NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHEME OF INSTRUCTION AND SYLLABI

Effective from academic year 2019-20

FOR PROGRAM

M.Tech. (Computer Science and Engineering)

NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL

VISION

Towards a Global Knowledge Hub, striving continuously in pursuit of excellence in Education, Research, Entrepreneurship and Technological services to the society.

MISSION

- Imparting total quality education to develop innovative, entrepreneurial and ethical future professionals fit for globally competitiveenvironment.
- Allowing stake holders to share our reservoir of experience in education and knowledge for mutual enrichment in the field of technicaleducation.
- Fostering product oriented research for establishing a self-sustaining and wealth creating centre to serve the societalneeds.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION

Attaining global recognition in Computer Science & Engineering education, research and training to meet the growing needs of the industry and society.

MISSION

- MS1: Imparting quality education through well-designed curriculum in tune with the challenging software needs of theindustry.
- MS2: Providing state-of-art research facilities to generate knowledge and develop technologies in the thrust areas of computer science andengineering.
- MS3: Developing linkages with world class organizations to strengthen industry- academia relationships for mutualbenefit.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

M.TECH IN COMPUTER SCIENCE AND ENGINEERING

PROGRAM EDUCATIONAL OBJECTIVES

PEO1	Design, develop and test software systems for engineering applications.
PEO2	Analyze technical solutions to computational problems and develop efficient algorithms.
PEO3	Work in multi-disciplinary teams to specify software requirements and to achieve project goals.
PEO4	Communicate effectively and demonstrate professional ethics with societal responsibilities.
PEO5	Engage in lifelong learning to keep pace with changing landscape of technologies for professional advancement.

Mapping of Mission statements with program educational objectives

Mission Statement	PEO1	PEO2	PEO3	PEO4	PEO5
Imparting quality education through well-designed curriculum in tune with the challenging software needs of theindustry	3	2	1	1	2
Providing state-of-art research facilities to generate knowledge and develop technologies in the thrust areas of computer science andengineering.	2	2	2		2
Developing linkages with world class organizations to strengthen industry- academia relationships for mutualbenefit	2	2	2	2	1

PROGRAM OUTCOMES: At the end of the program the student will be able to:

PO1	Engage in critical thinking and pursue investigations / research and development to solve practical problems.
PO2	Communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.
PO3	Demonstrate higher level of professional skills to tackle multidisciplinary and complex problems related to Computer Science and Engineering.
PO4	Apply concepts of theoretical computer science to design software systems satisfying realistic, economic, social, safety and security constraints.
PO5	Design and develop processes to meet targeted needs with optimum utilization of resources.
PO6	Develop robust, reliable, scalable techniques and tools for knowledge based systems.

MAPPING OF PROGRAM OUTCOMES WITH PROGRAME EDUCATIONAL OBJECTIVES

PO	PEO1	PEO2	PEO3	PEO4	PEO5
1	2	3	2	-	-
2	1	-	-	3	1
3	2	2	3	-	1
4	3	3	2	3	2
5	3	3	2	3	1
6	2	3	3	2	2

CURRICULAR COMPONENTS

Degree Requirements for M. Tech. in Computer Science and Engineering

Category of Courses	Credits Offered	Min. credits to be earned
Program Core Courses (PCC)	26	26
Departmental Elective Courses (DEC)	≥ 18	18
Seminar, Comprehensive Viva-voce	04	04
Program major Project (PCC)	27	27
Total	**≥ 75**	75

SCHEME OF INSTRUCTION

M.Tech. (Computer Science and Engineering) Course Structure

M.Tech. (CSE) I - Year , I - Semester

S. No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1	CS5101	Advanced Algorithms	3	0	0	3	PCC
2	CS5102	Advanced Operating Systems	3	0	0	3	PCC
3	CS5103	Data Science Fundamentals	3	0	0	3	PCC
4	CS5104	Advanced Software Engineering	3	0	0	3	PCC
5	CS5105	Advanced Operating Systems Lab	0	1	2	2	PCC
6	CS5106	Data Science Lab	0	0	2	1	PCC
7	CS5107	Advanced Software Engineering Lab	0	0	2	1	PCC
8		Elective – 1	3	0	0	3	DEC
9		Elective – 2	3	0	0	3	DEC
10	CS5141	Seminar – 1	0	0	2	1	PCC
		TOTAL	**18**	**1**	**8**	23	

M.Tech. (CSE) I – Year, II - Semester

S. No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1	CS5151	Advanced Computer Networks	3	0	0	3	PCC
2	CS5152	Machine Learning	3	0	0	3	PCC
3	CS5153	Advanced Computer Networks Lab	0	1	2	2	PCC
4	CS5154	Machine Learning Lab	0	1	2	2	PCC
5		Elective – 3	3	0	0	3	DEC
6		Elective – 4	3	0	0	3	DEC
7		Elective – 5	3	0	0	3	DEC
8		Elective – 6	3	0	0	3	DEC
9	CS5191	Seminar – 2	0	0	2	1	PCC
		TOTAL	**18**	**2**	**4**	23	

M.Tech. (CSE) II – Year, I - Semester

S. No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1	CS6142	Comprehensive Viva	0	0	0	2	PCC
2	CS6149	Dissertation Work – Part A	0	0	0	9	PCC
		TOTAL	**0**	**0**	**0**	**11**	

M.Tech. (CSE) II – Year, II - Semester

S. No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1	CS6199	Dissertation Work – Part B	0	0	0	18	PCC
		TOTAL	**0**	**0**	**0**	**18**	

Department Electives

Elective Courses – I year – I Semester			
CS5111	Business Intelligence	3-0-0	3
CS5112	Advanced Databases	3-0-0	3
CS5113	Computer Vision & Image Processing	3-0-0	3
CS5114	Wireless and Mobile Networks	3-0-0	3
CS5115	Game Theory	3-0-0	3
CS5116	Mathematics & Statistics for Data Analytics	3-0-0	3
CS5121	Big Data	3-0-0	3
CS5122	Distributed Computing	3-0-0	3
CS5123	Quantum Computing	3-0-0	3
CS5124	Cryptography and Network Security	3-0-0	3
CS5125	Applied Artificial Intelligence	3-0-0	3
CS5126	Bio-Informatics	3-0-0	3

	Elective Courses – I year - II Semester	L-T-P	Credits
CS5161	Service Oriented Architecture and MicroServices	3-0-0	3
CS5162	Privacy Preserving Data Publishing	3-0-0	3
CS5163	Software Reliability and Quality Management	3-0-0	3
CS5164	Research Study	3-0-0	3
CS5165	Formal Methods in Program Design	3-0-0	3
CS5166	Security and Privacy	3-0-0	3
CS5167	Cognitive Radio Networks	3-0-0	3
CS5168	Model Driven Frameworks	3-0-0	3
CS5169	Exploratory and Interactive Data Analysis	3-0-0	3
CS5170	Internet of Things	3-0-0	3
CS5171	Cloud Computing	3-0-0	3
CS5172	Optimization in Computer Science	3-0-0	3
CS5173	High Performance Computing	3-0-0	3
CS5174	Randomized and Approximation Algorithms	3-0-0	3
CS5175	Human Computer Interaction	3-0-0	3
CS5176	Social Media Analytics	3-0-0	3
CS5177	Models for Social Networks	3-0-0	3
CS5178	Advanced Compiler Design	3-0-0	3
CS5179	Deep Learning	3-0-0	3
CS5180	Natural Language Processing	3-0-0	3
CS5181	Information Retrieval	3-0-0	3
CS5182	Soft Computing Techniques	3-0-0	3
CS5183	Pattern Recognition	3-0-0	3
CS5184	Advanced Data Mining	3-0-0	3

DETAILED SYLLABUS

CS5101	Advanced Algorithms	PCC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Analyze worst-case running times of algorithms using asymptotic analysis
CO2	Prove the correctness of algorithms using inductive proofs and invariants
CO3	Analyze randomized algorithms with respect to expected running time, probability of error using tail inequalities
CO4	Classify problems into different complexity classes corresponding to both deterministic and randomized algorithms
CO5	Analyze approximation algorithms including algorithms that are PTAS and FPTAS

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3		3	3		
CO2	3		3	3		
CO3	3		3	3		
CO4	3		3	3		
CO5	3		3	3		

Detailed syllabus

Algorithm design techniques – Dynamic programming: Matrix chain multiplication, Optimal BST, Greedy algorithms – Shortest path algorithm, MST, Amortized analysis, Data structures for disjoint sets, Divide-and-Conquer- Karatsuba integer multiplication, Large integer multiplications using FFT, NP-Completeness: Poly-time, Poly-time verification, reducibility, NP-Complete problems, Approximation algorithms, Randomized algorithms: Las Vegas and Monte Carlo, Game-Theoretic Techniques: Game Tree Evaluation, The Minimax Principle, Randomness and Non-uniformity, Moments and Deviations: Occupancy Problems, The Markov and Chebyshev Inequalities, Randomized Selection, Two-Point Sampling, The Stable Marriage Problem, The Coupon Collector's Problem, Tail Inequalities: The Chernoff Bound, Routing in a Parallel Computer, A WiringProblem.

Reading:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein,*Introduction to Algorithms*, 2nd Edition, The MIT Press, 2001.
2. C.H. Papadimitriou, *Computational Complexity,* Addison-Wesley, Reading, MA,1994.
3. Rajeev Motwani and Prabhakar Raghavan, *Randomized Algorithms,* Cambridge University Press,1995.
4. Garey Michael R, Johnson Davis S, *Computers and Intractability: A Guide the theory of NP-Incompleteness,* W.H. Freeman &Co.,1979.

CS5102	Advanced Operating Systems	PCC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Design and implement Unix kernel data structures and algorithms
CO2	Analyze synchronization problems in uniprocessor and multiprocessor systems
CO3	Evaluate the scheduling requirements of different types of processes and find their solutions
CO4	Implement user level thread library and mimic the behavior of Unix kernel for scheduling, synchronization and signals

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		3		3	
CO2	2		3		3	
CO3	2		3		3	
CO4	2		3		3	

Detailed syllabus

Introduction to UNIX: The process and the kernel, Mode, space and context, Process abstraction, kernel mode, synchronization by blocking interrupts, process scheduling.

Introduction to Threads: Fundamental abstractions, Lightweight process design, Issues to consider, User level thread libraries, scheduler activationsSignals: Signal generation and handling, Unreliable signals, Reliable signals, Signals in SVR4, Signals implementation, Exceptions, Process Groups

Process Scheduling: Clock interrupt handling, Scheduler Goals, Traditional UNIX scheduling, Scheduling casestudies

Synchronization and Multiprocessing : Introduction, Synchronization in Traditional UNIX Kernels, Multiprocessor Systems, Multiprocessor synchronization issues, Semaphores, spin locks, condition variables, Read-write locks, Reference counts

Introduction to Intel X86 Protected Mode: Privilege Levels, Flat memory model, Descriptors - Segment, Task, Interrupt; GDT, LDT and IDT, Initializing to switch to protected mode operation, Processor Exceptions.

Kernel Memory Allocators: Resource map allocator, Simple power-of-two allocator, McKusick-Karels Allocator, Buddy system, SVR4 Lazy Buddy allocator, OSF/1 Zone Allocator, Hierarchical Allocator, Solaris Slab Allocator

File system interface and framework : The user interface to files, File systems, Special files, File system framework, The Vnode/Vfs architecture, Implementation Overview, File System dependent objects, Mounting a file system, Operations on files.

File System Implementations : System V file system (s5fs) implementation, Berkeley FFS, FFS functionality enhancements and analysis, Temporary file systems, Buffer cache and other special-purpose file systems

Distributed File Systems : Network File System (NFS), Remote File Sharing (RFS)

Advanced File Systems : Limitations of traditional file systems, Sun-FFS, Journaling approach 4.4 BSD, Log-Structured file system, Meta logging Episode FS, Watchdogs, 4.4 BSD portal FS, Stackable FS layers, 4.4 BSD FSinterface.

Reading:

1. Uresh Vahalia, *UNIX Internals*, Pearson Education, 2005.
2. Richard Stevens, Stephen A. Rago, *Advanced Programming in the UNIX Environment*, Pearson Education, 2/e,2005.

CS5103	Data Science Fundamentals	PCC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Apply statistical methods to data for inferences.
CO2	Analyze data using Classification, Graphical and computational methods.
CO3	Understand Data Wrangling approaches.
CO4	Perform descriptive analytics over massive data.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	2	3	2		3
CO2	2	2	3	2		3
CO3	2	2	3	2		3
CO4	2	2	3	2		3

Detailed syllabus

Overview of Random variables and distributions, Statistical learning, Assessing model accuracy, Descriptive Statistics, Dependent and Independent events. Linear Regression: Simple and multiple linear regressions, Comparison of Linear regression with K-nearest neighbors. Simple Hypothesis Testing, Student's t-test, paired t and U test, correlation and covariance, tests for association. Classification: Linear and Logistic Regression, LDA and comparison of classification methods Graphical Analysis: Histograms and frequency polygons, Box-plots, Quartiles, Scatter Plots, Heat Maps Programming for basic computational methods such as Eigen values and Eigen vectors, sparse matrices, QR and SVD, Interpolation by divided differences. Data Wrangling: Data Acquisition, Data Formats, Imputation, The split-apply-combine paradigm. Descriptive Analytics: Data Warehousing and OLAP, Data Summarization, Data de- duplication, Data Visualization using CUBEs.

Reading:

1. Gareth James Daniela Witten Trevor Hastie, Robert Tibshirani, *An Introduction to Statistical Learning with Applications in R*, February 11, 2013, web link: www.statlearning.com (1 to 4chapters)
2. Mark Gardener, *Beginning R The statistical Programming Language*, Wiley,2015.
3. Han , Kamber, and J Pei, *Data Mining Concepts and Techniques*, 3rd edition, Morgan Kaufman, 2012. (Chapter 2 and Chapter4)

CS5104	Advanced Software Engineering	PCC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Apply the Object Oriented Software-Development Process to design software
CO2	Design large-scale, reusable and complex software systems with Design and Architectural patterns
CO3	Develop and apply testing strategies for software applications
CO4	Analyze different Software Reliability parameters using Markovian Models, Finite Failure Category Models and Infinite Failure category Models
CO5	Design and Plan software solutions to security problems using various paradigms

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	3	2	2	2	2
CO2	2	3	2	2	2	2
CO3	2	3	2	2	2	2
CO4	2	3	2	2	2	3

Detailed Syllabus

Introduction and System Engineering: Introduction, Software Process and Methodology, System Engineering.Analysis and Architectural Design : Software Requirement Elicitation, Domain Modeling, Architectural Design.Modeling and Design of Interactive Systems and Other Types of Systems : Deriving Use Cases from Requirements, Actor-System Interaction Modeling, Object Interaction Modeling, Applying Responsibility-Assignment Patterns, Deriving a design class diagram, User Interface Design. Object State modeling of Event-Driven Systems, Activity Modeling for Transformational Systems.Implementation and Quality Assurance: Implementation Considerations, Software Quality Assurance, Software Testing.
Software Reliability Modeling: Markovian Models, Finite Failure Category Models, Infinite Failure Category Models. Comparison of Software Reliability Models.
Project Management and Software Security: Software Project Management, Software Security.

Reading:

1. Kung, David. Object-oriented software engineering: an agile unified methodology. McGraw-Hill Higher Education, 2013.
2. Gamma, Erich. Design patterns: elements of reusable object-oriented software, Pearson Education India, 1995.
3. M. Xie, *Software Reliability Modelling*, *World Scientific*; 1991.
4. John D. Musa, Anthony Iannino, Kazuhira Okumoto, *Software Reliability Measurement, Prediction, Application.McGraw-Hill Book Company*; 1987.

CS5105	Advanced Operating Systems Lab	PCC	0 – 1 – 2	2 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Implement basic/UNIX kernel level algorithms.
CO2	Implement the user level thread library and mimic the behavior of UNIX kernel for scheduling, synchronization and signals.
CO3	Implement File system image in a file and NFS using RPC.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		2	1	2	
CO2	2		2	2	2	
CO3	2		2	2	2	

Detailed syllabus

1. Write Command Interpreter Programs which accepts some basic Unix commands and displays the appropriate result. Each student should write programs for at least six commands.
2. Study the concept of Signals and write a program for Context Switching between two processes using alarmsignals.
3. Study pthreads and implement the following: Write a program which shows the performance improvement in using threads as compared with process.(Examples like Matrix Multiplication, Hyper quicksort, Merge sort, Traveling Sales Person problem)
4. Create your own thread library, which has the features of pthread library by using appropriate system calls (UContext related calls). Containing functionality for creation, termination of threads with simple round robin scheduling algorithm and synchronization features.
5. Implement all CPU Scheduling Algorithms using your threadlibrary
6. Study the concept of Synchronization and implement the classical synchronization problems using Semaphores, Message queues and shared memory (minimum of 3 problems)
7. A complete file system implementation inside a disk imagefile.
8. NFS server and NFS client implementation usingRPC

Reading:

1. *Richard Stevens, Stephen A. Rago, Advanced Programming in the UNIX Environment, Pearson Education, 2/e, 2005.*

CS5106	Data ScienceLab	PCC	0 – 0 – 2	1 Credit

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Demonstrate skills acquired in R Programming.
CO2	Interpret models in data using statistical analysis.
CO3	Prepare environment for distributed systems applications.
CO4	Write Programs for big data using Map Reduce.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	1	1	2	3		3
CO2	2	1	2	1		3
CO3	2	1	2	2		3
CO4	3	2	2	2		3

Detailed syllabus

1. Introduction to R: Installing R in windows, R Console (R window to edit and execute R Commands), Commands and Syntax (R commands and R syntax), Packages and Libraries (Install and load a package in R), Help In R, Workspace inR.
2. Familiarity of Data Structures in R: Introduction to Data Types (Why Data Structures?, Types of Data Structures in R), Vectors, Matrices, Arrays, Lists, Factors, Data Frames, Importing and ExportingData.
3. Graphical Analysis: Creating a simple graph (Using plot() command), Modifying the points and lines of a graph (Using type, pch, font, cex, lty, lwd, col arguments in plot() command), Modifying Title and Subtitle of graph (Using main, sub, col.main, col.sub, cex.main, cex.sub, font.main, font.sub arguments in plot() command), Modifying Axes of a Graph (Using xlab, ylab, col.lab, cex.lab, font.lab, xlim, ylim, col.axis, cex.axis, font.axis arguments and axis() command), Adding Additional Elements to a Graph (Using points(), text(), abline(), curve() commands), Adding Legend on a Graph (Using legend() command), Special Graphs (Using pie(), barplot(), hist() commands), Multiple Plots (Using mfrow or mfcol arguments in par() command and layoutcommand).
4. Descriptive Statistics: Measure of Central Tendency (Mean, Median and Mode), Measure of Positions (Quartiles, Deciles, Percentiles and Quantiles), Measure of Dispersion (Range, Median, Absolute deviation about median, Variance and Standard deviation), Measure of Distribution (Skewness and Kurtosis), Box and Whisker Plot (Box Plot and its parts, Using Box Plots to comparedistribution).
5. Comparing Population: Test of Hypothesis (Concept of Hypothesis testing, Null Hypothesis and Alternative Hypothesis), Cross Tabulations (Contingency table and their use, Chi-Square test, Fisher's exact test), One Sample t test (Concept, Assumptions, Hypothesis, Verification of assumptions, Performing the test and interpretation of results), Independent Samples t test (Concept, Type, Assumptions, Hypothesis, Verification of assumptions, Performing the test and interpretation of results), Paired Samples t test (Concept, Assumptions, Hypothesis, Verification of assumptions, Performing the test and interpretation of results), One way ANOVA (Concept, Assumptions, Hypothesis, Verification of assumptions, Model fit, Hypothesis testing, Post hoc tests: Fisher's LSD, Tukey'sHSD).
6. Experiments based on Linear Regression and Multiple RegressionMethods.
7. Set up and practice examples on Hadoop2.0;
8. Map Reduce implementation for Relational algebraoperations
9. Map reduce implementation for matrix multiplication.

Reading:

1. Mark Gardener, Beginning R The statistical Programming Language, Wiley, 2015.

CS5107	Advanced Software Engineering Lab		0 – 0 – 2	1 Credit

Course Outcomes: At the end of the course the student will be able to:

CO1	Capture and document the software requirements for complex engineering problems.
CO2	Collaborate with team members to establish goals, plan tasks and meet objectives using tools and techniques.
CO3	Design and implement solutions to the recursive problems.
CO4	Develop and conduct appropriate experiments, analyse and interpret data, and use engineering judgment to draw conclusions
CO5	Test the system compliance using various test cases
CO6	Determine software reliability parameters using testing data

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	3	3		1
CO2	2	2	3	3		2
CO3	2	2	3	2		2
CO4	2	2	3	3		2
CO5	2	2	2	2		1
CO6	2	2	2	2		2

Syllabus:

1. Develop a Software Project Plan Using Microsoft Project
   (Planning involves estimation – work breakdown structure – Budget Estimation – Effort Estimation – Resource Estimation – Perform Earned Value Analysis on software engineering tasks for a system or product)
2. Develop an SRS Document uses Rational Requisite Pro Tool.
   (Documenting the SRS along with Vision and Scope, Use Cases and Misuscases).
3. Design a solution for recursive problems (Creational, Structural and Behavioural Design Patterns).
4. Use of Agile methods for software development.
5. Write a program for the following : Quality Metrics and OO Metrics, coupling and cohesion.
6. Implement the threat modeling for the security applications.
7. Define test cases with the help of uses cases and verify the system compliance.
8. Case studies on Different Testing Tools to carry out the Functional Testing, Load/Stress testing. Perform automate testing using automated testing tool.
9. Apply different software reliability models to detemine reliability parameters using testing data.
10. Implement ISO/IEC 25020 Quality Standard for Software Applictions

Reading

1. Kung, David. Object-oriented software engineering: an agile unified methodology. McGraw-Hill Higher Education, 2013.
2. John D. Musa, Anthony Iannino, Kazuhira Okumoto, *Software Reliability Measurement, Prediction, Application.McGraw-Hill Book Company*; 1987.
3. Dr. K.V.K.K. Prasad , Software Testing Tools : Covering WinRunner, SilkTest, LoadRunne, JMeter, TestDirector and QTP with case studies, Dreamtech Press.
4. Rodney C. Wilson, Unix Test Tools and Benchmarks

CS5151	Advanced Computer Networks	PCC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Analyze computer network architectures and estimate quality of service
CO2	Design application level protocols for emerging networks
CO3	Analyze TCP and UDP traffic in data networks
CO4	Design and Analyze medium access methods, routing algorithms and IPv6 protocol for datanetworks
CO5	Analyze Data Center Networks and Optical Networks including next generation cellular networks

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	1	2	3	2	
CO2	2	1	2	2		
CO3	1		2	2		
CO4	1		2	2	1	
CO5	1		2	2	1	

Detailed syllabus

The internet architecture, Access Networks, The network Core, Peer-to-Peer Networks, Content Distribution Networks, Delay Tolerant Networks, Circuit Switching vs. Packet switching, Packet switching Delays and congestion, Client/Server and Peer-to-Peer Architectures, HTTP and HTTPS, FTP and SFTP, Domain Name Service, TCP and UDP sockets, MAC and LLC, Virtual LAN, Asynchronous Transfer Mode (ATM), Network Address Translator, Internet Control Message Protocol, SNMP, CIDR, IPv6, Routing Protocol Basics in advanced networks, Routing Information Protocol (RIP), Interior Gateway Routing Protocol (IGRP), Switching Services, Spanning Tree Protocol (STP), Standard Network Management Protocol, TCP and Mobile TCP, TCP Tahoe and TCP Reno, High speed TCP, Coexistence of UDP and TCP flows, Introduction to traffic Engineering, Requirement Definition for Traffic Engineering, Traffic Sizing, Traffic Characteristics, Delay Analysis, Connectivity and Availability, Introduction to Multimedia Services, Explaining Transmission of Multimedia over the Internet, Explaining IP Multicasting, VOIP, UnifiedCommunication, Virtual Networking, Data center Networking, Introduction to Optical Networking, SONET / SDH Standard, Next generation cellular networks, Secure Socket Layer, IP Sec, TLS, Kerberos, Domain name system Protection.

Reading:

1. Larry L. Peterson and Bruce S. Davie, *Computer Networks: A systems approach*, Morgan Kaufman, 5th Edition,2012.
2. Chwan-Hwa (John) Wu, J. David Irwin, *Introduction to computer networks and Cyber Security*, CRC press, Taylor & Francis Group,2014
3. Andrew S. Tanenbaum, David J. Wetherall, *Computer Networks*, Pearson, 5th Edition, 2014.
4. G. Wright and W. Stevens, *TCP/IP Illustrated*, Volume 1 and Volume 2, Addison-Wesley, 1996.

CS5152	Machine Learning	PCC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Identify instance based learning algorithms
CO2	Design Multi-Layer neural network to solve Supervised Learning problems
CO3	Use Radial Basis Functions for classification problems.
CO4	Apply Genetic Algorithm for optimization problems.
CO5	Understand the deep learning architectures which are appropriate for various types of learning tasks.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		1	1	2	
CO2	2	2	2	2	1	2
CO3	3	2	1	2	2	
CO4	2	3		2	1	
CO5	2	0	2		1	2

Detailed syllabus

INTRODUCTION – Well defined learning problems, Designing a Learning System, Issues in Machine Learning; - THE CONCEPT LEARNING TASK - General-to-specific ordering of hypotheses, Find-S, List then eliminate algorithm, Candidate elimination algorithm, Inductive bias - DECISION TREE LEARNING - Decision tree learning algorithm-Inductive bias- Issues in Decision tree learning; - ARTIFICIAL NEURAL NETWORKS – Perceptron, Gradient descent and the Delta rule, Adaline, Multilayer networks, Derivation of back-propagation rule-Back-propagation Algorithm- Convergence, Generalization; – EVALUATING HYPOTHESES – Estimating Hypotheses Accuracy, Basics of sampling Theory, Comparing Learning Algorithms; - - COMPUTATIONAL LEARNING THEORY – Sample Complexity for Finite Hypothesis spaces, Sample Complexity for Infinite Hypothesis spaces, The Mistake Bound Model of Learning; - INSTANCE-BASED LEARNING – k-Nearest Neighbor Learning, Locally Weighted Regression, Kernel Methods - Dual Representations, Constructing Kernels, Radial basis function networks, Gaussian Processes for Regression and Classification;Case-based learning - GENETIC ALGORITHMS – an illustrative example, Hypothesis space search, Genetic Programming, Models of Evolution and Learning; Sparse Kernel Machines- Maximum Margin Classifiers, Multi class SVMs, REINFORCEMENT LEARNING - The Learning Task, Q Learning, Nondeterministic rewards and actions, Temporal difference learning, Generalizing from examples, relationship to Dynamic Programming.INTRODUCTION TO DEEP LEARNING –Introduction to Perceptron, Multilayer networks, Back-propagation Algorithm,Deep Feed Forward network, regularizations, training deep models, dropouts, Convolutional Neural Networks (CNNs), Autoencoders.APPLICATIONS TO NATURAL LANGUAGE PROCESSING AND VISION ANALYTICS – Object recognition, sparse coding, computer vision, natural language processing.

Reading:

1. Tom M .Mitchell, *Machine Learning*, McGraw Hill,1997.
2. Christopher Bishop, *Pattern Recognition and Machine Learning*, Springer,2006.
3. Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, 2016.

CS5153	Advanced Computer Networks Lab	PCC	0 – 1 – 2	2 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Develop programs for client-server applications
CO2	Perform packet sniffing and analyze packets in network traffic.
CO3	Implement error detecting and correcting codes
CO4	Implement network security algorithms

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		2	2	2	
CO2	2		2	2	2	
CO3	2		2	2	2	
CO4	2		2	3	2	

Detailed syllabus

Assignment-1 Connection oriented Client server applications with TCP
Assignment-2 Connectionless Client server applications with UDP
Assignment-3 Programs using RPC remote procedure call
Assignment-4 client server applications using concurrent server
Assignment-5 client server applications using Multi-protocol server
Assignment-6 client server applications using super server
Assignment-7 Implement a chat and mail server
Assignment-8 Implement network security mechanisms

Reading:

1. W. Richard Stevens, UNIX Network Programming, Volume 1, Second Edition: Networking APIs: Sockets and XTI, Prentice Hall, 1998
2. W. Richard Stevens, UNIX Network Programming, Volume 2, Second Edition: Inter-process Communications, Prentice Hall, 1999
3. W. Richard Stevens, Stephen Rago, Advanced Programming in the UNIX Environment, Pearson Education, 2/e

CS5154	Machine Learning Lab	PCC	0 – 1 – 2	2 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Implement Adaline and use for playing 2 player games
CO2	Build neural network to solve classification problems
CO3	Build optimal classifiers using genetic algorithms
CO4	Develop Perceptron for linearly separable problems
CO5	Develop a Convolutional Neural Network for object recognition

Mapping of course outcomes with program outcomes:

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1		**2**		**2**	**1**	
CO2	**2**	**2**	**1**	**2**	**2**	
CO3	**2**	**3**	**2**	**2**	**2**	
CO4			**1**	**1**		
CO5	**2**		**2**	**1**	**1**	**2**

Detailed syllabus

1. Design and implement machine learning algorithm using least means square learning rule to play checkers game. The training experience should be generated by the system playing game withitself.
2. Implement a machine learning program to play 5× 5 Tic tac toegame.
3. Design and implement a feed forward neural network with 5 inputs, 3 hidden and 1 output units. It should use back-propagation algorithm with batch update to train the neural network to generate odd parity bit on its output given any 5 bit binary pattern on itsinputs.
4. Construct decision tree for the training examples given in following table for Playtennis domain using ID3 algorithm. Target attribute is*Playtennis.*

Outlook	**Temp**	**Humidity**	**Windy**	***Playtennis***
Sunny	75	70	true	play
Sunny	80	90	true	no play
Sunny	85	85	false	no play
Sunny	72	95	false	no play
Sunny	69	70	false	play
Overcast	72	90	true	play
Overcast	83	78	false	play
Overcast	64	65	true	play
Rainy	81	75	false	play
rainy	71	80	true	no play
rainy	65	70	true	no play
rainy	75	80	false	play
rainy	68	80	false	play

5. Implement perceptron learning algorithm and attempt to solve two input i) AND gate ii) Or Gate iii) EXOR gateproblems.
6. Implement the Gabil's method of using genetic algorithm to obtain the classifier for the 2 input EXORgate.
7. Design and implement genetic algorithm to learn conjunctive classification rules for the ***Play-golf*** problem described in followingtable.

Outlook	***Temperature***	***Humidity***	***Wind***	***PlayGolf***
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes

8. Implement the Candidate-Elimination Algorithm on followingData

Sky	*Air Temp*	*Humidity*	*Wind*	*Water*	*Forecast*	*Enjoy sport*
Sunny	*warm*	*Normal*	*light*	*warm*	*same*	*yes*
Sunny	*Warm*	*High*	*strong*	*cool*	*change*	*yes*
Rainy	*Cold*	*High*	*Strong*	*Warm*	*Change*	*No*
Sunny	*Warm*	*High*	*Strong*	*Warm*	*Same*	*Yes*
Sunny	*Warm*	*Normal*	*Strong*	*Warm*	*Same*	*yes*

9. Implement a Convolutional Neural Network for hand written digit recognition, for digits 0 to 9.

Reading:

1. Tom M. Mitchell, *Machine Learning*, McGraw Hill, 1997.
2. SS Shwartz and SB David, Undersatanding Machine Learning from theory to algorithms, Cambridge University Press, 2018
3. *Nikhil Buduma,* Fundamentals of Deep Learning *Designing Next-Generation MachineIntelligence Algorithms, Oreilly, 2017.*

CS5141	**Seminar-1**	**PCC**	**0 – 0 – 2**	**1 Credit**
CS5191	**Seminar-2**	**PCC**	**0 – 0 – 2**	**1 Credit**

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Analyze the selected topic, organize the content and communicate to audience in an effective manner
CO2	Practice the learning by self study

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	3	1	1		
CO2	2	3	1	1		

ELECTIVE COURSES

CS5111	Business Intelligence	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Identify the need for data warehouse for large organizations.
CO2	Determine the data sources to populate data warehouse.
CO3	Design Data warehouse models using appropriate schemas.
CO4	Develop Data warehouse for a domain using Data warehouse tools.
CO5	Design Data warehouse to meet business objectives.
CO6	Apply data analysis techniques for building Decision Support System.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2				2	3
CO2				2		3
CO3	2		2	2	2	3
CO4	2		2	2		3
CO5	2		2	2	2	3
CO6	2		2	3		3

Detailed syllabus

Business Intelligence (BI): Introduction – Definition, Leveraging Data and Knowledge for BI, BI Components, BI Dimensions, Information Hierarchy, Business Intelligence and Business Analytics. BI Life Cycle. Data for BI - Data Issues and Data Quality for BI, BI Implementation - Key Drivers, Key Performance Indicators and operational metrics, BI Architecture/Framework, Best Practices, Business Decision Making.; Business Analytics – Objective Curve, Web Analytics and Web Intelligence, Customer Relationship Management. Business/Corporate Performance Management - Dash Boards and Scorecards, Business Activity Monitoring, Six Sigma; Advanced BI – Big Data and BI, Social Networks, Mobile BI, emerging trends.; Working with BI Tools – Pentaho etc.; Overview of managerial, strategic and technical issues associated with Business Intelligence and Data Warehouse design, implementation, and utilization. Critical issues in planning, physical design process, deployment and ongoingmaintenance.

Data Warehousing (DW):

Introduction and Overview, Data Marts, DW architecture – DW components, Implementation options, Meta Data, Information delivery, ETL - Data Extraction, Data Transformation – Conditioning, Scrubbing, Merging, etc., Data Loading, Data Staging, Data Quality, Dimensional Modeling - Facts, dimensions, measures, examples, Schema Design – Star and Snowflake, Fact constellation, Slow changing Dimensions, OLAP - OLAP Vs OLTP, Multi-Dimensional Databases (MDD), OLAP – ROLAP, MOLAP, HOLAP, Data Warehouse Project Management - Critical issues in planning, physical design process, deployment and ongoing-maintenance.

Reading:

1. Efraim Turban, Ramesh Sharda, Jay Aronson, David King, *Decision Support and Business Intelligence Systems*, 9th Edition, Pearson Education,2009.
2. David Loshin, *Business Intelligence - The Savy Manager's Guide Getting Onboard with Emerging IT*, Morgan Kaufmann Publishers,2009.
3. Paulraj Punniah, *Data Warehousing Fundamentals: A comprehensive guide for IT professionals*, John Wiley publications,2001.

CS5112	Advanced Databases	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Design distributed database for application development.
CO2	Apply query optimization principles for optimizing query performance in centralized and distributed database systems
CO3	Design distributed database schema using principles of fragmentation and allocation.
CO4	Apply distributed transaction principles for handling transactions in distributed database applications.
CO5	Apply distributed database administration principles for managing distributed database.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		2	1	1	2
CO2	2		3	1	2	2
CO3	2		3	1		2
CO4	2		3	2		2
CO5	2		3	2	2	2

Detailed syllabus

Distributed Databases: Introduction to Distributed Database Systems, Distributed Database System Architecture; Top-Down Approach, Distributed Database Design Issues, Fragmentation, Allocation, Database Integration, Bottom-up approach, Schema Matching, Schema Integration, Schema Mapping; Data and Access Control, View Management, Data Security; Query processing problem, Objectives of Query processing, Complexity of Relational Algebra Operations, Characterization of Query Processors, Layers of Query Processing; Query Decomposition, Normalization, Analysis, Elimination of Redundancy and Rewriting; Localization of Distributed Data, Reduction for primary Horizontal, Vertical, derived Fragmentation; Distributed Query Execution, Query Optimization, Join Ordering, Static& Dynamic Approach, Semi-joins, Hybrid Approach; Taxonomy of Concurrency control Mechanisms, Lock-Based Concurrency Control, Timestamp-Based Concurrency Control, Optimistic Concurrency Control, Deadlock Management; Heterogeneity issues Advanced Transaction Models, Distributed systems 2PC& 3PC protocols, Replication protocols, Replication and Failures, HotSpares;
Parallel Databases: Introduction to Parallel Databases, Parallel Database System Architectures, Parallel Data Placement, Full Partitioning; Parallel Query Processing, Query Parallelism; Parallel Query Optimization, Search Space, Cost Model, Search Strategy; Load Balancing.

Reading:

1. M T Ozsu, Patrick Valduriez, *Principles of Distributed Database Systems*, Prentice Hall, 1999.
2. S. Ceri and G. Pelaggati, *Distributed Database System Principles and Systems*, MGH, 1985.

CS5113	Computer Vision & Image Processing	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Classify Image representations
CO2	Apply Image transformation methods
CO3	Implement image processing algorithms
CO4	Design face detection and recognition algorithms
CO5	Recover the information, knowledge about the objects in the scene

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	1					
CO2	2		2			1
CO3	2		2			1
CO4	2		2	2		1
CO5	2		2	2		2

Detailed syllabus

The image model and acquisition, image shape, sampling, intensity images, color images, range images, image capture, scanners. Statistical and spatial operations, Gray level transformations, histogram equalization, multi image operations. Spatially dependent transformations, templates and convolution, window operations, directional smoothing, other smoothing techniques. Segmentation and Edge detection, region operations, Basic edge detection, second order detection, crack edge detection, edge following, gradient operators, compass & Laplace operators. Morphological and other area operations, basic morphological operations, opening and closing operations, area operations, morphological transformations. Image compression: Types and requirements, statistical compression, spatial compression, contour coding, quantizing compression. Representation and Description, Object Recognition, 3-D vision and Geometry, Digital Watermarking. Texture Analysis.

Reading:

1. D. A. Forsyth, J. Ponce, *Computer Vision: A Modern Approach*, PHI Learning,2009.
2. Milan Soanka, Vaclav Hlavac and Roger Boyle, *Digital Image Processing and Computer Vision*, Cengage Learning, 2014
3. R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Pearson Education,2007

CS5114	Wireless and Mobile Networks	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Identify issues related to environment, communication, protocols in mobile computing
CO2	Evaluate the performance of mobile IPv4 and IPv6 architectures
CO3	Analyze the performance of MAC protocols for wired and wireless networks
CO4	Analyze the performance of transport layer protocols in mobile Ad-hoc networks
CO5	Design and analyze routing protocols for multi-hop wireless networks

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		2	2	2	
CO2	2		3	2	2	
CO3	2		3	2		
CO4	2		3	2	2	
CO5	2		2	2	2	

Detailed syllabus

Basic communication Technologies, Introduction to Mobile Networks, Types of Wireless networks (MANET: Mobile ad-hoc networks, WSN: Wireless Sensor Networks, VANET: Vehicular Ad-hoc Networks, PAN: Personal Area Networks, DTN: Delay Tolerant Network), Wireless Communication Fundamentals, Cellular Wireless Networks, Mobile Ad-hoc Networks, Medium Access Control Layer: MACA, MACAW, Wireless LAN, Mobile Network Layer (Mobile IP), DHCP, Routing in Mobile Ad hoc Networks (MANET): AODV (Ad-hoc On-Demand Distance Vector Routing Protocol), DSR (Dynamic Source Routing), Secure routing protocols in MANET, Wireless Sensor Networks: (Routing protocols, Localization methods, Sensor Deployment Strategies), Delay Tolerant Networks, Vehicular Ad-hoc Networks, Wireless Access Protocol, GPS, RFID.

Reading:

1. C D M Cordeiro, D. P. Agarwal, *Adhoc and Sensor Networks: Theory and applications*, World Scientific,2006.
2. Jochen Schiller, *Mobile Communications*, Second Edition, Pearson Education,2003.
3. Asoke K Talukder and Roopa R. Yavagal, *Mobile Computing – Technology, Applications and Service Creation*, TMH Pub., New Delhi,2006.

CS5115	Game Theory	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Analyze games based on complete and incomplete information about the players
CO2	Analyze games where players cooperate
CO3	Compute Nash equilibrium
CO4	Apply game theory to model network traffic
CO5	Analyze auctions using game theory

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		3	2	2	
CO2	2		3	2	2	
CO3	2		3	2	2	
CO4	2		3	2	2	
CO5	2		3	2	2	

Detailed syllabus

Games, Old and New; Games, Strategies, Costs, and Payoffs; Basic Solution Concepts Finding Equilibria and Learning in Games; Refinement of Nash: Games with turns and Subgame Perfect Equilibrium; Nash Equilibrium without Full Information: Bayesian Games; Cooperative Games, Markets and Their Algorithmic Issues; Is the NASH-Equilibrium Problem NP-Complete?; The Lemke-Howson Algorithm; The Class PPAD. Succinct Representations of Games; The Reduction; Correlated Equilibria; Bitmatrix Games and Best Response Condition; Equilibria Via Labeled Polytopes; The Lemke-Howson Algorithm; Integer Pivoting and Degenerate Games; Extensive Games and Their Strategic Form; Sub game Perfect Equilibria; Computing Equilibria with SequenceForm.

Model and Preliminaries; External Regret Minimization; Regret minimization and Game Theory; Generic Reduction from External to Swap Regret; On the Convergence of Regret-Minimizing Strategies to Nash Equilibrium in Routing Games; Fisher's Linear Case and the Eisenberg –Gale Convex Program; Checking if Given Prices are Equilibrium Prices; Two Crucial Ingredients of the Algorithm; The Primal-Dual Schema in the Enhanced Setting; Tight Sets and the Invariants; Balanced Flows; The Main Algorithm and Running Time; The Linear-Case of Arrow-Debreu Model; Algorithm for Single-Source Multiple-Sink Markets; Fisher Model with Homogeneous Consumers; Exchange Economics Satisfying WGS; Specific Utility Functions; Computing Nash Equilibria in Tree Graphical Games; Graphical Games and Correlated Equilibria; Graphical Exchange Economies.

Reading:

1. Noam Nisan, Tim Roughgarden, Eva Tardos, Vijay V. Vazirani, *Algorithmic Game Theory*, Cambridge University Press,2007.
2. Ronald Cohn Jesse Russell, *Algorithmic Game Theory*, VSD Publishers,2012.

CS5116	Mathematics & Statistics for Data Analytics	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Apply Mathematical concepts for Data Analytics.
CO2	Apply Statistical concepts for Data Analytics.
CO3	Analyze topic models hidden in the data.
CO4	Interpret data in high dimensional space.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2			2
CO2	3	2	2			2
CO3	3	2	3			2
CO4	3	2	2			2

Detailedsyllabus

Generating Functions, Applications of Mean value theorem, Linear Algebra, Eigen Values and Eigen Vectors, Sums of Series, Useful Inequalities, Probability, Probability Distributions, Maximum Likelihood Estimation, Tail bounds, Chernoff Bounds, Variational Methods, Hash Functions, High Dimensional Space Properties, Random Graphs, Singular Value Decomposition. Random Walks and Markov Chains. Topic Models, Hidden Markov Process, GraphicalModels.

Reading:

1. John Hopcroft, Ravindran Kannan, Foundations of Data Science, http://www.cs.cornell.edu/jeh/nosolutions90413.pdf

CS5121	Big Data	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Analyze big data challenges in different domains including social media, transportation, finance andmedicine
CO2	Explore relational model, SQL and capabilities of emergent systems in terms of scalability and performance
CO3	Apply machine learning algorithms for data analytics
CO4	Analyze the capability of No-SQL systems
CO5	Analyze MAP-REDUCE programming model for better optimization

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	3	2		3
CO2	3		3	2		3
CO3	3		3	2		3
CO4	2		3	2		3
CO5	2		3	2	2	3

Detailed syllabus

Overview of Big Data, Map Reduce basics, Overview of Hadoop, Map Reduce Algorithm Design, Inverted Indexing for Text Retrieval, Graph Algorithms, Data Mining with Big Data, No SQL databases, Stream Computing Challenges, Stages of analytical evolution, Big Data Analytics in Industry Verticals, Data Analytics Lifecycle, Operationalizing Basic Data Analytic Methods Using R, Analytics for Unstructured Data.

Reading:

1. Bill Franks, *Taming The Big Data Tidal Wave*, 1st Edition, Wiley,2012.
2. Jure Leskovec, Anand Rajaraman, J D Ullman, Mining MassiveDatasets.
3. Jimmy Lin and Chris Dyer, Data Intensive Text Processing with Map Reduce, Pre- production manuscript, Downloadable fromInternet.
4. Johannes Ledolter, Data Mining and Business Analytics with R, Wiley,2013.

CS5122	Distributed Computing	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Identify models of distributed computing
CO2	Analyze algorithms for coordination, communication, security and synchronization in distributed systems
CO3	Classify distributed shared memory models
CO4	Design and Implement distributed file systems
CO5	Design distributed algorithms for handling deadlocks

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		1	2	2	
CO2	2		2	2	2	
CO3	2		2	2	2	
CO4	2		2	2	2	
CO5	2		2	2	2	

Detailed syllabus

Distributed Computing Introduction: Types of distributed systems, synchronous vs. asynchronous execution, design issues and challenges.A Model of Distributed Computations: A Model of distributed executions, Global state of distributed system, Models of process communication.

Logical Time: Logical clocks, scalar time, vector time, Efficient implementation of vector clocks, Jard-Jourdan's adaptive technique, Matrix time, virtual time, Physical clock synchronization: NTP, Global state and snapshot recording algorithms: System model, Snapshot algorithms for FIFO channels, Variations of Chandy-Lamport algorithm, Snapshot algorithms for non-FIFO channels, Snapshots in a causal delivery system, Monitoring global state, Necessary and sufficient conditions for consistent global snapshots, Finding consistent global snapshots in a distributed computation.

Message ordering and group communication: Message ordering paradigms, Group communication, Causal order (CO), Total order, Propagation trees for multicast, Semantics of fault-tolerant group communication, Distributed multicast algorithms

Termination detection : Introduction, System model of a distributed computation, Termination detection using various methods and algorithms, Termination detection in a faulty distributed system, Distributed mutual exclusion algorithms: Lamport's algorithm, Ricart–Agrawala algorithm, Singhal's dynamic information-structure algorithm, Lodha and Kshemkalyani's fair mutual exclusion algorithm, Quorum-based mutual exclusion algorithms, Maekawa's algorithm, Agarwal–El Abbadi quorum-based algorithm, Token-based algorithms, Suzuki–Kasami's broadcast algorithm, Raymond's tree-based algorithm.

Deadlock detection in distributed systems: System model, Knapp's classification of distributed deadlock detection algorithms, Mitchell and Merritt's algorithm for the single resource model, Chandy–Misra–Haas algorithm for the AND and the OR model, Kshemkalyani–Singhal algorithm for the P-out-of-Qmodel.

Distributed shared memory: Abstraction and advantages, Memory consistency models, Shared memory mutual exclusion, Wait-freedom, Register hierarchy and wait-free simulations, Wait-free atomic snapshots of shared objects, Check pointing and rollback recovery : Issues in failure recovery, Checkpoint-based recovery, Log-based rollback recovery, Koo–Toueg coordinated checkpointing algorithm, Juang–Venkatesan algorithm for asynchronous check pointing and recovery, Manivannan– Singhal quasi-synchronous check pointing algorithm, Peterson–Kearns algorithm based on vector time, Helary–Mostefaoui–Netzer–Raynal communication-induced protocol.
Consensus and agreement algorithms : Problem definition, Agreement in a failure-free system, Agreement in systems with failures, Wait-free shared memory consensus in asynchronous systems
Failure detectors: Unreliable failure detectors, The consensus problem, Atomic broadcast, The weakest failure detectors, to solve fundamental agreement problems An adaptive failure detection protocol.

Reading:

1. Ajay D. Kshemakalyani, Mukesh Singhal, *Distributed Computing,* Cambridge University Press,2008.
2. Andrew S. Tanenbaum, Maarten Van Steen, *Distributed Systems - Principles and Paradigms,* PHI,2004.

CS5123	Quantum Computing	DEC	3 – 0 – 0	3 Credits

Prerequisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Understand quantum computation
CO2	Understand Hilber space, entanglement and basics of quantum mechanics
CO3	Compare between classical and quantum information theory.
CO4	Demonstrate quantum algorithms such as Shor's and Grover's
CO5	Analyze quantum algorithms including Deutsch's algorithm and Deutsch's-Jozsa algorithm
CO6	Design Quantum error correction and fault-tolerant computation approaches

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		1	2		
CO2	2		1	2		
CO3	2		2	2		
CO4	2		2	2		
CO5	2		2	2		
CO6	2		2	2		

Detailed Syllabus

Introduction to Quantum Computation: Quantum bits, Bloch sphere representation of a qubit, multiple qubits. Background Mathematics and Physics: Hilber space, Probabilities and measurements, entanglement, density operators and correlation, basics of quantum mechanics, Measurements in bases other than computational basis. Quantum Circuits: single qubit gates, multiple qubit gates, design of quantum circuits. Quantum Information and Cryptography: Comparison between classical and quantum information theory. Bell states. Quantum teleportation. Quantum Cryptography, no cloning theorem. Quantum Algorithms: Classical computation on quantum computers. Relationship between quantum and classical complexity classes. Deutsch's algorithm, Deutsch's-Jozsa algorithm, Shor factorization, Grover search. Noise and error correction: Graph states and codes, Quantum error correction, fault-tolerant computation.

Reading:

1. Nielsen M. A., *Quantum Computation and Quantum Information*, Cambridge University Press. 2002
2. Benenti G., Casati G. and Strini G., *Principles of Quantum Computation and Information*, *Vol. I: Basic Concepts, Vol II: Basic Tools and Special Topics*, World Scientific. 2004
3. Pittenger A. O., *An Introduction to Quantum Computing Algorithms*, 2000

CS5124	Cryptography and Network Security	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Understand the principles of design of cryptographic algorithms
CO2	Apply cryptographic algorithms to build security protocols
CO3	Identify the vulnerabilities of Internet protocols
CO4	Design firewalls and intrusion detection system

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3		2	3		
CO2	3		3	3		
CO3	3		3	3		
CO4	3		3	3		

Detailed syllabus:

History and overview of cryptography- One time pad and stream ciphers - Block ciphers block cipher abstractions: pseudo random permutations and pseudo random functions - Message integrity: definition and application - Collision resistant hashing - Arithmetic modulo primes - Cryptography using arithmetic modulo primes - Public key encryption - Arithmetic modulo composites - Authentication: authenticated encryption authenticated key exchange - Digital signatures : Definition and application more signature schemes - Identification protocols - Introduction to network security and associated techniques - Security issues in Internet protocols: TCP, DNS, and routing - Network defense tools: Firewalls, VPNs, Intrusion detection, and filters - Standards : Kerberos v4 Kerberos v5, PKI, IPsec AH ESP ,IPsec IKE, SSL/TLS, S/MIME andPGP

Reading:

1. J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, CRCPress,2008
2. A. Menezes, P. Van Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, CRC Press,2004
3. Charlie Kaufman, Radia Perlman, Mike Speciner, *Network Security: Private Communication in a Public World*, Prentice Hall,2002

CS5125	Applied Artificial Intelligence	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Solve searching problems using A*, Mini-Max algorithms.
CO2	Create logical agents to do inference using first order logic.
CO3	Use Bayesian learning for classification problems.
CO4	Understand different phases of Natural Language Processing.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	1	0	0	2	1	1
CO2	2	0	1	0	0	2
CO3	3	0	2	1	1	2
CO4	2	1	2	1	0	2

Detailed syllabus

INTRODUCTION – Agents and Objects, Evaluation of Agents, Agent Design Philosophies, Multi-agent System, Mobile Agents, Agent Communication,Knowledge query and Manipulation Language. What is AI? The Foundations of Artificial Intelligence;

INTELLIGENT AGENTS – Agents and Environments, Good Behavior: The Concept of Rationality, The Nature of Environments, The Structure of Agents.

SOLVING PROBLEMS BY SEARCH – Problem-Solving Agents, Formulating problems, Searching for Solutions, Uninformed Search Strategies, Breadth-first search, Depth-first search, Searching with Partial Information, Informed (Heuristic) Search Strategies, Greedy best-first search, A* Search: Minimizing the total estimated solution cost, Heuristic Functions, Local Search Algorithms and Optimization Problems, Online Search Agents and Unknown Environments.

INFERENCE IN FIRST-ORDER LOGIC – Syntax and Semantics of First-Order Logic, Using First-Order Logic, Knowledge Engineering in First-Order Logic; Propositional vs. First-Order Inference, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution.

SYMBOLIC REASONING UNDER UNCERTAINTY – Introduction to Nonmonotonic Reasoning, Logics for Nonmonotonic Reasoning, Implementation Issues,BAYESIAN LEARNING – Bayes theorem, Concept learning, Bayes Optimal Classifier, Naïve Bayes classifier, Bayesian belief networks, EM algorithm;

NATURAL LANGUAGE PROCESSING – Phrase Structure Grammars, Syntactic Analysis (Parsing), Augmented Grammars and Semantic Interpretation, Machine Translation, Speech Recognition.

Reading:

1. Tom M.Mitchell, Machine Learning, McGraw Hill, 1997.
2. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
3. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, 3rd Edition, Pearson, 2010.
4. Kevin Knight, Elaine Rich and B. Nair, Artificial Intelligence, Third Edition, McGraw Hill, 2017.

CS5126	Bio-Informatics	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Classify models used in bioinformatics.
CO2	Compute homologues, analyze sequences, construct and interpret evolutionary trees.
CO3	Analyze protein sequences to retrieve protein structures from databases.
CO4	Design of biological data model
CO5	Apply homology modeling and computational drug design.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1			2			2
CO2	2		2			2
CO3	2		2			2
CO4	2		2			1
CO5	2		2			

Detailed syllabus

Introduction and Biological databases - Introduction, Sequence Alignment - Pair wise sequence alignment, Database similarity searching, Multiple sequence alignment, Profiles and hidden markov models, Molecular Phylogenetics - Phylogenetics basics, Phylogenetic Tree Construction Methods and Programs, Genomics and Proteomics - Genome mapping, assembly and comparison, Functional genomics, Proteomics, Structural Bioinformatics - Basics of protein structure, Protein structure prediction.

Reading:

1. Jin Xiong, *Essential Bioinformatics*, 1th Edition, Cambridge University Press,2011.
2. Arthur M Lesk, *Introduction to Bioinformatics*, 2nd Edition, Oxford University Press,2007.

CS5161	Service Oriented Architecture and Microservices	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Understand software-oriented architectures
CO2	Design medium scale software project development using SOA principles
CO3	Develop SOA messages from business use cases
CO4	Design and implement modern SOA and SOA-specific methodologies, technologies and standards
CO5	Create composite services by applying composition style
CO6	Design Applications Using Microservices

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	1		1	1		
CO2	2		2	2		
CO3	2		2	2		1
CO4	2		2	2		1
CO5	2		2	2		1
CO6	2		2	2		1

Detailed syllabus

Introduction To SOA, Evolution Of SOA: Fundamental SOA; Common Characteristics of contemporary SOA; Common tangible benefits of SOA; An SOA timeline (from XML to Web services to SOA); The continuing evolution of SOA (Standards organizations and Contributing vendors); The roots of SOA (comparing SOA to Past architectures). Web Services and Primitive SOA: The Web services framework· Services (as Webservices); Service descriptions (with WSDL); Messaging (with SOAP). Web Services And Contemporary SOA – I Message exchange patterns; Service activity; Coordination; Atomic Transactions; Business activities; Orchestration; Choreography. Web Services And Contemporary SOA-2: Addressing; Reliable messaging; Correlation; Polices; Metadata exchange; Security; Notification and eventing. Principles Of Service - Orientation: Services orientation and the enterprise; Anatomy of a service oriented architecture; Common Principles of Service orientation; How service orientation principles interrelate; Service orientation and object orientation; Native Web service support for service orientation principles. Service Layers: Service orientation and contemporary SOA; Service layer abstraction; Application service layer, Business service layer, Orchestration service layer; Agnostic services; Service layer configuration scenarios. Business Process Design: WS-BPEL language basics; WS Coordination overview; Service oriented business process design; WS addressing language basics; WS Reliable Messaging language basics. SOA Platforms: SOA platform basics; SOA support in J2EE; SOA support in. ET; Integration considerations. Microservices: Introduction to Microservices, Challenges, SOA vs Microservices, Design and Implementation of Microservices

Reading:

1. Thomas Erl, *Service-Oriented Architecture: Concepts, Technology and Design*, Prentice Hall Publication, 2005.
2. Michael Rosen, Boris Lublinsky, *Applied SOA Service Oriented Architecture and Design Strategies*, Wiely India Edition, 2008.
3. Wolff, Eberhard. *Microservices: flexible software architecture*. Addison-Wesley Professional, 2016.

CS5162	Privacy Preserving Data Publishing	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Apply anonymization methods for sensitive data protection
CO2	Apply state-of-art techniques for data privacy protection
CO3	Design privacy preserving algorithms for real-world applications .
CO4	Identify security and privacy issues in OLAP systems
CO5	Apply information metrics for Maximizing the preservation of information in the anonymization process

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2			3		2
CO2	2			3		2
CO3	2			3		2
CO4	2			3		2
CO5	2			3		2

Detailed syllabus

Fundamentals of defining privacy and developing efficient algorithms for enforcing privacy, challenges in developing privacy preserving algorithms in real-world applications, privacy issues, privacy models, anonymization operations, information metrics, Anonymization methods for the transaction data, trajectory data, social networks data, and textual data, Collaborative Anonymization, Access control of outsourced data, Use of Fragmentation and Encryption to Protect Data Privacy, Security and Privacy in OLAP systems, Extended Data publishing Scenarios , Anonymization for Data Mining, publishing social science data, continuous user activity monitoring (like in search logs, location traces, energy monitoring), social networks, recommendation engines and targeted advertising.

Reading:

1. Benjamin C.M. Fung, Ke Wang, Ada Wai-Chee Fu and Philip S. Yu, *Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques*, 1st Edition, Chapman & Hall/CRC,2010.
2. Charu C. Aggarwal, *Privacy-Preserving Data Mining: Models and Algorithms*, 1st Edition, Springer,2008.

CS5163	Software Reliability and Quality Management	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Understand Software Reliability during different phases of Software Development Life Cycle
CO2	Analyze Software Reliability parameters using Markovian Modeling
CO3	Estimate Software Reliability parameters using Maximum Likelihood and Least Square Method
CO4	Evaluate performance of Binomial-Type, Poison-Type and Markovian Models
CO5	Predict Software Reliability using Intelligent Techniques
CO6	Design Quality Attributes for Software Quality Assurance (SQA)

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	1		1	1		2
CO2	2			2		2
CO3	2			2		2
CO4	2			2		2
CO5	2			2		2
CO6	2			2		2

Detailed syllabus

Introduction to Software Reliability: The need for Software Reliability, Some Basic Concepts, Software Reliability and Hardware Reliability, Availability, Modelling and General Model Characteristics. Software Reliability Modeling: Halstead's Software Metric, McCabe's Cyclomatic Complexity Metric, Error Seeding Models, Failure Rate Models, Curve Fitting Models, Reliability Growth Models, Markov Structure Models, Time Series Models, Non-homogeneous Poison Process Models. Markovian Models: General Concepts, General Poison-Type Models, Binomial -Type Models, Poison-Type Models, Comparison of Binomial-Type and Poison-Type Models, Fault Reduction Factor for Poison-Type Models. Descriptions of Specific Models: Finite Failure Category Models, Infinite Failure Category Models. Parameter Estimation: Maximum Likelihood Estimation, Least Squares Estimation, Bayesian Inference. Comparison of Software Reliability Models: Comparison Criteria, Comparison of Predictive Validity of Model Groups, Evaluation of other Criteria. Software Reliability Prediction: Problems associated with different Software Reliability Models, Software Reliability prediction parameters, Intelligent Techniques for Software Reliability Prediction. Software Quality Management: Software Quality Attributes, Quality Measurement & Metrics, Verification & Validation Techniques, Verification & Validation in the Life Cycle, Software Quality Assurance functions, Tool support for SQA.

Reading:

1. M. Xie, *Software Reliability Modelling*, World Scientific; 1991.
2. John D. Musa, Anthony Iannino, Kazuhira Okumoto, *Software Reliability Measurement, Prediction, Application*. McGraw-Hill Book Company; 1987.
3. Hoang Pham, *System Software Reliability*, Springer; 2005
4. David C. Kung, *Object-Oriented Software Engineering: An Agile Unified Methodology,* McGraw-Hill Education (India) Edition 2015.

CS5164	Research Study	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Comprehend popular techniques in the chosen area of research.
CO2	Relate some technological problems to the research areas.
CO3	Justify the approaches to the problems.
CO4	Write survey paper.
CO5	Revise some method in the concerned domain for better solution.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	3	2			
CO2	2	3	2			
CO3		3	2			
CO4		3	2			
CO5		3	2			

Detailed syllabus

Research Monographs, Articles, Papers as prescribed by the faculty.

CS5165	Formal Methods in program Design	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Understand basic concepts used in program design such as determinism / non-determinism, synchrony/asynchrony, separation of concerns like correctness and complexity, programs and implementation
CO2	Demonstrate the concept of progress and proofs thereof
CO3	Analyze architecture and mappings by taking different case studies
CO4	Understand program structuring and program composition

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		2	2		
CO2	2		1	2		
CO3	1		2	2		
CO4	1		2	2		

Detailed syllabus

Foundations of parallel programming, Nondeterminism, Absence of control flow, Synchrony and Asynchrony, States and assignments, extricating proofs from program text, separation of concerns: correctness and complexity, programs and implementation. UNITY program structure, Assignment statement, Assign section, Initially-section, Always-section, Proving assertions about Assignment statement, Quantified assertions, conventions about priorities of logical relations. Fundamental concepts, proofs and theorems about: Unless / Ensures / Leads-to / Fixed-point. Proving bounds on progress. Introduction about Architecture and Mappings. All pairs shortest path problem: solution strategy, formal description, proof of correctness, creating the program. Implementation on sequential architectures, parallel synchronous architectures, asynchronous shared-memory architecture, and distributed architecture. Complexities on each of the architectures. Formal description and programs for saddle-point-of-a-matrix, reachability in directed graphs, prime number generation, comparing two ascending sequences, computing the maximum of a set of numbers, Boolean matrix multiplication. Program structuring, program composition by Union, Union theorem, composing specifications, substitution axiom, hierarchical program structures, superposition and superposition theorem, design specifications. Introduction to communicating processes.

Reading:

1. *K. Mani Chandy, Jayadev Misra, Parallel Program Design, A foundation, Addison-Wesley Publishing, 1988.*
2. *David Gries, The Science of Programming, Springer, 1981.*
3. *Jean Gaullier, Logic for Computer Science: Foundations of Automatic Theorem Proving, 2nd Edition, Harper & Row, Computer Science Technology Series, 2015*

CS5166	Security and Privacy	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Evaluate the risks and vulnerabilities in protocols/Standards.
CO2	Apply Number Theory and Algebra required for designing cryptographic algorithms.
CO3	Design symmetric key and asymmetric key encryption techniques.
CO4	Design authentication, message integrity and authenticated encryption protocols.
CO5	Design and analyze security of systems including distributed storage and Electronic voting.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		3	3		
CO2	2		3	3		
CO3	2		3	3		
CO4	2		3	3		
CO5	2		3	3		

Detailed syllabus

Introduction to Security – risks, threats and vulnerabilities, Cryptography, Stream Ciphers – One-time Pad (OTP), Perfect secrecy, Pseudo-random generators (PRG), Attacks on stream ciphers and OTP, Real world stream ciphers, Semantic security, Case Study- RC4, Salsa 20, CSS in DVD encryption, A5 in GSM, Block ciphers- DES, attacks, AES, Block ciphers from PRG, Modes of operation – one-time key and many-time keys, CBC, CTR modes, Message Integrity – MAC, MAC based on PRF, NMAC, PMAC, Collision resistance – Birthday attack, Merkle-Damgard construction, HMC, Case study:SHA-256, Authenticated encryption, Key exchange algorithms, Public key cryptosystems – RSA, ElGamal, Elliptic curve cryptosystems – PKC, key exchange, IBE, Case studies – HTTPS – SSL/TLS, SSH, IPSec, 802.11i WPA, System design and analysis – Survivable distributed storage system, Electronic voting system.

Reading:

1. J. Thomas Shaw, *Information Security Privacy*, ABA,2012.
2. J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, CRC press,2008.
3. Menezes, et. al, *Handbook of Applied Cryptography*, CRC Press,2004.
4. A. Abraham, *Computational Social Networks: Security and Privacy*, Springer,2012.

CS5167	Cognitive Radio Networks	Elective	3 – 0 – 0	3 Credits

Prerequisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Understand Software Defined Radio (SDR) architecture, Relationship between SDR and Cognitive Radio (CR)
CO2	Demonstrate CR capabilities, architecture and dynamic spectrum access.
CO3	Analyze different spectrum sensing mechanisms and spectrum management functions in CR.
CO4	Demonstrate upper layer issues and cooperative communications in CR networks (CRN).
CO5	Analyze different security attacks and countermeasures in CRN.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	1		1	1	1	1
CO2	2		1	1	1	1
CO3	2		2	1	1	2
CO4	2		2	1	2	2
CO5	3		3	2	1	2

Detailed Syllabus

SOFTWARE DEFINED RADIO (SDR): Introduction to SDR, Definitions and potential benefits, Evolution of SDR, Essential functions, SDR architecture, Design principles of SDR, Reconfigurable wireless communication systems, SDR and Cognitive Radio Relationship. COGNITIVE RADIO (CR) Introduction to CR technology, Features and capabilities, CR functions, CR architecture, CR Components, Cognitive cycle, CR and Dynamic spectrum access, Interference temperature, CR standardization.SPECTRUM SENSING AND SPECTRUM MANAGEMENT:Spectrum sensing to detect Primary System, Primary signal detection- Techniques, Cooperative sensing, Spectrum decision, Spectrum sharing and Spectrum Pricing, Spectrum mobility, Mobility management of heterogeneous wireless networks, Regulatory Issues and International Standards.UPPER LAYER ISSUES IN CR NETWORKS (CRN): Routing in CRN, Control of CRN: Flow Control, End-to-End Error Control, Congestion Control in Transport Layer, Congestion Control in Internet, Self-Organized Networks, Cooperative communications, Cooperative wireless networks, Cross-layer design, Next generation (xG) wireless networks and Architecture.SECURITY IN CRN: Security requirements of CRN, Selfish and Malicious attacks, Intentional Jamming Attack -Primary Receiver Jamming Attack, Sensitivity Amplifying Attack, Overlapping Secondary User Attack, Biased Utility Attack, Asynchronous Sensing Attack, False Feedback Attack, Network Endo-Parasite Attack (NEPA), Channel Ecto-Parasite Attack (CEPA), Low cost Ripple effect Attack (LORA), Key Depletion Attack, Licensed User Emulation Attack, Common Control Channel Jamming, Objective Function Attacks, Spectrum Sensing Data Falsification Attack, Fabrication Attack, On–off Attack, Denial of Service Attack, Resource Hungry Attack, Lion Attack, Jellyfish Attack,Challenges and threats in cognitive radio networks, Countermeasures, Challenges and open problems in cognitive radio networks.

Reading:

1. Kwang-Cheng Chen and Ramjee Prasad, “Cognitive Radio Networks”, John Wiley & Sons, Ltd, 2009.
2. Alexander M. Wyglinski, Maziar Nekovee, and Y. Thomas Hou, “Cognitive Radio Communications and Networks - Principles and Practice”, Elsevier Inc., 2010.
3. Ahmed Khattab, Dmitri Perkins, MagdyBayoumi, “Cognitive Radio Networks : From Theory to Practice”, Springer, 2013.
4. E. Biglieri, A.J. Goldsmith., L.J. Greenstein, N.B. Mandayam, H.V. Poor, “Principles of Cognitive Radio”, Cambridge University Press, 2013.

CS5168	Model Driven Frameworks	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Apply software development techniques with reference to model driven software development
CO2	Design and implement the practical application of domain-specific modeling language.
CO3	Identify verification and translation of specifications.
CO4	Analyze emerging model-driven development techniques.
CO5	Integrate a set of models to perform effective software specifications.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		2	2		
CO2	2		2	1		
CO3	2	2	2	1		
CO4	2		1	1		
CO5	2	2	2	2		

Detailed syllabus

MDSD basic ideas and terminology: The challenges, The goals of MDSD, MDSD approach, architecture. Case study: a typical web application. Concept formation: Common MDSD concepts and terminology, model driven architecture, architecture centric MDSD, Generative Programming. Classification: MDSD vs. CASE, 4GL, wizard, roundtrip engineering, MDSD and Patterns, MDSD and domain driven design. MDSD capable target architecture: Software Architecture in the context of MDSD. Building blocks of software architecture. Architecture reference model, balancing the MDSD platform, MDSD and CBD, SOA, BMP.

Building domain architecture: DSL construction, General transformation architecture, technical aspects of building transformations, the use of interpreters. Code generation techniques: categorization, generation techniques Model transformations with QVT, M2M language requirements. MDSD tools: roles, architecture, selection criterion and pointers.

Software processes - modular-based software design - Model-driven Architecture (MDA): What is meta-modeling, Meta-levels vs. Levels of abstraction, MDA Frameworks: Platform Independent Model PIM and Platform Specific Model. System modeling- MOF's meta-modeling.

Reading:

1. Thomas Stahl, Markus Voelter, *Model-Driven Software Development: Technology, Engineering, Management*, Wiley,2006.
2. Anne Kleppe, Jos Warmer and Wim Bast, *MDA Explained. The Model Driven Architecture, Practice and Promise*, Pearson Education, Boston, USA,2003.

CS5169	Exploratory and Interactive Data Analysis	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Manage, Explore, Analyse and synthesize the results of specific data processing methods
CO2	Apply the knowledge of data analysis in the fieldssuch as diagnosis; forecasting; planning; decisionmaking.
CO3	Highlight the statistical features of observed datasets
CO4	Apply Data Analysis to various sizes and complexity of the data sets.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		2	2		3
CO2	3	3	3	2		3
CO3	2	3				3
CO4	2	3	2	2		3

Detailed syllabus

Data conception, Statistical Data Elaboration, 1-D statistical analysis. 2-D statistical analysis, *N-D Statistical* analysis, Factor analysis, Principal Component Analysis, 2-D Correspondence Analysis, N-D Correspondence Analysis, Classification of Individuals-Variables and Data Sets, Classification and Analysis of Proximities Data Sets, Singular ValueDecomposition,Advanced exploratory data analysis, Data classification or clustering, Data input.

Reading:

1. Michel Jambu, *Exploratory and Multivariate Data Analysis*, Academic Press,1991.
2. Francois Husson, Sebastien Le, Jérôme Pagès, *Exploratory Multivariate Analysis by Example Using R*, CRC Press,2010.

CS5170	Internet of Things	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Understand current and future directions of Internet of Things
CO2	Design and develop communication protocols in Internet of Things
CO3	Develop smart environment and applications which advance the Internet of Things
CO4	Analyze the societal impact of Internet of Things
CO5	Analyze vulnerabilities, including recent attacks, involving the Internet of Things

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1			2	2	1	
CO2	3		3	2	1	
CO3	2		2	3	1	3
CO4	2	3	3	3	1	
CO5	2		3	3	1	

Detailed Syllabus:

Internet of Things (IoT) frameworks and applications, IoT Standards, Smart Environments, Communication capabilities and Device Intelligence, Sensor and RFID Technology, Wireless Technologies for IoT, Zigbee/IEEE 802.15.4, IEEE 802.15.6 WBANs, Comparison of WPAN technologies, Mobile IPv6 for IoT, Machine-to-Machine communication models, Service Discovery in IoT, Service oriented Middleware, Resource management in IoT, Web of Things, Sensor Web, Crowd sourcing, Securing Internet of Things: vulnerabilities and attacks.

Reading:

1. O Hersent, D Boswarthick and O Elloumi, The Internet of Things: Key applications and protocols, Wiley, 2012.
2. Daniel Minoli, *Building The Internet Of Things with IPv6 and MIPv6: The Evolving World of M2M Communications*, John Wiley & Sons,2013.
3. Dieter Uckelmann, Mark Harrison and Florian Michahelles, *Architecting the Internet of Things*, Springer,2011.
4. Nik Bessis and Ciprian Dobre, *Big Data and Internet of Things: A Roadmap for Smart Environments*, Springer,2014.
5. Giancarlo Fortino and Paolo Trunfio, *Internet of Things Based on Smart Objects- Technology, Middleware and Applications*, Springer,2014.

CS5171	Cloud Computing	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Identify cloud services for application
CO2	Analyze Cloud infrastructure including Google Cloud and Amazon Cloud.
CO3	Analyze authentication, confidentiality and privacy issues in Cloud computing environment.
CO4	Analyze the financial and technological implications for selecting cloud computing platforms.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		2	3	1	
CO2	2		2	3	1	2
CO3	2		2	3	1	2
CO4	2		2	3	1	2

Detailed syllabus

Introduction - Cloud Computing Architecture, Cloud Delivery Models, The SPI Framework, SPI Evolution, The SPI Framework vs. the Traditional IT Model, Cloud Software as a Service (SaaS), Cloud Platform as a Service (PaaS), Cloud Infrastructure as a Service (IaaS) Google Cloud Infrastructure - Google File System – Search engine – MapReduce - Amazon Web Services - REST APIs - SOAP API - Defining Service Oriented Architecture, Combining thecloudandSOA,CharacterizingSOA,LooseningUponCoupling,MakingSOAHappen,
Catching the Enterprise Service Bus, Telling your registry from your repository, Cataloging services, Understanding Services in the Cloud.
Serving the Business with SOA and Cloud Computing, Query API - User Authentication-Connecting to the Cloud - OpenSSH Keys - Tunneling / Port Forwarding - Simple Storage Service - S3, EC2 - EC2 Compute Units, Platforms and storage, EC2 pricing, EC2 customers Amazon Elastic Block Storage - EBS - Ubuntu in the Cloud - Apache Instances in EC2 – Amazon Cloud Services- Amazon Elastic Compute Cloud (Amazon EC2), Amazon SimpleDB, Amazon Simple Storage Service (Amazon S3), Amazon CloudFront, Amazon Simple Queue Service (Amazon SQS), Amazon Elastic MapReduce, Amazon Relational Database Service (Amazon RDS) , EC2 Applications - Web application design - AWS EC2 Capacity Planning – Apache Servers - Mysql Servers - Amazon Cloud Watch - Monitoring Tools.

Reading:

1. Judith Hurwitz, R Bloor, M Kanfman, F Halper, *Cloud Computing for Dummies*, 1st Edition, Wiley Publishers,2009.
2. Gautam Shroff, *Enterprise Cloud Computing*, Cambridge,2010.
3. Ronald Krutz and Russell Dean Vines, *Cloud Security*, 1st Edition, Wiley,2010.

CS5172	Optimization in Computer Science	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Understand various optimization models and methodologies.
CO2	Prepare and solve linear programming model.
CO3	Identify the models applicable to various applied problems in Computer Science.
CO4	Analyze methods to solve problems related to Computer Science

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		3	2	2	2
CO2	2		3	2	2	2
CO3	2		3	2	2	2
CO4	2		3	2	2	2

Detailed Syllabus:
Introduction to Problem Solving with mathematical models, linear programs, nonlinear programs, discrete programs and multiobjective optimization models. Improving search, Local and Global Optima.
Mathematical Preliminaries like Vectors and Matrices, approximation using the Taylor series, solution of nonlinear equations, quadratic forms and convex functions.
Linear Programming models. LP optimal solutions and standard form. Simplex algorithm and its representations. Interior point methods for LP. Duality and Sensitivity in LP.
Multiobjective and Goal Programming. Shortest Path and Discrete Dynamic Programming. Network Flows, single commodity and multicommodity flows.
Discrete Optimization models like Knapsack, set packing, travelling salesman and network design models. Solving by total enumeration, branch and bound and improved search heuristics like simulated annealing and genetic algorithm.
Unconstrained Nonlinear Optimization with gradient search, Newton's method.
Constrained Nonlinear Programming with Lagrange multiplier methods, KKT conditions, quadratic programming and separable programming methods.

Reading:

1. R. L. Rardin ,*Optimization in Operations Research*, Pearson Education, 2017.
2. M. Asghar Bhatti, *Practical Optimization Methods with Mathematica Applications*, Springer, 2000
3. Hamdy A. Taha, *Operations Research: An Introduction*, Pearson Education, 2010.

CS5173	High Performance Computing	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Design and analyze parallel algorithms for real world problems and implement them on available parallel computer systems.
CO2	Optimize the performance of a parallel program to suit a particular hardware and software environment.
CO3	Write Programs using accelerator technologies of GPGPUs with CUDA, OpenCL.
CO4	Design algorithms suited for Multicore processor systems using OpenCL, OpenMP, and Threading techniques.
CO5	Analyze the communication overhead of interconnection networks and modify the algorithms to meet the requirements.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		2	2	2	
CO2	2		2	2	2	
CO3	2		2	2		
CO4	2		2	2	1	
CO5	2		2	2		

Detailed syllabus

Introduction: Implicit parallelism, Limitations of memory system performance, control structure, communication model, physical organization, and communication costs of parallel platforms, Routing mechanisms for interconnection networks, Mapping techniques

Parallel algorithm design: Preliminaries, decomposition techniques, tasks and interactions, mapping techniques for load balancing, methods for reducing interaction overheads, parallel algorithm models, Basic communication operations: Meaning of all-to-all, all-reduce, scatter, gather, circular shift and splitting routing messages in parts.

Analytical modeling of parallel programs: sources of overhead, performance metrics, the effect of granularity on performance, scalability of parallel systems, minimum execution time, minimum cost-optimal execution time, asymptotic analysis of parallel programs

Programming using message passing paradigm: Principles, building blocks, MPI, Topologies and embedding, Overlapping communication and computation, collective communication operations, Groups and communicators

Programming shared address space platforms: Threads, POSIX threads, Synchronization primitives, attributes of threads, mutex and condition variables, Composite synchronization constructs, OpenMP.

Multi-core Programming: Multi-core processor, CPU Cache, Cache coherence protocols, Memory Consistency Models, An Overview of Memory Allocators, Programming Libraries-PThreads, TBB, OpenMP, Dense Matrix Algorithms: matrix vector multiplication, matrix-matrix multiplication, solving system of linear equations,

Sorting: Sorting networks, Bubble sort, Quick sort, Bucket sort and other sorting algorithms

Graph algorithms: Minimum spanning tree, single source shortest paths, all-pairs shortest paths, Transitive closure, connected components, algorithms for sparse graphs

Reading:

1. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, *Introduction to Parallel Computing*, Second Edition, Pearson Education,2007
2. Benedict R Gaster, Lee Howes, David R Kaeli Perhaad Mistry DanaSchaa, *Heterogeneous Computing with OpenCL*, McGraw-Hill, Inc. Newyork , 2011
3. Michael J. Quinn*, Parallel Programming in C with MPI and OpenMP*, McGraw-Hill International Editions, Computer Science Series,2004
4. Jason Sanders, Edward Kandrot, *CUDA By Example – An Introduction to General- Purpose GPU Programming*, Addison Wesley,2011.

CS5174	Randomized and Approximation Algorithms	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Design and analyze efficient randomized algorithms
CO2	Apply tail inequalities to bound error-probability
CO3	Analyze randomized algorithms with respect to probability of error and expected running time.
CO4	Analyze approximation algorithms and determine approximation factor.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3		2	2		
CO2	3		2	2		
CO3	3		2	2		
CO4	3		2	2		

Detailed syllabus

Introduction, Las Vegas and Monte Carlo Algorithms, Computational Model and Complexity Classes, Game Tree Evaluation, The Markov and Chebyshev Inequalities, The Stable Marriage Problem, The Coupon Collectors Problem, The Chernoff Bound, Routing in a Parallel Computer, The Probabilistic Method: Overview, probabilistic analysis, use of indicator random variables, Randomly permuting arrays, Birthday paradox, analysis using indicator random variables, Balls and bins, Streaks, Online hiring problem, Maximum Satisfiability, Expanding Graphs, The Lovasz Local Lemma, Markov Chains, Random Walks on Graphs, Graph Connectivity, Expanders and Rapidly Mixing Random Walks, Pattern Matching, Random Trees, Skip Lists, Hash Tables, Linear Programming, The Min-Cut Problem, Minimum Spanning Trees, The DNF Counting Problem, Maximal Independent Sets, Perfect Matching, The Online approximations paging Problem, Adversary Models and Paging against an Oblivious Adversary, Vertex cover problem, traveling salesman problem with triangle inequality, general traveling salesman problem, set-covering problem, a greedy approximation algorithm, analysis Randomization and linear programming, randomized approximation, subset-sum problem, Absolute approximations, Planar Graph Coloring, MaximumProgramsStoredProblem,NP-hardAbsoluteApproximationsε-approximations, Polynomial time approximations schemes, Scheduling Independent Tasks , 0/1 Knapsack, Fully Polynomial time approximations scheme , Rounding , Interval Partitioning , Separation, probabilistically goodalgorithms.

Reading:

1. Rajeev Motwani and Prabhakar Raghavan, *Randomized Algorithms*, Cambridge University Press,1995.
2. J. Hromkovic, *Design and Analysis of Randomized Algorithms*, Springer,2005.

CS5175	Human Computer Interaction	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Design and Develop processes and life cycle of Human Computer Interaction
CO2	Analyze product usability evaluations and testing methods.
CO3	Apply the interface design standards/guidelines for cross cultural and disabled users.
CO4	Categorize, Design and Develop Human Computer Interaction in proper architecturalstructures.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	2	2	2		2
CO2	2	2	2	2		
CO3	2	2	2	2		2
CO4	2	1	2	2		

Detailed syllabus

HCI foundations- Input–output channels, Human memory, Thinking: reasoning and problem solving, Emotion, Individual differences, Psychology and the design of interactive systems, Text entry devices, Positioning, pointing and drawing, Display devices, Devices for virtual reality and 3D interaction, Physical controls, sensors and special devices, Paper: printing andscanning.

Designing- Programming Interactive systems- Models of interaction, Frameworks and HCI, Ergonomics, Interaction styles, Elements of the WIMP interface, The context of the interaction, Experience, engagement and fun, Paradigms for interaction,

Cantered design and testing- Interaction design basics-The process of design, User focus, Scenarios, Navigation design, Screen design and layout, Iteration and prototyping, Design for non-Mouse interfaces, HCI in the software process, Iterative design and prototyping, Design rules, Principles to support usability, Standards and Guidelines, Golden rules and heuristics, HCIpatterns.

Implementation support - Elements of windowing systems, Programming the application, Using toolkits. User interface management systems, Evaluation techniques, Evaluation through expert analysis, Evaluation through user participation, Universal design, User support, Models and Theories - Cognitive models, Goal and task hierarchies, Linguistic models, The challenge of display-based systems, Physical and device models, Cognitive architectures Collaboration and communication - Face-to-face communication, Conversation, Text-based communication, Group working, Dialog design notations, Diagrammatic notations, Textual dialog notations, Dialog semantics, Dialog analysis and design, Human factors and security - Groupware, Meeting and decision support systems, Shared applications and artifacts, Frameworks for groupware Implementing synchronous groupware, Mixed, Augmented and Virtual Reality.

Reading:

1. A Dix, Janet Finlay, G D Abowd, R Beale, *Human-Computer Interaction*, 3rd Edition, PearsonPublishers,2008.
2. Shneiderman, Plaisant, Cohen and Jacobs, *Designing the User Interface: Strategies for Effective Human Computer Interaction*, 5th Edition, Pearson Publishers,2010.

CS5176	Social Media Analytics	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Classify social networks
CO2	Analyze social media and networking data
CO3	Apply Social networks Visualization tools
CO4	Analyze the social data using graph theoretic computing approach
CO5	Identify application driven virtual communities from social networks
CO6	Apply sentiment mining

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		1	2		3
CO2	3		2	2		3
CO3	3		2	2		3
CO4	2		2	2		3
CO5	1		2	2		3
CO6	2		2	2		3

Detailed syllabus

Introduction to social network analysis, Vertex or node, edge, neighbors, degree, shortest path, cycle, tree, complete graph, bipartite graphs, directed graphs, weighted graphs, adjacency matrix, connected components, Games on networks, game theory strategies, dominant strategies, dominated strategies, pure strategies and mixed strategies, Nash equilibrium, multiple equilibria-coordination games, multiple equilibria-the Hawk-Dove game, mixed strategies, Modeling network traffic using game theory. Technological networks (internet, telephone network, power grids, transportation networks), social networks (facebook, movie collaboration, paper collaboration), information networks (web), biological networks (neural networks, ecological networks), Random models of networks, Erdos-Renyi model of random graph, models of the small world, decentralized search in small-world , random graphs with general degree distributions, models of network formation, Spread of influence through a network, influence maximization in networks, spread of disease on networks, Information networks, structure of the web, link analysis and web search, page rank, spectral analysis of page rank and hubs and authorities, random walks, auctions and matching markets, sponsored search markets.

Reading:

1. David Easley and Jon Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World,* Cambridge University Press,2010.
2. Mark Newman, *Networks: An Introduction,* Oxford University Press,2010.
3. Hansen, Derek, Ben Sheiderman, Marc Smith, *Analyzing Social Media Networks with NodeXL: Insights from a Connected World*, Morgan Kaufmann,2011.
4. Avinash Kaushik, *Web Analytics 2.0: The Art of Online Accountability*, Sybex,2009.

CS5177	Models for Social Networks	DEC	3 – 0 – 0	3 Credits

Pre-requisites: Algorithms; Probability and Statistics

Course Outcomes: At the end of the course the student will be able to:

CO1	Develop random graph models for real-world networks
CO2	Understand the spread of information, disease, influence, etc., on networks
CO3	Design and develop models and algorithms for web search and sponsored search
CO4	Apply game-theoretic approaches to interaction on networks.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		2	2		2
CO2	2		2	2		2
CO3	2		2	2		2
CO4	2		2	3	2	2

Detailed Syllabus

Introduction to Networks: Empirical Study of Networks: Technological networks, social networks, networks of information, biological networks; Fundamentals of network theory: Mathematics for networks, Measures and metrics, the large-scale structure of networks; Overview of available network data: Newman's Graph data sets, SNAP Graph library; Social network analysis software : Programming in Python and the NetworkX library, a distribution of Python for scientific computing andvisualization.

Random Models of Networks: Random graphs: Basic properties of random graphs, degree distribution, clustering coefficient, giant component, small components; The Erdos - Renyi model of random graph; Inadequacy of the Erdos - Renyi model: A simple alternate random graph model; The Kleinberg result; Random graphs with general degree distributions: The configuration model, Generating functions for degree distributions, Generating functions for the small components, Power-law degree distributions, Directed random graphs, , Models of network formation : The ``preferential attachment'' model of Barabasi and Albert, Vertex copying models, Stochastic Kronecker Graphs.

The Spread of "influence" through a Network: Stochastic Kronecker Graphs, The Christakis-Fowler work on the spread of obesity, happiness, etc. via social networks, Modeling information cascades, Viral Marketing.

Spread of Disease on Networks: Random mixing models: SI, SIS, SIR, SIRS, basic differential equations, Basic Reproductive Number and analysis of branching processes, Analysis of SIR on the Configuration model, Synchronization in disease incidence, explanation via models, and observational studies from Syphilis, Example of studies with some specific diseases.

Information Networks : The structure of the web, Link analysis and web search, Page rank, Spectral Analysis of Page rank and hubs and authorities, random walks, Auctions and matching markets, Sponsored search markets.

Games on Networks: Basics of Game Theory: strategies, dominant strategies, dominated strategies, pure strategies and mixed strategies, Nash equilibrium, Modeling network traffic as a game; Braess Paradox, Modeling Voluntary Vaccination as a game, Example of game-theoretic analysis applied to flu vaccine behavior.

Reading

1. David Easley and Jon Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*, Cambridge University Press,2010.
2. Mark Newman**,** *Networks: An Introduction*, Oxford University Press,2010.

CS5178	Advanced Compiler Design	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Understand code generation methods
CO2	Apply scalar variable optimizations and procedural optimizations on intermediate code.
CO3	Apply machine level optimizations on the low level intermediate code.
CO4	Perform loop restructuring transformations

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	1		1	1		
CO2	2		2	2	2	
CO3	2		2	2	2	
CO4	2		2	2	2	
CO5						

Detailed syllabus

Review of compiler fundamentals – lexical analysis, parsing, semantic analysis and intermediate code generation, error recovery, run time storage management, code generation. Code optimization – Peephole optimization, control flow analysis, data flow analysis, dependence analysis, redundancy elimination, loop optimization, procedural and inter procedural optimization, instruction scheduling. Compiling for High performance architectures, Compiling for scalar pipeline, compiling for vector pipeline, super scaler and VLIW processors, compiling for multiple issue processors, compiling for memory hierarchy. Parallelization and Vectorization, Dependence and dependence testing. Loop Normalization, Induction variable Exposure, Enhancing Fine Grained Parallelism, Loop Interchange, Scalar Expansion, Scalar and Array Renaming, Node splitting, Index-set splitting, Loop skewing

Reading:

1. Randy Allen, Kennedy, *Optimizing Compilers for Modern Architectures: A dependence-based approach*, Morgan Kaufmann Publishers, 2001
2. Steven S. Muchnick, *Advanced Compiler Design and implementation*, Morgan Kaufmann Publishers, 1997
3. Keith D. Cooper & Linda Torczon, *Engineering a Compiler*, Morgan Kaufmann, 2004.

CS5179	Deep Learning	PCC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Identify Convolutional Neural Networks models to solve Supervised Learning Problems
CO2	Design Autoencoders to solve Unsupervised Learning problems
CO3	Apply Long Shot Term Memory (LSTM)Networks for time series analysis classification problems.
CO4	Apply Classical Supervised Tasks for Image Denoising, Segmentation and Object detection problems.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3		3	2	1	2
CO2	2		3	2	1	3
CO3	2		3	2	1	3
CO4	2		3	2	1	3

Detailed syllabus

INTRODUCTION – History of Deep Learning, Introduction to Neural Network, Perceptrons, Perceptron Learning Algorithm. Multilayer Perceptrons (MLPs), Representation of MLPs, Sigmoid Neurons, Gradient Descent. FeedForward Neural Networks, Backpropagation. Gradient Descent (GD), Momentum Based GD, Stochastic GD;REGULARIZATION- Bias Variance Tradeoff, L2 regularization, Early stopping, Dataset augmentation, Parameter sharing and tying; SUPERVISED DEEP LEARNING - Convolutional Neural Networks,Building blocks of CNN, Transfer Learning,LeNet, AlexNet, ZF-Net, VGGNet, GoogLeNet, ResNet Models, Visualizing ConvolutionalNeural Networks; Unsupervised Learning with Deep Network, Autoencoders, Variational Autoencoder, Regularization in autoencoders, Denoising autoencoders, Sparse autoencoders; Recent Trends in Deep Learning Architectures, Residual Network, Skip Connection Network, Fully Connected CNN, Recurrent Neural Networks (RNN), Long Shot Term Memory (LSTM)Networks, Generative Adversarial Networks (GAN); Classical Supervised Tasks with Deep Learning, Image Denoising, Semantic Segmentation, Object Detection.

Reading:

4. Deep Learning- Ian Goodfelllow, YoshuaBenjio, Aaron Courville, The MIT Press.
5. Christopher Bishop, *Pattern Recognition and Machine Learning*, Springer,2006.

CS5180	Natural Language Processing	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Understand language modeling with N-Grams.
CO2	Apply syntactic parsing to produce parse trees.
CO3	Analyze semantics with dense vectors.
CO4	Apply lexical semantics with word senses.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	1	2	3		
CO2	2	1	2	3		
CO3	2	1	2	2		
CO4	2	1	2	2		

Detailed Syllabus:
Introduction, Regular Expressions, Text Normalization and Edit Distance. Finite State Transducers, Language Modeling with N-Grams, Spelling Correction and the Noisy Channel, Naive Bayes Classification and Sentiment, Part-of-Speech Tagging, Syntactic Parsing, Statistical Parsing, Dependency Parsing, Vector Semantics, Lexicons for Sentiment and Affect Extraction, Information Extraction, Semantic Role Labeling and Argument Structure, Seq2seq Models and Machine Translation, Dialog Systems and Chatbots, Speech Recognition and Synthesis

Reading:

1. Daniel Jurafsky and James H. Martin, Speech and Language Processing (3rd ed.)
2. Allen, James, Natural Language Understanding, Second Edition, Benjamin/ Cumming, 1995.
3. Charniack, Eugene, Statistical Language Learning, MIT Press, 1993.
4. C. Manning and H. Schutze, “Foundations of Statistical Natural Language Processing”, MIT Press. Cambridge, MA:,1999

CS5181	Information Retrieval	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Understand the basics of Information retrieval like what is a corpus, what is precision and recall of an IR system
CO2	Apply the data structures like Inverted Indices used in Information retrieval systems
CO3	Understand the basics of web search
CO4	Develop different techniques for compression of an index including the dictionary and its posting list
CO5	Analyze different components of an Information retrieval system
CO6	Develop the ability to develop a complete IR system from scratch

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1			2	2		2
CO2	2		2	3		2
CO3	2		2	2		2
CO4	2		2	3		2
CO5	2		2	2		2
CO6	2		2	2		3

Detailed syllabus

Boolean retrieval, the term vocabulary and postings lists, Dictionaries and tolerant retrieval, Introduction to index-construction and index-compression. Scoring, term weighting and the vector space model, Computing scores in a complete search system, Evaluation in information retrieval, Introduction to Relevance feedback and query expansion. Probabilistic information retrieval, review of basic probability theory, the probability ranking, principle, the binary independence model. Language models for information retrieval, Language modeling versus other approaches to IR, Text classification and Naive Bayes, Bayesian Network approaches to IR. Vector space classification, Support vector machines and machine learning on documents, Flat clustering, Hierarchical clustering, Matrix decomposition and latent semantic indexing.

Introduction to Web search basics, Web crawling and indexes, Link analysis, Typical Assignments: Based on techniques studied, implementation of those techniques, study of research papers.

Reading:

1. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, *An Introduction to Information Retrieval*, Cambridge University Press, Cambridge, England, 2009
2. Stefan Büttcher, Charles L, A. Clarke, Gordon V. Cormack, *Information Retrieval: Implementing and evaluating search engines*, MIT Press, 2010
3. David A. Grossman, Ophir Frieder, *Information Retrieval: Algorithms and Heuristics*, Springer, 2004.
4. Frakes, *Information Retrieval: Data Structures and Algorithms*, Pearson, 2009.

CS5182	Soft Computing Techniques	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Understanding of optimizations problems, comprehend the fuzzy logic and the concept of fuzziness involved in various systems and fuzzy set theory
CO2	Understand the fundamental theory and concepts of neural networks and Identify different neural network architectures, algorithms, applications and their limitations.
CO3	Apply genetic algorithms and neural networks to solve real world problems
CO4	Apply soft computing techniques to solve engineering and other societal problems

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		2	2	2	
CO2	2	2	2	2		
CO3	2		2	2	2	
CO4	2	2	2	2	2	

Detailed syllabus

Overview of course and Basic of Soft Computing, Introduction of Neural Networks, Learning Process and Learning Task, Supervised Learning – Single and Multi – Layer Network, Associative Memory, Self-organizing Maps, Neuro-Dynamics, Hopfield Network, Fuzzy Logic and Systems-Fuzzy Sets and Membership Functions, Operations on Fuzzy Sets, Fuzzification. Fuzzy Numbers- Uncertain Fuzzy Values, Fuzzy Numbers and its L-R representation, Operations on Fuzzy Numbers. Fuzzy Relations, Fuzzy Inference Systems- Architecture of Fuzzy Inference System, Fuzzy Inference Rules and Reasoning, Defuzzification. Applications of Fuzzy Logic, Genetic algorithms and evolutionary computation. Applications of Genetic Algorithms & Hybrid Systems.

Reading:

1. R.A. Aliev, R.R. Aliev, *Soft Computing and Its Applications,* World Scientific Publications, 2001.
2. Roger Jang, Tsai Sun, Eiji Mizutani, *Neuro-Fuzzy and Soft Computing: A computational Approach to Learning & Machine Intelligence*, PHI, 2008.
3. Simon Haykin, *Neural Network: A Comprehensive Foundation*, PHI, 1999.
4. Kishan Mehtrotra, S. Ranka, *Elements of artificial Neural Networks*, Penram International Publishing (India), 2009
5. Timothy Ross*, Fuzzy Logic with Engineering Applications*, 3rd Edition, McGraw-Hill, 2010.
6. Bart Kosko, *Neural Networks and Fuzzy Systems*, PHI, 1994.

CS5183	Pattern Recognition	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Understand the generation of pattern features using various transforms based on data.
CO2	Analyze pattern features using probability theory and build classifiers using known probability distribution.
CO3	Build classifiers using non parametric methods and linear classifiers using perception model.
CO4	Develop linear, nonlinear classifiers using SVM model.
CO5	Build classifiers using syntactic model.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		1	1		
CO2	2		2	2		2
CO3	2		2	2		2
CO4	2		2	2		2
CO5	2		2	2		2

Detailed syllabus

Introduction: Pattern Recognition, Challenges, advance concepts, Probability Theory: Basic concepts, Random variable, Discrete distribution, Continuous distribution, Continuous distribution, Probability generating and moment generating functions, : Functions and operations on random variable, Estimation of distribution parameters, Confidence interval estimation, Hypothesis testing and type I and II errors, Goodness of Fit test, Joint distribution and correlation analysis, Classifiers: Building classifier for single feature using Bay's rule and error analysis, decision boundaries for various cases, Building classifiers using KNN, Parzen windows, Building linear classifiers, Building linear classifiers using SVM, SVM optimization, Building classifiers using syntactic methods, Context dependent classification, Feature generation, Unsupervised learning

Reading:

1. Kishore Trivedi, *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, John Wiley and Sons, New York, 2001.
2. Sergios Theodoridis, Konstantinos Koutroumbas, *Pattern Recognition*, 4th Edition, Elsevier, ISBN-9781597492720, Printbook, Release Date: 2008.
3. Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern Classification*, 2nd Edition, Wiley, ISBN: 978-0-471-05669

CS5184	Advance Data Mining	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Analyze Algorithms for sequential patterns.
CO2	Extract patterns from time series data.
CO3	Develop algorithms for Temporal Patterns.
CO4	Identify computing frameworks for Big Data analytics.
CO5	Extend the Graph mining algorithms to Web Mining.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3		2	2		3
CO2	3		2	2		3
CO3	2		2	2		3
CO4	2		2	2		3
CO5	3		2	2		3

Detailed syllabus

Review of Frequent Item set Mining. Sequential Pattern Mining concepts, primitives, scalable methods; Closed Sequential Patterns. Transactional Patterns and other temporal based frequent patterns, Mining Time series Data, Periodicity Analysis for time related sequence data, Trend analysis, Similarity search in Time-series analysis;. Graph Mining, Mining frequent sub-graphs, finding clusters, hub and outliers in large graphs, Graph Partitioning; Web Mining. Classification- Decision Tree learning, Bayesian Learning, Class Imbalance Problem. Review of Clustering methods. Trajectory Pattern Mining: Moving together patterns, Sequential Pattern mining from trajectories, Trajectory Clustering.

Reading:

1. Jiawei Han and M Kamber, *Data Mining Concepts and Techniques*, Second Edition, Elsevier Publication,2011.
2. Vipin Kumar, Pang-Ning Tan, Michael Steinbach, *Introduction to Data Mining,* Addison Wesley,2006.
3. Research Papers

CS6142	Comprehensive Viva	PCC	0 – 0 – 0	2 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Comprehend and correlate the understanding of all courses in post graduate curriculum of Computer Science and Engineering

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	3	3	3	3	3

CS6149	Dissertation Part-A	PRC	0 – 0 – 0	9 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Identify the problem of a research project through literature survey.
CO2	Analyze the technical feasibility of the project.
CO3	Propose a solution for the research problem.
CO4	Analyze and design the proposed solution using software engineering practices.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	3	2	2	2	1
CO2	2	3	2	2	2	1
CO3	2	3	2	2	2	1
CO4	2	3	2	2	2	1

CS6199	Dissertation Part-B	PRC	0– 0 – 0	18 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Synthesize and apply prior knowledge to designing and implementing solutions to open-ended computational problems while considering multiple realistic constraints
CO2	Design and Develop the software with software engineering practices and standards
CO3	Analyze Database, Network and Application Design methods
CO4	Evaluate various validation and verification methods
CO5	Practice CASE tools for solving software engineering CASE Studies
CO6	Analyze professional issues, including ethical, legal and security issues, related to computing projects

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3		2	2	2	1
CO2	2		2	2	2	1
CO3	2		2	2	2	1
CO4			2	2	2	1
CO5			2	2	2	1
CO6			2	2	2	1