHY API Specification, flexible frame structure, Service Data Adaptation Protocol (SDAP), Centralized RAN, open RAN; multi-access edge computing (MEC); software defined networking (SDN), network function virtualization (NFV); network slicing; restful API for service-based interface; private networks.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Discuss cellular radio concepts.

CO2: To have knowledge of the mobile system specifications.

CO3: Classify frequency and handoff management techniques in mobile communication.

CO4: Outline cellular mobile communication standards.

CO5: Analyze various methodologies to improve the cellular capacity.

Text Books:

1. Mobile Cellular Telecommunications: Analog and Digital Systems by W. C. Y. Lee; Tata McGraw Hill Publication.
2. Wi-Fi, Bluetooth, Zigbee and WiMax by H. Labiod, H. Afifi and C. D. Santis, Springer.
3. Wireless Communications: Principles and Practice by T. S. Rappaport; Pearson Publication.

Reference Books:

1. Wireless Communications and Networks: 3G and Beyond by I. S. Misra; Tata McGraw Hill Publication.
2. Wireless and Digital Communications by K. Feher; PHI Publication.
3. 4G, LTE-Advanced Pro and The Road to SG by Erik Dahlman.

Course Name: **Networked Wireless Systems**
Course Code: **CS-729**
Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To impart knowledge about the Networks, Resource sharing, Design issues for the network layers.
- To introduce the fundamental concepts related to the network architectures.
- To enable the students to understand the Data centric, hierarchical, location-based, energy efficient routing etc.

Course Content

Introduction to Networked wireless systems, Sensor network, IoT etc.

Sensor Networks: Introduction, applications, design issues, requirement of Sensor node architecture. Overview and motivation, Resource sharing, Design issues for the network layers Networking devices. Principles of internetworking, Tunneling, Fragmentation, Naming and addressing concepts, Hierarchical naming, Domain name system, Name resolution process, IP address classes and concept of sub netting, Classless Inter-domain routing (CIDR) and DHCP concepts. Network architecture: The internet protocols: IP, ICMP, ARP and RARP. The design issues for the transport layer, addressing, establishing connection, flow control and multiplexing. The internet protocols: TCP and UDP, Multicast routing, Mobility in networks, Mobile IP, Emerging trends in networking. Optimization goals, evaluation metrics, network design principles. Sensor network operating systems and brief introduction to sensor network programming.

Network & Routing protocols: MAC protocols; contention based and contention free based MAC protocols, hybrid protocols and energy efficiency; need for energy efficiency and power control, Active and Passive Power Conservation Mechanisms, Data centric, hierarchical, location-based, energy efficient routing etc. Sensor deployment, scheduling and coverage issues, self-configuration and topology control. Data Processing: Querying, data collection and processing, collaborative information processing and group connectivity. Target tracking, localization and identity management. Power management, Security and privacy.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Implement and realize various privileged operations critical to the functioning of an operating system and network.

CO2: Understand the basics of network architecture.

CO3: Understand various addressing and sub-netting techniques, routing protocols, etc.

CO4: Understanding data collection and processing concepts of sensor and IoT networks.

Text Books:

1. Wireless Sensor Networks: A Networking Perspective by Jun Zheng, Abbas Jamalipour, Wiley-IEEE Press.
2. Wireless Sensor Networks-An Information Processing Approach by Feng Zhao, Leonidas Guibas, Morgan Kauffman.
3. Wireless Sensor Networks: From Theory to Applications by Ibrahiem M. M. El Emary and S. Ramakrishnan, CRC Press.
4. Wireless sensor networks by Edgar H. Callaway, AUERBACH Publications.

Reference Books:

1. Wireless Sensor Networks: Principles and Practice by Fei Hu and Xiaojun Cao, CRC Press.
2. Protocols and Architectures for Wireless Sensor Networks by Andreas Willig and Holger Karl, Wiley Publications.

Course Name: **Quantum Computing**
Course Code: **CS-730**
Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- Introduction to the fast-growing field of quantum computing.
- To Study the structural units of quantum computers of the future, forming an understanding of the differences between quantum bits and classical bits.
- To study of basic quantum logical operations and algorithms for processing quantum information.
- Mastering basic knowledge about the practical use of quantum algorithms and quantum programming skills.

Course Content

Introduction: Information and Computations. Characteristics of Computational Systems. Computability and Algorithms. Computational Complexity. Classical computation versus quantum computation. Quantum Computing. The Multiverse Interpretation of Quantum Mechanics. Linear Algebra for Quantum Computing: Review of linear algebra in the context of quantum information, Dirac's bracket notation, limitation of classical algorithms. The four postulates of quantum mechanics, qubits, quantum gates and circuits. Error correction and fault-tolerant quantum computing.
Introduction to Quantum Algorithms: I-Deutsch's algorithm, analysing quantum algorithms, and implementing quantum circuits via QISKIT. Basic quantum algorithms II-Simon's problem and the Bernstein -V-azirani algorithm.
Grover's quantum search algorithm, the BBBV Theorem, and applications of Grover's algorithm. RSA, and Shor's integer factorization algorithm. Quantum Cryptography: Introduction to quantum cryptography (post-quantum security, quantum key distribution). Introduction to quantum information (superdense coding, nocloning theorem, quantum teleportation) Applications (quantum money, the Elitzur-Vaidman bomb).

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Analyze the behavior of basic quantum algorithms.
CO2: Implement simple quantum algorithms and information channels in the quantum circuit model.
CO3: Simulate a simple quantum error-correcting code
CO4: Understand the principles of quantum information and quantum communication: - Understand quantum teleportation and its limits.

Text Books:

1. Nielsen, M. A.. Cluster-state quantum computation.
2. Nielsen, M. A., & Chuang, I. L. Quantum Computation and Quantum Information (Vol. 10th anniversary ed).
3. Hassanien, A. E., Elhoseny, M., & Kacprzyk, J. Quantum Computing:An Environment for Intelligent Large Scale Real Application.

Reference Books:

1. Sakkaris, P., & Sudhakaran, R. A Multilayer Network Approach to Quantum Computing.
2. Chris Bernhardt, Quantum Computing for Everyone (The MIT Press).
3. Riley Tipton Perry, "Quantum Computing from the Ground Up", World Scientific Publishing Ltd.

Course Name: **Machine Learning**
Course Code: **CS-731**
Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To impart knowledge about the concepts of machine learning.
- To introduce students about various regression and classification techniques.
- To revisit concepts of probability etc. to get in-depth knowledge of machine learning.

Course Content

Introduction to Machine Learning: Problems, data, and tools, Visualization tools, Decision Tree Learning, Artificial Neural Networks, Bayesian Learning, Deep Learning, Instance-Based Learning. Linear regression, Logistic regression, online gradient descent, Kernel Methods, SSE, closed form, normal equations, features, Overfitting and complexity, training, validation, test data, Classification problems, decision boundaries, nearest neighbor methods. Bayes optimal decisions, Naive Bayes and Gaussian class-conditional distribution, Linear classifiers: Bayes Rule and Naive Bayes Model. Radial Basis Function Networks, Support Vector Machines, Genetic Algorithms, Reinforcement Learning. Bagging, random forests, boosting, Unsupervised learning: clustering, k-means, hierarchical agglomeration, Latent space methods, PCA, Text representations, naive Bayes and multinomial models. Clustering and latent space models, VC-dimension, structural risk minimization, margin methods and support vector machines (SVM), Machine Learning Applications.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Develop an understanding what is involved in learning models from data.
CO2: Understand a wide variety of learning algorithms.
CO3: Apply principles and algorithms to evaluate models generated from data.
CO4: Apply the algorithms to a real-world problem

Text Books:

1. Introduction to Machine Learning by Ethem Alpaydin, PHI Learning.
2. Machine Learning: An Algorithmic Perspective by Stephen Marsland, Chapman and Hall/CRC.
3. Pattern Recognition and Machine Learning by Christopher M. Bishop, Springer.

Reference Books:

1. Machine Learning by Tom Mitchell, McGraw Hill Education.
2. Fundamentals of Machine Learning for Predictive Data Analytics, John D. Kelleher, Brian Mac Namee, and Aoife D'Arcy, MIT Press.

Course Name: **Advances in Machine Learning**

Course Code: **CS-732**

Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To introduce the principles of deep learning and its applications.
- To enable the students in practical skills to design, implement, and train practical deep learning systems.
- To provide a structured approach covering Fundamentals of Machine Learning, Neural Networks, Modern Deep Learning, and Applications and other advanced topics.

Course Content

Introduction to Deep Learning: Overview of neural networks and deep learning, Deep learning applications in computer vision, natural language processing, Deep Neural Networks: Early Models, Perceptron Learning, Multilayer Perceptrons (MLPs) and feedforward neural networks, Backpropagation, Initialization, Training & Validation, Parameter Estimation - MLE, MAP, Bayesian Estimation, Activation functions, loss functions, and optimization techniques, Regularization methods and hyperparameter tuning; Introduction to Convolutional Neural Networks (CNNs), Architecture; Recurrent Neural Networks (RNNs): Introduction to RNNs and Long Short-Term Memory (LSTM) networks, Generative Adversarial Networks (GANs): Fundamentals of GANs and adversarial training, Introduction to Reinforcement Learning, Transformers, Large Language Models. Introduction to Generative AI: Definition and scope, Applications of Generative AI, Importance of Generative AI in various domains, Ethical considerations and challenges, Introduction to language models and their role in AI, Traditional approaches to language modelling, Deep learning based approaches, Introduction to GPT and its significance, Pre-training and fine-tuning processes in GPT, Architecture and working of GPT models, Overview of GPT variants and their use cases

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Demonstrate the ability to apply concepts from linear algebra, probability, optimization, and machine learning to solve complex problems in the context of deep learning.

CO2: Evaluate the advantages and disadvantages of deep learning neural network architectures and compare them with other approaches in the context of case studies.

CO3: Design, implement, and train deep learning models using convolutional, recurrent, and other neural network architectures to address real-world problems effectively.

CO4: Analyze, design, and implement solutions to real-world computer vision problems, NLP and other problems using deep learning techniques.

Text Books:

1. Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT Press.
2. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aurélien Géron, O'Reilly Media.
3. Neural Networks and Deep Learning by Michael Nielsen, Determination Press.

Reference Books:

1. Deep Learning for Computer Vision by Rajalingappaa Shanmugamani, Packt Publishing.
2. Generative Deep Learning by David Foster, O'Reilly Media.

Course Name: **Computer Graphics**
Course Code: **CS-733**
Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To introduce the theoretical concepts of display devices in computer graphics.
- To impart knowledge about the various approaches for curve, Geometric Transformations and Clipping.
- To enable the students to understand the basics surface detection and applications.

Course Content

Basics of computer graphics and devices: Introduction of graphics, Area of Computer Graphics, Design and Drawing, Animation Multimedia applications, Simulation, how are pictures actually stored and displayed, Difficulties for displaying pictures. Graphics devices- Cathode Ray Tube, Quality of Phosphors, CRTs for Color Display, Beam Penetration CRT, The Shadow - Mask CRT, Direct View Storage Tube. Simple line drawing methods: Point Plotting Techniques, Qualities of good line drawing algorithms, The Digital Differential Analyzer (DDA), Bresenham's Algorithm, Generation of Circles, scan line polygon fill, boundary fill and flood fill algorithms. Two Dimensional Geometric Transformations And Window Clipping: Basic transform-translation, rotation, scaling, homogeneous coordinates, composite transform, affine transform, Need for Clipping and Windowing, point clipping, line clipping-Cohen-Sutherland Line Clipping, Liang-Barsky line clipping, polygon clipping-Sutherland-Hodgeman Polygon clipping

Three-dimensional modeling and geometric transform: Translation, Scaling, reflection, shears, composite transform, Modelling Coordinate Transform, three-dimensional clipping, perspective view of cube.

Three-Dimensional input techniques and object representation: Display methods, visible line and surface identification, surface rendering, curved lines and surfaces, spline representations, Bezier Curves and surfaces

Visible-Surface Detection: Basic classification of visible surface detection algorithm, depth buffer method, A- Buffer method, scan line method, BSP-tree method, area subdivision method, Octree method.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understand 2D and 3D Transform.

CO2: Understand and apply the concept of modeling and clipping on various problems.

CO3: Can solve given problem on surface rendering and detection.by selecting the appropriate algorithm design technique and justify the selection.

CO4: Know the concepts of various aspect of graphics that can be applied on any domain related problem

Text Books:

1. Computer Graphics by D D Heran, M. P Baker, Pearson
2. Computer Graphics by S Bhattacharya, Oxford

Reference Books:

1. Computer Graphics Principles and Pratice, J. D. Foley, A V Dam, S K. Feiner, J Hughes, M McGuire, D F. Sklar, and K Akeley, Addison–Wesley
2. Computer Graphics With an introduction to multimedia, R, Chopra, S, Chand

Course Name: **Digital Image Processing**

Course Code: **CS-734**

Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To impart knowledge about the fundamental steps of digital image processing.
- To introduce the fundamental and advanced concepts relevant to digital image processing.
- To able the students to understand the various image segmentation techniques and morphological algorithms.
- To able the students to understand the various image compression techniques

Course Content

Introduction: Digital image representation, Fundamental steps in image processing, Elements of Digital Image processing systems, Elements of visual perception, Image model, Sampling and quantization, Neighborhood relationship between pixels, Imaging geometry. Image Enhancement: Enhancement by point processing, Intensity transformation, Histogram processing, Image subtraction, Image averaging, Spatial filtering, Smoothing filters, Sharpening filters, Wavelet based Low-Pass, High-Pass filtering.

Image Segmentation: Detection of discontinuities, Edge linking and boundary detection, Thresholding, Region oriented segmentation, Use of motion in segmentation, Morphological Watershed techniques, Frequency domain techniques

Image Restoration: A model of the image degradation/restoration process, noise models, restoration in the presence of noise–only spatial filtering, Weiner filtering, constrained least squares filtering, geometric transforms; Introduction to the Fourier transform and the frequency domain, estimating the degradation function. Color Image Processing.

Morphological Image Processing: Preliminaries, dilation, erosion, open and closing, hit or miss transformation, convex hull Image Compression: Coding redundancy, Inter-pixel redundancy, fidelity criteria, image compression models, error-free compression, lossy predictive coding, image compression standards.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understand fundamental steps of digital image processing.

CO2: Understand and implement image enhancement techniques in spatial and frequency domain.

CO3: Implement and compare various image compression techniques.

CO4: Understand and implement pattern recognition and classification techniques.

Text Books:

1. Digital Image Processing, by R. Gonzalez and R. E. Wood, Prentice Hall of India
2. Introductory Computer Vision and Image Processing by Andrian Low. McGraw Hill.
3. Digital Image Processing using MATLAB by R. Gonzalez, R. E. Wood and Steven L. Eddins Gatesmark Publishing

Reference Books:

1. Pattern Recognition-Statistical, Structural and neural approach by Robert Schalkoff, John Willey & Sons.
2. Fundamentals of Digital Image Processing by A K Jain, Pearson.

Course Name: **Data Mining**

Course Code: **CS-735**

Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To introduce students to the basic applications, concepts, and techniques of data mining.
- To develop skills for using recent data mining software (example R) to solve practical problems in a variety of disciplines.
- To gain experience doing independent study and research.

Course Content

Introduction: Types of data mining problems. The process of data mining. Statistical evaluation of big data: statistical prediction, performance measures, pitfalls in data-mining evaluation. Data preparation: data models, data transformations, handling of missing data, time-dependent data, and textual data. Data reduction: feature selection, principal components, smoothing data, case subsampling.

Predictive modeling: Classification, mathematical models, linear models, neural nets, SVM, distance solutions, logic solutions, decision trees, rule based classifier, Bayesian Classifiers, model combination. Data mining algorithms: Cluster Analysis, basic concepts, algorithms and additional issues, association Analysis, Basic Concepts and Algorithms, Anomaly Detection. Solution analyses and Future trends: graphical trend analyses, comparison of methods text mining, visualization, distributed data. Case studies.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Categorize and carefully differentiate between situations for applying different data-mining techniques: frequent pattern mining, association, correlation, classification, prediction, and cluster and outlier analysis.

CO2: Design and implement systems for data mining.

CO3: Evaluate the performance of different data-mining algorithms.

CO4: Propose data-mining solutions for different applications.

Text Books:

1. Data Mining Concepts and Techniques by Jiawei Han and Michelien Kamber, Morgan Kaufmann.

2. Data Warehousing, Data mining and OLAP by Alex Berson and Stephen Smith, McGraw Hill.

Reference Books:

1. Introduction to Data Mining by Pang-Ning Tan Michael Steinbach Vipin Kumar

Course Name: **Architecture of Large Systems**

Course Code: **CS-736**

Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To impart knowledge about examining the qualitative and quantitative computer design tradeoffs.
- To introduce the fundamental concepts relevant to art of selecting and interconnecting hardware components to create a computer that meets functional, performance and cost goals.
- To enable the students to understand the basic non-classical architectures such as parallel processors, multi-core chips, pipelined and VLIW machines.

Course Content

Introduction: review of basic computer architecture, quantitative techniques in computer design, measuring and reporting performance, CISC and RISC processors. Pipelining: Basic concepts, instruction and arithmetic pipeline, data hazards, control hazards, and structural hazards, techniques for handling hazards, Exception handling, Pipeline optimization techniques. Compiler techniques for improving performance. Hierarchical memory technology: Inclusion, Coherence and locality properties; Cache memory organizations, Techniques for reducing cache misses; Virtual memory organization, mapping and management techniques, memory replacement policies. Instruction-level parallelism: basic concepts, techniques for increasing ILP, superscalar, super pipelined and VLIW processor architectures. Array and vector processors. Multiprocessor architecture: taxonomy of parallel architectures. Centralized shared-memory architecture: synchronization, memory consistency, interconnection networks. Distributed shared-memory architecture. Cluster computers. Non von Neumann architectures: data flow computers, reduction computer architectures, systolic architectures.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understand that how hardware and software (especially the operating system and compilers must work synergistically together to provide optimum throughput.

CO2: Understand the elements of modern computer along with measures of performance.

CO3: Understand the concepts of pipelining, multiprocessors, parallel processors, etc.

Text Books:

1. Computer Organization and Design: A Hardware/Software Interface by David Patterson and John Henessey, Morgan Kaufmann publication.
2. Computer Architecture: A Quantitative Approach by John Henessey & David Patterson, Morgan Kaufmann publication.

Reference Book:

1. Computer Organization and Architecture, Designing for Performance by William Stallings, Pearson Education.
2. Computer System Architecture by M. Morris Mano, Prentice Hall of India.

Course Name: **Embedded Systems**

Course Code: **CS-737**

Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To discuss the major components that constitutes an embedded system.
- To implement small programs to solve well-defined problems on an embedded platform.
- To develop familiarity with tools used in an embedded environment.

Course Content

Introduction: Embedded system, Processor, hardware units, software embedding, SOC, NOC, VLSI circuit; Device and Device drivers, I/O devices, timer and counting devices, serial communication using IC, LAN and advanced I/O buses between the networked multiple devices, Host system, parallel communication using ISA, PCI, PCI X, and advanced buses, device drivers, parallel port device drivers in a system, serial port device drivers. Interrupt service handling mechanism; Software and programming concepts: processor and memory selection for embedded system, embedded programming in C++, Java and UML, multiple processes and applications, problem of sharing data by multiple tasks and routines, interprocess communication; Real time OS: OS services, I/O subsystem, Network OS, Real time Embedded system, Need of well tested and debugged RTOS, Introduction to C/OS II. Case Studies of programming with RTOS: Smart card embedded system, Hardware and Software co design: specification and design of an embedded system, use of software tools for development of an embedded system. Recent advances in embedded applications.

Course Outcomes

Upon successful completion of the course, the students will be able to

C01: Understand steps and important properties of graphs and relative importance of graph mining.

C02: Understand and implement graph pattern matching for applications.

C03: Implement and compare various traversal approaches of graphs.

C04: Understand and apply graph mining for real time applications.

Text Books:

1. Embedded System Architecture, Programming and Design by R. Kamal, Tata McGraw Hill.
2. Hardware Software Codesign of Embedded System by R. Niemann, Kulwer Academic.
3. Embedded Real Time System Programming by Sriram. V. Iyer and P. Gupta, Tata McGraw Hill.

Reference Books:

1. Computer as Components: Principles of Embedded Computer System Design by W. Wolf, Elsevier.
2. Embedded System Design by S. Heath, Elsevier.
3. Real Time Systems Theory and Practice by R. Mall, Pearson.
4. Embedded System Design: A Unified Hardware/Software Approach by F. Vahid & T. Givargis, Wiley.
5. Network on Chips: Technology and Tools by G. D. Michelli and Luca Benin, Morgan & Kaufman Publication.

Course Name: **Graph Mining and Algorithms**

Course Code: **CS-738**

Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To impart knowledge about the various steps of graph mining approaches.
- To understand and solve pattern representation using graph.
- To understand the matching approaches in graph mining.
- To introduce the applicability of graph mining for real time applications.

Course Content

Introduction to graphs: Introduction to basic graphs terminology, graph representation, types of graphs, basic graph decomposition algorithms into constituent parts, connectivity of graphs, matching on graphs.

Graph Mining: Motivation for Graph Mining, Applications of Graph Mining, Mining Frequent. Subgraphs –Transactions, BFS, DFS, Diagonal and Greedy Approaches, Constraint-based mining. Graph traversal and matching algorithms: Graph colouring approach, graphs on surface, directed graphs, Shortest path algorithms, algorithms to discover minimum spanning tree, Flows in Networks and some important flow algorithms, Searching Graphs and Related algorithms. Exact Graph Matching, Inexact Graph Matching: Graph Edit Distance, Other Inexact Graph Matching Techniques Mining Frequent Subgraph: Apriori-based Approach, Pattern-Growth Approach, Closed and Maximal Subgraphs, Mining Subgraphs in a Single, the Computational Bottleneck

Mining Significant Graph Patterns: gboost: A Branch-and-Bound Approach, gPLS: A Partial Least Squares Regression Approach, LEAP: A Structural Leap Search Approach, raphSig: A Feature Representation Approach, Mining Representative Orthogonal Graphs: Randomized Maximal Subgraph Mining, Orthogonal Representative Set Generation. Graph Mining Applications: Web mining, centrality analysis, Link analysis approach, graph clustering and community detection, Software Bug Localization: Static call graphs, Dynamic call graphs, Node classification and Link prediction, Influential spreaders, Influence maximization, Geo-social and location based networks.

Course Outcomes

Upon successful completion of the course, the students will be able to

C01: Understand steps and important properties of graphs and relative importance of graph mining.

C02: Understand and implement graph pattern matching for applications.

C03: Implement and compare various traversal approaches of graphs.

C04: Understand and apply graph mining for real time applications.

Text Books:

1. Aggarwal, Charu C., and Haixun Wang, eds. Managing and mining graph data. New York: Springer.
2. Learning Resources 1. Diestel, R.Graph Theory, 4th ed.Springer-Verlag, Heidelberg.
3. J. Han and M. Kamber, Data minining–Concepts and Techniques, 2ndEdition, Morgan kaufmanPublishers.
4. J Leskovec, A Rajaraman, Jeff Ullman. Mining of Massive Datasets. Book 2nd edition. Cambridge University Press.

Reference Books:

1. David Easley and Jon Kleinberg. Networks, Crowds, and Markets. Cambridge University Press.
2. Deepayan Chakrabarti and Christos Faloutsos. Graph Mining: Laws, Tools, and Case Studies. Synthesis Lectures on Data Mining and Knowledge Discovery, Morgan & Claypool Publishers.

Course Name: **Internet of Things**
Course Code: **CS-739**
Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To impart knowledge about Internet of Things (IoT).
- To introduce the fundamental concepts relevant to design issues related to Internet of Things.
- To enable the students to understand the basic principles of mobility in Internet of Things

Course Content

Introduction to IoT: Sensing, Actuation, Communication Protocols, Sensor Networks, IoT architecture, standards considerations. Machine-to-Machine Communications, Devices and gateways, Local and wide area networking, Data management, Business processes in IoT, Everything as a Service (XaaS), M2M and IoT Analytics, Knowledge Management. IoT reference Model, Sensors for IoT Applications, IoT Map Device, Wireless Sensor Structure, Energy Storage Module, Power Management Module, RF Module, Sensing Module, ACOEM Eagle, EnOcean Push Button, NEST Sensor, Ninja Blocks, Wearable Electronics, Implementation of IoT with Raspberry Pi, Clayster libraries, SDN for IoT. Interfacing the hardware: Internal representation of sensor values, Persisting data, External representation of sensor values, Exporting sensor data, development of the actuator project. Security Architecture in the Internet of Thing, RFID False Authentications, Application of Geographical Concepts and Spatial Technology to the Internet of Things: Applying spatial relationships, functions, and models, Interoperability in IoT. Introduction to Arduino Programming, Integration of Sensors and Actuators with Arduino, Case Studies: Agriculture, Healthcare, and Activity Monitoring. Sensor-Cloud, Smart Cities and Smart Homes.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Solve basic network design problems of Internet of Things.
CO2: Define the concept of data communications protocols and convergence of technologies.
CO3: Understand the principles and various research issues related to Internet of Things.

Text Books:

1. Internet of Things Principles and Paradigms by Rajkumar Buyya and Amir Vahid Dastjerdi, Elsevier.
2. From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence by Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stamatis Karnouskos, Stefan Avesand and David Boyle, Academic Press.

Reference Book:

1. Sensors, Actuators and Their Interfaces by N. Ida, Scitech Publishers.

Course Name: **Advance Computer Networks**

Course Code: **CS-740**

Course Type: **Programme Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To impart knowledge about the principles of internetworking.
- To introduce the fundamental concepts relevant to design issues of network layers, various wireless networks concepts, etc.
- To enable the students to understand the basic principles of mobility in wireless networks.

Course Content

Overview and motivation: Resource sharing, Design issues for the network layers Networking devices. Principles of internetworking, Tunneling, Fragmentation, Naming and addressing concepts, Hierarchical naming, Domain name system, Name resolution process, IP address classes and concept of sub netting, Classless Inter-domain routing (CIDR) and DHCP concepts, The internet protocols: IP, ICMP, ARP and RARP. The design issues for the transport layer, addressing, establishing connection, flow control and multiplexing. The internet protocols: TCP and UDP, Multicast routing, Mobility in networks, Mobile IP, Emerging trends in networking.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understand network models and architectures.

CO2: Analyze the performance of various MAC, routing, and transport protocols, and design of new protocol.

CO3: Solve basic network design problems using knowledge of wired and wireless networks.

CO4: Apply knowledge of networking technologies to design a network as per the organization requirements.

Text Books:

1. Data Communications and Networking by B.A. Forouzan, McGraw Hill.

2. Computer Networks by A.S. Tanenbaum, PHI.

Reference Book:

1. Understanding TCP/IP by Libor D. and Alena K, PACKT Publishing.

2. Introduction to Wireless and Mobile Systems by Dharma P. Agrawal and Q-An Zeng, Thomson Learning Inc.

FCourse Name: **Artificial Intelligence**
Course Code: **CS-701**
Course Type: **Institute Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To impart knowledge about Artificial Intelligence.
- To give understanding of the main abstractions and reasoning for intelligent systems.
- To enable the students to understand the basic principles of Artificial Intelligence in various applications

Course Content

Introduction: Overview of AI problems, AI problems as NP, NP-Complete and NP Hard problems. Strong and weak, neat and scruffy, symbolic and sub-symbolic, knowledge-based and data-driven AI. Problem spaces (states, goals and operators), problem solving by search, Heuristics and informed search, Minmax Search, Alpha-beta pruning. Constraint satisfaction (backtracking and local search methods). Knowledge Representation and Reasoning: propositional and predicate logic, Resolution and theorem proving, Temporal and spatial reasoning. Probabilistic reasoning, Bayes theorem. Totally-ordered and partially-ordered Planning. Goal stack planning, Nonlinear planning, Hierarchical planning. Learning: Learning from example, Learning by advice, Explanation based learning, Learning in problem solving, Classification, Inductive learning, Naive Bayesian Classifier, decision trees. Definition of agents, Agent architectures (e.g., reactive, layered, cognitive), Multi-agent systems- Collaborating agents, Competitive agents, Swarm systems and biologically inspired models. Representing and Using Domain Knowledge, Expert System Shells, Explanation, Knowledge Acquisition.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Solve basic AI based problems.
CO2: Define the concept of Artificial Intelligence.
CO3: Apply AI techniques to real-world problems to develop intelligent systems.
CO4: Select appropriately from a range of techniques when implementing intelligent systems.

Text Books:

1. Artificial Intelligence by Elaine Rich, Kevin Knight and Shivashankar B Nair, Tata McGraw Hill.
2. Introduction to Artificial Intelligence and Expert Systems by Dan W. Patterson, Pearson Education.
3. Artificial Intelligence: A Modern Approach by S. Russell and P. Norvig, Prentice Hall.

References Books:

1. A First Course in Artificial Intelligence by Deepak Khemani, McGraw Hill Education.
2. Artificial Intelligence By Example: Acquire advanced AI, machine learning, and deep learning design skills, 2nd Edition.

Course Name: **Machine Learning for Engineers**
Course Code: **CS-702**
Course Type: **Institute Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- Introduce concepts in linear algebra and to use it as a platform for machine learning.
- Introduce concepts of gradient descent, the concepts of constrained and nonlinear optimization
- Introduce the concepts of dimensionality reduction, principal components analysis, machine learning basic and optimization.

Course Content

The Statistical Theory of Machine Learning: Classification, Regression, Aggregation, Regularization, Linear Algebra, Probability and Statistics, Vectors Spaces: - linear independence, basis and rank, Norms, Lengths and distances, Angles and orthogonality, Vector Calculus: - Partial differentiation and gradients, Gradients of vector-valued functions, Gradients of matrices, Backpropagation and automatic differentiation. optimization using gradient descent.
Nonlinear Optimization - Minutiae of Gradient Descent – learning rate decay, initialization, Properties of optimization in learning – typical objective functions, stochastic gradient descent, how optimization in machine learning is different, tuning hyperparameters, importance of feature pre-processing, Challenges in Gradient-based optimization, local optima and flat regions, differential curvature, examples of difficult topologies like cliffs and valleys, momentum-based learning, AdaGrad, RMSProp, Adam. Dimensionality reduction and PCA – problem setting, maximum variance perspective, projection perspective, eigenvector and low-rank approximations, PCA in high dimensions, key steps of PCA in practice, latent variable perspective, Mathematical preliminaries of SVM, primal/dual perspective for SVM, nonlinear SVM - kernels. Vectors Spaces - linear independence, basis and rank, affine spaces, Norms, inner products, Lengths and distances, Angles and orthogonality, Orthonormal basis Machine Learning and Evaluation Method: Introduction to Machine Learning, Problems, data, and tools, Visualization tools, Decision Tree Learning, Artificial Neural Networks, Bayesian Learning, Deep Learning, Instance-Based Learning, Regression Techniques, Linear regression, SSE. data models, data transformations, handling of missing data, time-dependent data, and textual data. Data reduction: feature selection, principal components, smoothing data, case subsampling Evaluation Precision, Recall, F-measure, Measure, Normalized recall, Latent Semantic Indexing, Low rank, approximation, Problems with Lexical Semantics.

Course Outcomes

Upon successful completion of the course, the students will be able to
CO1: Learn concepts of linear algebra that form the foundation of data science problems.
CO2: Learn different optimization methods and apply for machine learning.
CO3: Learn the concepts of dimensionality reduction, and principal components analysis.
CO4: Understand the basics of machine learning algorithms and their evaluation method.

Text Books:

1. Introduction to Machine Learning by Ethem Alpaydin, PHI Learning.
2. Machine Learning: An Algorithmic Perspective by Stephen Marsland, Chapman and Hall/CRC.
3. Machine Learning by Tom Mitchell, McGraw Hill Education.

Reference Books:

1. Pattern Recognition and Machine Learning by Christopher M. Bishop, Springer.
2. Probabilistic Graphical Models by D. Koller, and N. Friedman, MIT Press.

Course Name: **Data Structures & Algorithms**

Course Code: **CS-703**

Course Type: **Institute Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To impart knowledge about basic data structures
- To impart knowledge about the various approaches to design an algorithm.
- To introduce the fundamental concepts relevant to understand the concepts of time and space complexity, worst case, average case and best-case complexities.
- To enable the students to understand the basics of algorithms.

Course Content

Introductions to Data Structures and Algorithms: Algorithm complexity, concept of algorithmic efficiency, run time analysis of algorithms, Asymptotic Notations, Arrays, Linked List, Stack, Queues, Minimum and maximum priority queue, Trees and Graphs (Memory Representation of each and a few basic operations)

Algorithm Design Approaches: Divide and Conquer approach, examples of some sorting techniques like Merge and Quick Sort; Greedy Algorithms; Graph Algorithms: Representation of graphs, BFS, DFS, single source shortest path, all pair shortest path; Dynamic programming: Overview, difference between dynamic programming and divide and conquer, Traveling salesman Problem, longest Common sequence, 0/1 knapsack., Backtracking: 8-Queen Problem, Sum of subsets, graph coloring, Hamiltonian cycles. Computational Complexity: Complexity measures, Polynomial vs non-polynomial time complexity; NP-hard and NP-complete classes and examples.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understand asymptotic notations to analyze the performance of algorithms.

CO2: Understand and apply various problem-solving techniques such as divide and conquer, greedy algorithm, dynamic programming, etc.

CO3: Solve given problem by selecting the appropriate algorithm design technique and justify the selection.

CO4: Know the concepts of P, NP, NP-hard and NP-complete problems.

Text Books:

1. Fundamentals of Computer Algorithms by E. Horowitz and S. Sahni, Galgotia.

2. Introduction to Algorithms by T.H. Cormen, C.E.Leiserson, R.L. Rivest, MIT Press, Cambridge.

3. Data Structure Using C by Aaron M. Tenenbaum, Y. Langsam, M. Augensten, Pearson

Reference Books:

1. The Design and Analysis of Computer Algorithms by A.V. Aho, J.E. Hopcroft and J.D. Ullman, Addison Wesley.

2. Algorithm Design by J. Kleinberg and É. Tardos, Addison-Wesley.

Course Name: **Computer Networks**

Course Code: **CS-704**

Course Type: **Institute Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To impart knowledge about the network models and architectures.
- To introduce the fundamental concepts relevant to performance of various routing protocols and design of new routing protocol.
- To enable the students to understand computers, software, networking technologies and information assurance to an organization's management, operations, and requirements.

Course Content

Introductory Concepts: Goals and Applications of Networks, LAN, WAN, MAN, Network software: Protocol hierarchies, design issues of layers, Interfaces and services. Reference Model: The OSI reference model, TCP/IP reference model, Example networks: Novell Netware, The ARPANET, The Internet, X-25 Networks, network standards. Introduction and basics of Physical Layer, Data Link Layer, Medium access sublayer, network layer, session layer, presentation and application layer. Introduction to Wireless Networks, basic concepts and issues of wireless networks like hidden layer problem etc. wireless technologies like GSM, CDMA, and evolution from 1G to 6G. Type of wireless networks like Ad-hoc networks, MANETS, sensor networks etc.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1 Understand network models and architectures

CO2 Identify the pros and cons of choosing a suitable MAC layer protocol

CO3 Analyze the performance of various routing protocols and design of new routing protocol

CO4: Solve basic network design problems using knowledge of common local and wide area network architectures

CO5: Understanding wireless networks and basic concepts

Text Books:

1. Computer Networks by A.S. Tanenbaum, Prentice Hall of India.
2. Computer Networking: A Top-Down Approach Featuring the Internet by J. Kurose and K.W. Ross, Addison-Wesley.
3. Data and Computer Communication by W. Stallings, Prentice Hall of India.

Reference Books:

1. Network Programmability and Automation, Jason Edelman, 1st edition, O'Reilly.
2. Computer Networks A Top Down Approach, A Forouzan and F Mosharraf, Mc Graw Hill

Course Name: **Programming for Problem Solving**

Course Code: **CS-705**

Course Type: **Institute Elective**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To introduce students to the basic applications, concepts, of different tools used in research.
- To develop skills for using recent software or simulators to solve practical problems in a variety of disciplines.
- To gain experience doing independent study and research.

Course Content

Introduction to Research and Problem solving: Overview of Research in Programming, Importance of research in developing solutions, Fundamentals of Problem-Solving, Importance of defining problems clearly and understanding requirements, Divide and conquer, abstraction, and algorithmic thinking, Research Methods in Programming; Research methodologies applicable to programming-experimental, empirical, and theoretical research.

Foundations of Programming in Python and MATLAB: Introduction to Python and Matlab: Overview of Python and Matlab as a high-level, interpreted programming language, Applications of Python and Matlab in research, data science, and scientific computing, Basic Syntax and Data Types: Introduction to Python and Matlab's basic syntax, Control Structures, Functions and Modular Programming, Importance of modular programming for code organization and reusability, Data Manipulation and Visualization; Object Oriented Programming (OOPs), Introduction to libraries and modules for data manipulation and visualization in Python and Matlab (e.g., NumPy, Pandas, Matplotlib), Error Handling and Debugging. ML libraries and tools: Probabilistic problems, General ML libraries and tools for solving probabilistic problems, preprocessing techniques (Digital Image Processing, such as, gaussian filters, canny edge detection, quantization, thresholding, etc. Natural Language Preprocessing, such as, Stemming, Lemmatization, etc., sound preprocessing, etc.), Logic implementation (Tensorflow, Pytorch, Scikit learn, etc.) (contains algorithms to implement rnn, cnn, lstm, etc. and customise them to bring your own novelty), Visualisation tools - (Matplotlib, seaborn, etc).

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Categorize and carefully differentiate between situations for applying different tools and techniques.

CO2: Design and implement of different problems.

CO3: Evaluate the performance of different algorithms.

CO4: Propose implementation solutions for different applications.

Text Books:

1. Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow by Aurélien Géron.
2. Programming and problem solving with python by Ashok Namdev Kamthane and Amit Ashok Kamthane.
3. Engineering Problem Solving with MATLAB: Second Edition by Delores M. Etter.

Reference Books:

1. Computer Networking Problems and Solutions by Russ White
2. Designing Data-Intensive Applications by Martin Kleppmann

END OF SYLLABUS