NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL

RULES AND REGULATIONS

SCHEME OF INSTRUCTION AND SYLLABI

for M. Tech. CSE Program

(Effective from 2021-22)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

National Institute of Technology Warangal

Vision and Mission of the Institute

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 competitive environment.
- Allowing stake holders to share our reservoir of experience in education and knowledge for mutual enrichment in the field of technical education.
- Fostering product-oriented research for establishing a self-sustaining and wealth creating centre to serve the societal needs.

Department of Computer Science and Engineering

Vision and Mission of the Department

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 the industry.
- MS2: Providing state-of-art research facilities to generate knowledge and develop technologies in the thrust areas of computer science and engineering.
- MS3: Developing linkages with world class organizations to strengthen industry- academia relationships for mutual benefit.

Department of Computer Science and Engineering:

Brief about the Department:

The Department of Computer Science and Engineering was established in the year 1991. The department offers high quality undergraduate, postgraduate and doctoral programs. The B. Tech. (Computer Science and Engineering) program was started in the year 1983 with an intake of 20 students. The intake was subsequently increased to 120 in 2008. M. Tech (Computer Science and Engineering) program was started in 1987 with an intake of 18 and subsequently increased to 20 in 2008. M. Tech (Information Security) was introduced in the year 2008 Under ISEAP sanctioned by Ministry of Communication and Information Technology (MCIT), DOE, GOI, New Delhi with intake of 20. Later, it was renamed as Computer Science and Information Security. The Master of Computer Applications (MCA) program was started in 1986 with an intake of 30 and increased to 46 from 2008. B. Tech, M. Tech. (CSE) and M. Tech. (CSIS) programs were accredited in 2014 by NBA as per Washington Accord.

List of Programs offered by the Department:

Program	Title of the Program
B.Tech.	Computer Science and Engineering
M. Tech.	Computer Science and Engineering
	Computer Science and Information Security
MCA	Master in Computer Applications
Ph. D.	Computer Science and Engineering

Note: Refer to the Rules and Regulations for M. Tech. program (weblink) given on the institute website.

M.Tech. – Computer Science and Engineering

Program Educational Objectives

PEO-1	Design, develop and test software systems for engineering applications.
PEO-2	Analyze technical solutions to computational problems and develop efficient algorithms.
PEO-3	Work in multi-disciplinary teams to specify software requirements and to achieve project goals.
PEO-4	Communicate effectively and demonstrate professional ethics with societal responsibilities.
PEO-5	Engage in lifelong learning to keep pace with changing landscape of technologies for professional advancement.

Program Articulation Matrix

PEO / **Mission Statements**	**PEO-1**	**PEO-2**	**PEO-3**	**PEO-4**	**PEO-5**
Imparting quality education through well-designed curriculum in tune with the challenging software needs of the industry	3	2	1	1	2
Providing state-of-art research facilities to generate knowledge and develop technologies in the thrust areas of computer science and engineering.	2	2	2	-	2
Developing linkages with world class organizations to strengthen industry-academia relationships for mutual benefit	2	2	2	2	1

1-Slightly; 2-Moderately; 3-Substantially

M. Tech. – COMPUTER SCIENCE & ENGINEERING

Program Outcomes

PO-1	Engage in critical thinking and pursue investigations / research and development to solve practical problems.
PO-2	Communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.
PO-3	Demonstrate higher level of professional skills to tackle multidisciplinary and complex problems related to Computer Science and Engineering.
PO-4	Apply concepts of theoretical computer science to design software systems satisfying realistic, economic, social, safety and security constraints.
PO-5	Design and develop processes to meet targeted needs with optimum utilization of resources.
PO-6	Develop robust, reliable, scalable techniques and tools for knowledge-based systems.

MAPPING OF PROGRAM OUTCOMES WITH PROGRAME EDUCATIONAL OBJECTIVES

PO	PEO-1	PEO-2	PEO-3	PEO-4	PEO-5
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
Professional Core Courses (PCC)	29
Professional Elective Courses (PEC)	15
Seminar I and II	02
Comprehensive Viva-voce	02
Dissertation Work (12+20)	32
Total	80

SCHEME OF INSTRUCTION

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

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	Mathematics for Computer Science	1	1	0	2	PCC
3	CS5103	Advanced Operating Systems	3	0	0	3	PCC
4	CS5104	Data Science Fundamentals	3	0	0	3	PCC
5	CS5105	Advanced Software Engineering	3	0	0	3	PCC
6	CS5106	Machine Learning	3	0	0	3	PCC
7		Elective – 1	3	0	0	3	PEC
8	CS5107	Advanced Operating Systems Lab	0	0	2	1	PCC
9	CS5108	Computational Thinking	0	1	2	2	PCC
		Total	**19**	**2**	**4**	**23**	

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	Deep Learning	3	0	0	3	PCC
3		Elective – 2	3	0	0	3	PEC
4		Elective – 3	3	0	0	3	PEC
5		Elective – 4	3	0	0	3	PEC
6		Elective – 5	3	0	0	3	PEC
7	CS5153	Advanced Computer Networks Lab	0	0	2	1	PCC
8	CS5154	Deep Learning Lab	0	1	2	2	PCC
9	CS5198	Seminar - I	0	0	2	1	SEM
		Total	**18**	**1**	**6**	**22**	*

*Students of MTech (CSE) programme desirous of taking an exit will be awarded with **PG Diploma in CSE** if they complete Seminar-II (1 credit) after course work, followed by 2-months Project (4 credits) .

Note: PCC – Professional Core Courses
PEC – Professional Elective Courses

SCHEME OF INSTRUCTION

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

II – Year, I – Semester

S. No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1	CS6148	Seminar-II	0	0	2	1	SEM
2	CS6147	Comprehensive Viva	0	0	0	2	CVV
3	CS6149	Dissertation Work – Part A	0	0	0	12	DW
		Total	**0**	**0**	**2**	**15**	

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	20	DW
		Total				**20**	
		Total Credits				**80**	

Credit Distribution Table – Semester-wise and Category-wise

Cat code	I year		II year		Total
	Sem I	Sem II	Sem I	Sem II	
PCC	20	9			29
PEC	3	12			15
SEM		1	1		2
CVV			2		2
DW			12	20	32
Total	23	22	15	20	**80**

Professional Elective Courses:

I – Year, I – Semester

Semester	Elective Number	Course Code	Course Title
1	1	CS5111	Privacy Preserving Data Publishing
1	1	CS5112	Advanced Databases
1	1	CS5113	Computer Vision & Image Processing
1	1	CS5114	Cloud Computing
1	1	CS5115	Game Theory
1	1	CS5116	Distributed Computing
1	1	CS5117	Quantum Computing
1	1	CS5118	Cryptography and Network Security
1	1	CS5119	Advanced Artificial Intelligence
1	1	CS5120	Big Data
1	1	CS5121	Bio-Informatics
1	1	CS5122	Advanced Data Structure
1	1	CS5212	Mathematical models for Internet
1	1	CS5214	Computability and Complexity
1	1	CS5215	Information Systems Control and Auditing
1	1	CS5216	Probabilistic Algorithms
1	1	CS5217	Biometric Security
1	1	CS5218	Unix Internals
1	1	CS5219	Secure Software Engineering
[M.Tech. CSIS courses]			
1	1	CS5201	Web and Database Security
1	1	CS5202	Foundations of Cryptography
1	1	CS5220	Secure Cloud Computing
1	1	CS5222	Digital Video Processing
1	1	CS5223	Information Security and Secure Coding
1	1	CS5224	Scripting languages for information security
1	1	CS5225	Wireless and Mobile Networks

I – Year, II – Semester

2	2/3/4/5	CS5161	Service Oriented Architecture and Micro-Services
2	2/3/4/5	CS5162	Information Theory and Coding
2	2/3/4/5	CS5163	Software Reliability and Quality Management
2	2/3/4/5	CS5164	Research Study
2	2/3/4/5	CS5165	Formal Methods in Program Design
2	2/3/4/5	CS5166	Security and Privacy

2	2/3/4/5	CS5167	Cognitive Radio Networks
2	2/3/4/5	CS5168	Model Driven Frameworks
2	2/3/4/5	CS5169	Exploratory and Interactive Data Analysis
2	2/3/4/5	CS5170	Internet of Things
2	2/3/4/5	CS5171	Real Time Systems
2	2/3/4/5	CS5172	Optimization in Computer Science
2	2/3/4/5	CS5173	High Performance Computing
2	2/3/4/5	CS5174	Randomized and Approximation Algorithms
2	2/3/4/5	CS5175	Human Computer Interaction
2	2/3/4/5	CS5176	Social Media Analytics
2	2/3/4/5	CS5177	Models for Social Networks
2	2/3/4/5	CS5178	Advanced Compiler Design
2	2/3/4/5	CS5179	Software Defined Networks
2	2/3/4/5	CS5180	Natural Language Processing
2	2/3/4/5	CS5181	Information Retrieval
2	2/3/4/5	CS5182	Soft Computing Techniques
2	2/3/4/5	CS5183	Advanced Data Mining
2	2/3/4/5	CS5184	Fault Tolerant Systems
2	2/3/4/5	CS5185	Fog and Edge Computing
2	2/3/4/5	CS5186	Mobile Security
2	2/3/4/5	CS5187	Reinforcement Learning
[M.Tech. CSIS courses]			
2	2/3/4/5	CS5251	Network Security
2	2/3/4/5	CS5252	Data Privacy
2	2/3/4/5	CS5261	Foundations of Block Chain Technology
2	2/3/4/5	CS5262	Secure Operating Systems
2	2/3/4/5	CS5265	Secure Protocols for Electronic Commerce
2	2/3/4/5	CS5267	Network Coding
2	2/3/4/5	CS5268	Public Key Infrastructure and Trust Management
2	2/3/4/5	CS5269	Cyber laws and Intellectual Property Rights
2	2/3/4/5	CS5271	Digital Forensics
2	2/3/4/5	CS5272	Secure Dependable and Distributed Computing
2	2/3/4/5	CS5273	Data Hiding
2	2/3/4/5	CS5275	Information Security Risk Management
2	2/3/4/5	CS5276	Privacy Enhancing Technologies
2	2/3/4/5	CS5277	Security of E-Based Systems
2	2/3/4/5	CS5279	Cyber crime and Information Warfare
2	2/3/4/5	CS5282	Cyber Security
2	2/3/4/5	CS5285	Privacy and Security for online social networks

Note: A student is allowed to register a maximum of one elective from other departments and a maximum of two electives from other M.Tech. programmes offered by the Department of CSE.

DETAILED SYLLABUS

CS5101	ADVANCED ALGORITHMS	Credits 3-0-0: 3

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	Classify problems into different complexity classes corresponding to deterministic, approximation and parameterized algorithms.
CO3	Analyze the complexity of graph problems for different graph classes.
CO4	Analyze approximation algorithms and determine approximation factor.
CO5	Design and analyze efficient randomized algorithms.

Course Articulation Matrix:

	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	-	-

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Average case analysis of algorithms, Correctness of Master Theorem, Selection in Worst Case Linear Time, Large integer multiplications using FFT, Dynamic Programming - Matrix Chain Multiplication Problem, Optimal Binary Search Tree, Linear Algorithm for Domination in Trees, Maximum Cardinality Search and Chordal Graphs, Greedy Algorithm for Optimal Coloring of Chordal Graphs, NP-completeness, Efficient Reduction Proofs via Examples, Domination in Subclasses of Bipartite Graphs and Chordal Graphs, Exact Exponential Algorithm for Domination Problems, Treewidth, Parameterized Complexity Classes, APX-hardness and APX-completeness, Approximation Algorithm for Connected Dominating Set Problem, The Stable Marriage Problem, The Coupon Collector`s Problem.

Text Books/Reference Books/Online Resources:

1. *Introduction to Algorithms*, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, PHI, 2009, Third Edition.
2. *Fundamentals of Computer Algorithms*, Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, Universities Press, 2011, Second Edition.
3. *Algorithmic Graph Theory and Perfect Graphs*, Martin Charles Golumbic, 2004, Elsevier, Second Edition.
4. *Treewidth: Computations and Approximations*, Ton Kloks, Springer-Verlag, 1994.
5. *Graph Classes A Survey* : Andreas Brandstädt, Van Bang Le and Jeremy P.Spinard, SIAM, 1987.
6. *Algorithms and Complexity*, Herbert S. Wilf, AK Peters/CRC Press, 2002, Second Edition.
7. *Parameterized Complexity*, Rodney G. Downey and M. R. Fellows, Springer, 2012.
8. *Approximation Algorithms*, Vijay V. Vajirani, Springer, 2001.
9. *Randomized Algorithms*, Rajeev Motwani and Prabhakar Raghavan, Cambridge University Press, 1995.

Course Code: CS 5102	MATHEMATICS FOR COMPUTER SCIENCE	Credits 1-1-0: 2

Pre-Requisites: None

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

CO1	Apply the concepts of Linear Algebra.
CO2	Use the Mathematical concepts to solve real world problems.
CO3	Apply the concepts of Probability and Distribution for a given problem.
CO4	Use the concepts of Discrete Mathematics for solving the given problem.

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Mathematical Foundations: Introduction and Motivation.

Linear Algebra: System of linear equations, Matrices, Solving Systems of Linear Equations, Vector Spaces, Linear Independence, Basis and Rank, Linear Mappings.

Analytic Geometry: Norms, Inner Product, Lengths and Distances, Angles and Orthogonality, Orthonormal Basis, Orthogonal Complement, Fourier Transform.

Matrix Decomposition: Determinant and Trace, Eigen values and Eigen vectors, Eigen decomposition and Diagonalization, Single Value Decomposition, Matrix Approximation.

Vector Calculus: Differentiation of univariate functions, Partial Differentiation and Gradients, Gradients of vector-values functions, Gradients of Matrices.

Probability and Distribution: Construction of a Probability Space, Discrete and Continuous Probabilities, Sum Rule, Product Rule, Bayes Theorem, Summary Statistics and Independence.

Discrete Mathematics: Sets and Relations, Mathematical Logic and Induction, Elementary Combinatorics, Recurrence Relations, Lattices as Partially Ordered Sets, Graphs, Trees. Groups, Rings and Fields.

Text Books/Reference Books/Online Resources:

1. Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong, *Mathematics for Machine Learning*, Cambridge University Press, 2020. (for topics other than Discrete Mathematics)
2. Joe L. Mott, Abraham Kandel, Theodore P. Baker, *Discrete Mathematics for Computer Scientists and Mathematicians*, Second Edition, PHI, 2001.
3. J. P. Tremblay and R. Manohar, *Discrete Mathematical Structures with Applications to Computer Science*, MGH, 1997.
4. Kenneth H. Rosen, *Discrete Mathematics and its Applications with Combinatorics and Graph Theory*, Seventh Edition, MGH, 2011.

Course Code: CS 5103	ADVANCED OPERATING SYSTEMS	Credits 3-0-0: 3

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

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

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 activations Signals: 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 case studies

Synchronization and Multiprocessing : Introduction, Synchronization in Traditional UNIX Kernels, Multiprocessor Systems, Multiprocessor synchronization issues, Semaphores, spin locks, condition variables, Read-write locks, Reference counts, Practice and solving problems on synchronization, process scheduling and threads.

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 FS interface.

Text Books/Reference Books/Online Resources:

1. Thomas M. Cover and Uresh Vahalia, *UNIX Internals*, Pearson Education, 2005.
2. Richard Stevens, Stephen A. Rago, *Advanced Programming in the UNIX Environment*, Pearson

Course Code: CS 5104	DATA SCIENCE FUNDAMENTALS	Credits 3-0-0: 3

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 and data visualization over massive data.

Course Articulation Matrix:

	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

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Overview of Random variables and distributions.

Statistical learning: Assessing model accuracy, Bias-Variance Trade-Off, 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; association rules and correlations; PCA and SVD.

Classification: Linear and Logistic Regression, Bayesian Learning, LDA, QDA, K-Nearest Neigbhour, and comparison of classification methods.

Data Visualization and Graphical Analysis: Visualized exploratory data Analysis, Histograms and frequency polygons, Box-plots, Quartiles, Scatter Plots, Heat Maps. Matrix visualization, Scientific Design Choices in Data Visualization, Higher-dimensional Displays and Special Structures, Visual data mining.

Data Wrangling: Data Acquisition, Data Formats, Imputation, split-apply-combine paradigm.

Descriptive Analytics: Data Warehousing and OLAP, Data Summarization, Data de- duplication, Data Visualization using CUBEs.

Text Books/Reference Books/Online Resources:

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)
4. Chun-houh Chen, Wolfgang Hardle, Antony Unwin, Handbook of Data Visualization, Springer, 2008
5. https://www.kdnuggets.com/topic/data-science
6. https://www.kdnuggets.com/topic/data-visualization

Course Code: CS 5105	ADVANCED SOFTWARE ENGINEERING	Credits 3-0-0: 3

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

Course Articulation Matrix:

	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
CO5	2	3	2	2	2	3

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5106	MACHINE LEARNING	Credits 3-0-0: 3

Pre-Requisites: None

Course Outcomes:

At the end of the course, the student will be able to

CO1	Construct prediction models for statistical data
CO2	Design Multi-Layer neural network to solve Supervised Learning problems
CO3	Classify non-linear data like face recognition, disease prediction
CO4	Apply Genetic Algorithm for optimization problems.
CO5	Design applications like games and agent-based controllers

Course Articulation Matrix:

	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		2		1	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to machine learning, issues related to machine learning: pre-processing, inductive bias, variance, feature extraction, feature selection techniques. Different types of learning, training and testing, hypothesis and cost function. Mathematics for machine learning. Classification techniques: DECISION TREES (DT)-construction of decision trees using different algorithms, Regression tree, tree pruning, rule extraction from trees, multivariate trees. Ensemble learning: Bagging and boosting and different techniques of bagging and boosting. Artificial neural networks (ANN): different learning rules, single-layer perceptron, multi-layer neural nets, backpropagation algorithm, feed-forward networks, network training, radial basis function networks, recurrent neural networks. Bayesian learning: probabilistic reasoning: prior, likelihood and posterior, belief networks: modeling independencies, Markov equivalence in belief networks, hidden Markov models (HMM).Naïve Bayes classifier, learning with hidden variables, Expectation Maximisation (EM). - GENETIC ALGORITHMS – an illustrative example, Hypothesis space search, Genetic Programming, Models of Evolution and Learning. Instance-based learning: Nearest-Neighbour classification, condensed-neighbour classification. Unsupervised linear dimensionality reduction: principal component analysis(PCA), PCA vs singular value decomposition, working on high-dimensional data, latent-semantic analysis: information retrieval. Supervised linear dimensionality reduction: Fisher's linear discriminant. Kernel methods: dual representations, kernel construction, learning with hyperparameters, support vector machines (maximum margin classifier), linear and multiclass SVMs. REINFORCEMENT LEARNING - The Learning Task, Q Learning, Nondeterministic rewards and actions, Temporal difference learning, Generalizing from examples, relationship to Dynamic Programming.

Text Books/Reference Books/Online Resources:

1. Tom M .Mitchell, Machine Learning, McGraw Hill, 1997.
2. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

Course Code: CS 5107	ADVANCED OPERATING SYSTEMS LAB	Credits 0-0-2: 1

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.

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

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

Text Books/Reference Books/Online Resources:

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

Course Code: CS 5108	COMPUTATIONAL THINKING	Credits 0-1-2: 2

Pre-Requisites: None

Course Outcomes:

At the end of the course, the student will be able to

CO1	Choose appropriate data structures to solve problem
CO2	Solve mathematical and real world problems
CO3	Implement Linear and Logistic Regression methods
CO4	Implement Graph algorithms

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Problems on Arrays and Linked List, Implementation of Stack and Queues, Types of Trees, AVL Tree operations, red-black tree operations, B-tree operations, In-place merging, radix sort, shell sort, BFS and DFS traversal of graphs, solution to travelling salesperson problem, finding the connected components, graph colouring, min-cut partitioning of a graph. Shortest path algorithms, Minimum spanning tree algorithms.

Practice Linear Algebra: System of linear equations, Matrices, Solving Systems of Linear Equations, Vector Spaces, Linear Independence, Basis and Rank, Linear Mappings.

Problem solving on Matrix Decomposition: Determinant and Trace, Eigen values and Eigen vectors, Eigen decomposition and Diagonalization, Single Value Decomposition, Matrix Approximation.

Solving problems on the following data computation topics: Data pre-processing, Statistical learning: Assessing model accuracy, Descriptive Statistics, Dependent and Independent events; Linear Regression: Simple and multiple linear regressions, Linear regression with K-nearest neighbors. Simple Hypothesis Testing, Student's t-test, paired t and U test, correlation and covariance, tests for association; association rules and correlations; PCA and SVD.

Classification: Linear and Logistic Regression, Bayesian Learning, LDA, QDA, K-Nearest Neigbhour, and comparison of classification methods

Text Books/Reference Books/Online Resources:

1. Data structures and algorithm analysis in C++(Java): Mark Weiss
2. A. M. Tenenbaum, Y. Langsam, and M. J. Augenstein, Data Structures Using C and C++, Prentice Hall, 2/e, 1995
3. 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)
4. Mark Gardener, *Beginning R The statistical Programming Language*, Wiley,2015.

Course Code: CS 5151	ADVANCED COMPUTER NETWORKS	Credits 3-0-0: 3

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 data networks
CO5	Analyze Data Center Networks and Optical Networks

Course Articulation Matrix:

	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	

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Network Architecture, Performance: Bandwidth and Latency, High Speed Networks, Network-Centric View, Error Detection, Reliable Transmission, Ethernet and Multiple Access Networks, Overlay Networks: Routing Overlays, Peer-to-Peer Networks and Content Distribution Networks, Client-Server Networks, Delay-Tolerant Networks, Switching: Circuit-Switched Networks, Datagram Networks, Virtual-Circuit Networks, Message-Switched Networks, Asynchronous Transfer Mode: Evolution, Benefits, Concepts, Exploring Broadband Integrated Services Digital Network, Layer and Adaptation Layer, IPv4: Address Space, Notations, Classful, Classless, Network Address Translation, Datagram, Fragmentation and Checksum IPv6 Addresses: Structure, Address Space, Packet Format and Extension Headers, ICMP, IGMP, ARP, RARP, Congestion Control and Resource Allocation: Problem, Issues, Queuing, TCP Congestion Control, Congestion-Avoidance Mechanisms and Quality of Service, Internetworking: Intra-Domain and Inter-Domain Routings, Unicast Routing Protocols: RIP, OSPF and BGP, Multicast Routing Protocols: DVMRP, PIM-DM, PIM-SM, CBT, MSDP and MOSPF, Spanning Tree Algorithm, Optical Networking: SONET/SDH Standards, Traffic Engineering: Requirement, Traffic Sizing, Characteristics, Protocols, Time and Delay Considerations, Connectivity, Availability, Reliability and Maintainability and Throughput, Multimedia Over Internet: Transmission, IP Multicasting and VoIP, Domain Name System: Name Space, Domain Name Space, Distribution, Domains, Resolutions and Dynamic Domain Name System, SNMP, Security: IPSec, SSL/TLS, PGP and Firewalls, Datacenter Design and Interconnection Networks.

Text Books/Reference Books/Online Resources:

1. Larry L. Peterson and Bruce S. Davie, *Computer Networks: A System Approach*, Fifth Edition, Morgan Kaufmann, Elsevier, 2012.
2. Behrouz A. Forouzan, *Data Communications and Networking*, McGraw Hill, Fifth Edition, 2017.
3. Chwan-Hwa (John) Wu, J. David Irwin, *Introduction to Computer Networks and Cyber Security*, CRC press, Taylor & Francis Group,2014
4. Andrew S. Tanenbaum, David J. Wetherall, *Computer Networks*, Pearson, 5th Edition, 2014.
5. G. Wright and W. Stevens, *TCP/IP Illustrated*, Volume 1 and Volume 2, Addison-Wesley, 1996.
6. Dayanand Ambawade, Deven Shah, Mahendra Mehra and Mayank Agarwal, Advanced Computer Network, Dreamtech Press, 2016.
7. R. Srikant, The Mathematics of Internet Congestion Control, Springer, 2004.
8. J. L. Boudec and P. Thiran, Network Calculus, Springer, 2011.

Course Code: CS 5152	DEEP LEARNING	Credits 3-0-0: 3

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.

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to Biological Neurons, Artificial Neural Networks, McCulloch Pitts Neuron, Learning processes, Perceptron, Perceptron convergence theorem, XOR problem, Multilayer perceptron, Back Propagation (BP) Learning, Activation functions: Sigmoid, Linear, Tanh, ReLU, Leaky ReLU, SoftMax, loss functions, First and Second order optimization methods, Optimizers: Gradient Descent (GD), Batch Optimization, Momentum Based GD, Stochastic GD, AdaGrad, RMSProp, Adam; Introduction to Self Organizing Maps; Sequence to sequence models, RNN, Vanishing and Exploding Gradients, GRU, LSTM for NLP Applications; Convolutional Neural Network, Building blocks of CNN, Transfer Learning; Regularization: Bias Variance Tradeoff, L2 regularization, Early stopping, Dataset augmentation, Parameter sharing and tying, Dropout; Autoencoders : Unsupervised Learning with Deep Network, Autoencoders, Stacked, Sparse, Denoising Autoencoders, Variational Autoencoders; Recent Trends in Deep Learning Architectures, Residual Network, Skip Connection Network, GoogleNet, DensenNet, SqueezNet, MobileNet, NasNet Models; Classical Supervised Tasks with Deep Learning, Segmentation Unet, FCN models, Object Localization (RCNN), FRCNN with Applications; Transformer, Generative Adversarial Network, Design own neural network models on Image, vision and NLP Applications.

Text Books/Reference Books/Online Resources:

1. Deep Learning- Ian Good felllow, Yoshua Benjio, Aaron Courville, The MIT Press.
2. Christopher Bishop, Pattern Recognition and Machine Learning, Springer,2006.
3. Simon Haykin, “Neural Networks, A Comprehensive Foundation”, 2nd Edition, Addison Wesley Longman, 2001.

Course Code: CS 5153	ADVANCED COMPUTER NETWORKS LAB	Credits 0-0-2: 1

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

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Assignment-1 Implementation of client server programs for different network applications
Assignment-2 Study and analysis of the network using Wireshark network protocol analyzer
Assignment-3 Implementation of topology generation for network simulation
Assignment-4 Implementation of queuing management
Assignment-5 Implementation of MAC-layer protocols
Assignment-6 Implementation of routing protocols
Assignment-7 Implementation of transport-layer protocols
Assignment-8 Implementation of network security mechanisms
Assignment-9 Implementation of intermodal traffic systems using simulation of urban mobility
Assignment-10 Study and analysis of cloud and fog-based simulation tools and their applications

Text Books/Reference Books/Online Resources:

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

Course Code: CS 5154	DEEP LEARNING LAB	Credits 0-1-2: 2

Pre-Requisites: None

Course Outcomes:

At the end of the course, the student will be able to

CO1	Implement Multilayer Feed Backward Neural network on MNIT digits dataset
CO2	Build RNN, LSTM, BiLSTM Networks for time series analysis classification problems.
CO3	Design Autoencoders to solve Unsupervised Learning problems
CO4	Implement Classical Supervised Tasks for Image Denoising, Segmentation and Object detection problems.

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

1. Implement perceptron learning algorithm and attempt to solve two input i) AND gate ii) Or Gate iii) EXOR gate problems.
2. Design and implement a perceptron learning algorithm and attempt to solve XOR problem
3. Implement a Multilayer Feed Backward Neural network algorithm on MNIT digits dataset.
4. Build your own Recurrent networks and Long short-term memory networks on IMDB movie reviews classification data.
5. Design and implement a BiLSTM and BERT on given a product review dataset to classify the review rating from 1 to 5 classes
6. Design and implement Autoencoders for credit card fraud detection.
7. Design and implement a Convolutional Neural Network for image classification on the Fashion-MNIST dataset.
8. Implement a VGG19 model for image classification with and without Transfer Learning on Grocery dataset.
9. Implement a U-Net convolutional neural network model on segmentation of electron microscopic (EM) images of the brain dataset.
10. Implement a FRCNN algorithm for object detection on small object dataset.

Text Books/Reference Books/Online Resources:

1. Deep Learning- Ian Goodfelllow, Yoshua Benjio, Aaron Courville, The MIT Press.
2. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
3. Simon Haykin, “Neural Networks, A Comprehensive Foundation”, 2nd Edition, Addison Wesley Longman, 2001.

Course Code: CS 5198	SEMINAR-I	Credits 0-0-2: 1

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 communication to audience in an effective manner
CO2	Practice the learning by self study

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

Course Code: CS 6148	SEMINAR-II	Credits 0-0-2: 1

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 communication to audience in an effective manner
CO2	Practice the learning by self study

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

Course Code: CS 6147	COMPREHENSIVE VIVA	Credits 0-0-0: 2

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

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

Course Code: CS 6149	DISSERTATION WORK – PART A	Credits 0-0-0: 12

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, design and implement the proposed solution using software engineering practices

Course Articulation Matrix:

	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

1 - Slightly; 2 - Moderately; 3 – Substantially

Course Code: CS 6199	DISSERTATION WORK – PART B	Credits 0-0-0: 20

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 the solution through 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

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

Course Code: CS 5111	PRIVACY PRESERVING DATA PUBLISHING	Credits 3-0-0: 3

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

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5112	ADVANCED DATABASES	Credits 3-0-0: 3

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.

Course Articulation Matrix:

	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

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5113	COMPUTER VISION & IMAGE PROCESSING	Credits 3-0-0: 3

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

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5114	CLOUD COMPUTING	Credits 3-0-0: 3

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.

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Cloud Computing at a Glance, Historical Development, Building Cloud Computing Environments, Principles of Parallel and Distributed Computing, Virtualization, Cloud Computing Architecture, Cloud Application Programming, Concurrent Computing, High-Throughput Computing, Data-Intensive Computing, Cloud Platforms and New Developments, Cloud Applications, Advanced Topics in Cloud Computing.

Text Books/Reference Books/Online Resources:

1. Rajkumar Buyya, Christian Vecchiola and S. Thamarai Selvi, Mastering Cloud Computing: Foundations and Applications Programming, Morgan Kaufmann, 2013.
2. Kai Hwang, Geoffrey C. Fox and Jack J. Dongarra, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Morgan Kaufmann, 2012.
3. Judith Hurwitz, R Bloor, M Kanfman, F Halper, Cloud Computing for Dummies, 1st Edition, Wiley Publishers, 2009.
4. Gautam Shroff, Enterprise Cloud Computing, Cambridge, 2010.
5. Ronald Krutz and Russell Dean Vines, Cloud Security, 1st Edition, Wiley, 2010.

Course Code: CS 5115	GAME THEORY	Credits 3-0-0: 3

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

Course Articulation Matrix:

	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
CO5	2		3	2	2	2

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5116	DISTRIBUTED COMPUTING	Credits 3-0-0: 3

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

Course Articulation Matrix:

	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	

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5117	QUANTUM COMPUTING	Credits 3-0-0: 3

Pre-Requisites: 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

Course Articulation Matrix:

	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		

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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

Course Code: CS 5118	CRYPTOGRAPHY AND NETWORK SECURITY	Credits 3-0-0: 3

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

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

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 and PGP.

Text Books/Reference Books/Online Resources:

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

Course Code: CS 5119	ADVANCED ARTIFICIAL INTELLIGENCE	Credits 3-0-0: 3

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.

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

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..

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5120	Big Data	Credits 3-0-0: 3

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 and medicine
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

Course Articulation Matrix:

	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

1 - Slightly; 2 - Moderately; 3 – Substantially

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 Massive Datasets.
3. Jimmy Lin and Chris Dyer, Data Intensive Text Processing with Map Reduce, Pre-production manuscript, Downloadable from Internet.
4. Johannes Ledolter, Data Mining and Business Analytics with R, Wiley, 2013.

Course Code: CS 5121	BIO-INFORMATICS	Credits 3-0-0: 3

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.

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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

Course Code: CS 5122	ADVANCED DATA STRUCTURES	Credits 3-0-0: 3

Pre-Requisites: None

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

CO1	Determine efficient data structures to solve real world problems
CO2	Analyse and apply trees and heaps
CO3	Analyse external data structures
CO4	Implement Graph algorithms

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Binary search trees, search efficiency, insertion and deletion operations, importance of balancing, AVL trees, searching, insertion and deletions in AVL trees, Tries, 2-3 tree, B-tree, B+ tree, Splay tree, Red-Black tree, k-dimensional tree, Data Structure for Disjoint Sets
Heaps as priority queues, heap implementation, insertion and deletion operations, binary heaps, binomial and Fibonacci heaps, heapsort, heaps in Huffman coding.
File structures like sequential, indexed sequential and direct file and its processing procedure.
Graph algorithms: DFS, BFS, topological sorting, shortest path algorithms, and network flow problems, string algorithms, suffix trees, geometric algorithms.
Run Lists, Tape Sorting, Sorting on Disks, Generating Extended Initial Runs. Distribution-dependent Hashing Function, Dynamic Hashing Techniques, Probabilistic Data Structures, Bloom Filter, Min Hash, Skip List, Kinetic Data Structure, Kinetic Heap, Kinetic Shorted List, Kinetic Minimum Spanning Tree,

Text Books/Reference Books/Online Resources:

1. Introduction to algorithms: Cormen, Leiserson, Rivest and Stein
2. Data structures and algorithm analysis in C++(Java): Mark Weiss
3. A. M. Tenenbaum, Y. Langsam, and M. J. Augenstein, Data Structures Using C and C++, Prentice Hall, 2/e, 1995

Course Code: CS 5161	SERVICE ORIENTED ARCHITECTURE & MICROSERVICES	Credits 3-0-0: 3

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

Course Articulation Matrix:

	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

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5162	INFORMATION THEORY AND CODING	Credits 3-0-0: 3

Pre-Requisites:

Course Outcomes:

At the end of the course, the student will be able to

CO1	Understand the foundation of information theory
CO2	Design various coding techniques for various channel conditions
CO3	Analyze the channel performance using information theory
CO4	Implement various error control techniques

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Mathematical Foundation of Information Theory in communication system. Measures of Information-Self information, Shannon's Entropy, joint and conditional entropies, mutual information and their properties. Discrete memoryless channels: channel capacity, fundamental theorem of information theory. Coding Theory: Huffman codes, Shannon-Fano coding, robustness of coding techniques, Information measure-noiseless coding, Error correcting codes: minimum distance principles, Hamming bound, general binary code, group code, linear group code Convolution encoding: algebraic structure, Gilbert bound Threshold decoding: threshold decoding for block codes Cyclic binary codes: BCH codes, generalized BCH code and decoding, optimum codes, concepts of non-cyclic codes.

Text Books/Reference Books/Online Resources:

1. Thomas M. Cover and Joy A. Thomas, Elements of Information Theory, 2nd Edition, Wiley, 2006.
2. J. H. van Lint: Introduction to Coding Theory, Third Edition, Springer, 1998.
3. M. Medard and A. Sprintson, (editors): Network Coding – Fundamentals and Applications, Acadamic Press, 2012.

Course Code: CS 5163	SOFTWARE RELIABILITY & QUALITY MANAGEMENT	Credits 3-0-0: 3

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)

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5164	RESEARCH STUDY	Credits 3-0-0: 3

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.

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

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

Course Code: CS 5165	FORMAL METHODS IN PROGRAM DESIGN	Credits 3-0-0: 3

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

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5166	SECURITY AND PRIVACY	Credits 3-0-0: 3

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.

Course Articulation Matrix:

	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		

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5167	COGNITIVE RADIO NETWORKS	Credits 3-0-0: 3

Pre-Requisites: 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.

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5168	MODEL DRIVEN FRAMEWORKS	Credits 3-0-0: 3

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.

Course Articulation Matrix:

	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		

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5169	EXPLORATORY AND INTERACTIVE DATA ANALYSIS	Credits 3-0-0: 3

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 fields such as diagnosis; forecasting; planning; decision making.
CO3	Highlight the statistical features of observed datasets
CO4	Apply Data Analysis to various sizes and complexity of the data sets.

Course Articulation Matrix:

	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

1 - Slightly; 2 - Moderately; 3 – Substantially

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 Value Decomposition, Advanced exploratory data analysis, Data classification or clustering, Data input.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5170	INTERNET OF THINGS	Credits 3-0-0: 3

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

Course Articulation Matrix:

	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	

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5171	REAL TIME SYSTEMS	Credits 3-0-0: 3

Pre-Requisites: None

Course Outcomes:

At the end of the course, the student will be able to

CO1	Understand the requirements of a real-time application and analyze the performance of different task scheduling algorithms for real-time systems.
CO2	Understand the basic concepts of fault-tolerance and different fault-tolerance techniques available for real- time systems.
CO3	Use simulated software to develop and test different fault tolerant models.
CO4	Understand the concept of embedded systems and use various software tools for development of embedded systems.

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to Real-Time systems, applications of Real-Time systems, basic model of Real-Time systems, characteristics of Real-Time systems, types of Real-Time systems: hard, firm, soft, timing constraints, modelling timing constraints.
Real-Time task scheduling: basic concepts, clock driven scheduling, table driven scheduling, cyclic, schedulers, hybrid schedulers, event driven scheduling, EDF Scheduling, RMA, DMA, resource sharingamong RT tasks, Priority inversion, Priority Inheritance Protocol, Highest Locker Protocol, Priority Ceiling Protocol.
Introduction to Fault Tolerant Computing: Basic concepts and Fault tolerant scheduling of tasks Faults and their manifestations, Fault/error modelling, Reliability, availability and maintainability analysis, System evaluation, performance reliability trade-offs. System level fault diagnosis, Hardware and software redundancy techniques. Fault tolerant system design methods, Mobile computing and Mobile communication environment, Fault injection methods, Software fault tolerance, testing of fault tolerant software, fault modeling, built in self-test, data compression, error correcting codes, simulation of software/hardware, fault tolerant system design, CAD tools for design for testability.
Real-Time Embedded system, Need of well tested and debugged RTOS, Introduction to C/OS II. Case Studies of programming with RTOS: Smart card embedded system, Hardware and Software co-design: specification and design of an embedded system use of software tools for development of an embedded system. Recent advances in embedded applications.

Text Books/Reference Books/Online Resources:

1. R. Mall, Real-Time Systems, Pearson, 2007
2. P. A. Laplante, Real-Time Systems Design & Analysis, Willey, 2011
3. S. V. Iyer & P. Gupat, Embedded Real-Time System Programming, Tata McGraw Hill, 2004
4. R. Kamal, Embedded System Architecture, Programming and Design, Tata McGraw Hill, 2007

Course Code: CS 5172	OPTIMIZATION IN COMPUTER SCIENCE	Credits 3-0-0: 3

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

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5173	HIGH PERFORMANCE COMPUTING	Credits 3-0-0: 3

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.

Course Articulation Matrix:

	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		

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5174	RANDOMIZED AND APPROXIMATION ALGORITHMS	Credits 3-0-0: 3

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.

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

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-hardAbsoluteApproximations$^{\varepsilon}$-approximations, Polynomial time approximations schemes, Scheduling Independent Tasks , 0/1 Knapsack, Fully Polynomial time approximations scheme , Rounding , Interval Partitioning , Separation, probabilistically goodalgorithms.

Text Books/Reference Books/Online Resources:

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

Course Code: CS 5175	HUMAN COMPUTER INTERACTION	Credits 3-0-0: 3

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 architectural structures.

Course Articulation Matrix:

	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		

1 - Slightly; 2 - Moderately; 3 – Substantially

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 and scanning.

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, HCI patterns.

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.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5176	SOCIAL MEDIA ANALYTICS	Credits 3-0-0: 3

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

Course Articulation Matrix:

	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

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to social network and media analysis – Examples of Social Media and their characteristics, Society as a graph, candidates of social media and network data for analysis, Random graphs with general degree distributions, models of network formation, Properties of Large-Scale Networks: Six-degree separation, Scale-free distributions, Small-world effect, and strong community structure – strong and weak ties;

Social relatedness: networks and centrality Measures - degree, closeness, betweenness, edge betweenness, eccentricity, clustering coefficient, eigenvector; social media analytical applications.

Community Detection and graph-based clustering: communities in social media, node-centric community detection, group-centric community detection, network-centric community detection, hierarchy-centric community detection, Topology discovery, Community Evaluation;

Link Prediction: Challenges in link prediction, link prediction methods and algorithms, clustering approaches for link prediction;

Social Listening and Sentiment Analysis: Sentiments and Opinions, lexicon based methods, machine learning based methods, feature-based sentiment analysis, slang sentiment analysis;

Social Recommendation Systems: Classical recommendation algorithms – content-based methods, Collaborative Filtering, extending individual recommendation to groups of individuals; Recommendations using social context – using social context alone, extending classical methods with social context - Social Recommendation Using collaborative filtering, community detection and probabilistic matrix factorization, recommendations constrained by social context; evaluating recommendations.

Social Signal Processing: Understanding social interactions, social media content, speech and facial actions as social signals, Automatic analysis of social emotions, multimodal conversational analysis, SSP applications.

Information Diffusion in Social Media: Herd Behaviour – Bayesian Modeling of Herd Behaviour, Intervention; Information Cascades – Independent Cascade Model (ICM), Maximizing the spread of cascades, Intervention; Diffusion of Innovations – Innovation characteristics, diffusion of innovations models, modelling diffusion of Innovations, Intervention; Epidemics.

Text Books / Reference Books / Online Resources:

1. Reza Zafarani, Mohammad Ali Abbasi, Huan Liu, "*Social Media Mining – An Introduction*", Cambridge University Press, 2014.
2. Charu C Aggarwal (Ed.), "*Social Network Data Analytics*", Springer, 2011
3. Hansen, Derek, Ben Sheiderman, Marc Smith., "*Analyzing Social Media Networks with NodeXL: Insights from a Connected World*", Morgan Kaufmann, 2011.

Course Code: CS 5177	MODELS FOR SOCIAL NETWORKS	Credits 3-0-0: 3

Pre-Requisites: None

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.

Course Articulation Matrix:

	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

1 - Slightly; 2 - Moderately; 3 – Substantially

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 Network X library, a distribution of Python for scientific computing and visualization.
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.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5178	ADVANCED COMPILER DESIGN	Credits 3-0-0: 3

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

Course Articulation Matrix:

	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	

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5179	SOFTWARE DEFINED NETWORKS	Credits 3-0-0: 3

Pre-Requisites: None

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

CO1	Differentiate between traditional networks and software defined networks and understand the key benefits and use cases of SDN.
CO2	Interpret the SDN data plane devices and OpenFlow Protocols
CO3	Implement the operation of SDN control plane with different controllers
CO4	Apply techniques that enable applications to control the underlying network using SDN
CO5	Evaluate Network Functions Virtualization components and their roles in SDN

Course Articulation Matrix:

	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	

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Networking Basics: Switching, Addressing, Routing
SDN Background and Motivation: Evolving network requirements-The SDN Approach: Requirements, SDN Architecture, and Characteristics of Software-Defined Networking.
SDN Data plane and OpenFlow: Data plane Functions, Data plane protocols, OpenFlow: Switch-Controller Interaction, Flow Table, Packet Matching, Actions and Packet Forwarding Flow Table Structure, Flow Table Pipeline, The Use of Multiple Tables, Group Table, Extensions and Limitations, Data plane scalability.
SDN Control Plane: SDN Control Plane Architecture: Control Plane Functions, Southbound Interface, Northbound Interface, Routing, Cooperation and Coordination among Controllers, Controller placement problem, SDN controllers: OpenDaylight, Ryu, ONOS, Floodlight, Control plane scalability, fault tolerance.
SDN Application Plane: SDN Application Plane Architecture: Northbound Interface, Network Applications, User Interface- Network Services Abstraction Layer: Abstractions in SDN, Frenetic-Traffic Engineering Measurement and Monitoring, Security, network updates, SDN usecases: Traffic engineering, network management, network virtualization.
Network Functions Virtualization: Background and Motivation for NFV- NFV Principles, High-Level NFV Framework, NFV Benefits and Requirements- SDN vs. NFV, Network Functions, Service Creation and Chaining, NFV Orchestration, VNF deployment, Service function Chain Deployment. NFV Reference Architecture: NFV Management and Orchestration.
Emerging SDN Models: Protocol Models: NETCONF, BGP, MPLS, Controller Models, Application Models: Proactive, Declarative, External, SDN in Datacenters: Multitenancy, Failure Recovery, SDN in Internet eXchange Points (IXPs)

Text Books/Reference Books/Online Resources:

1. Paul Goransson Chuck Black Timothy Culver: Software Defined Networks: A Comprehensive Approach, Morgan Kaufmann, 2016.
2. Ken Gray Thomas Nadeau: Network Function Virtualization, Morgan Kaufmann, 2016.
3. Larry Peterson , Carmelo Cascone , Bruce Davie: Software-Defined Networks: A Systems Approach, Systems Approach, 2021

Course Code: CS 5180	NATURAL LANGUAGE PROCESSING	Credits 3-0-0: 3

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.

Course Articulation Matrix:

	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		

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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

Course Code: CS 5181	INFORMATION RETRIEVAL	Credits 3-0-0: 3

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

Course Articulation Matrix:

	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

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5182	SOFT COMPUTING TECHNIQUES	Credits 3-0-0: 3

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

Course Articulation Matrix:

	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	

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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.

Course Code: CS 5183	ADVANCED DATA MINING	Credits 3-0-0: 3

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.

Course Articulation Matrix:

	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

1 - Slightly; 2 - Moderately; 3 – Substantially

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.

Text Books/Reference Books/Online Resources:

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

Course Code: CS 5184	FAULT TOLERANT SYSTEMS	Credits 3-0-0: 3

Pre-Requisites: None

Course Outcomes:

At the end of the course, the student will be able to

CO1	Understand the risk of computer failures and their comparison with other equipment failures
CO2	Know the different advantages and limits of fault avoidance and fault tolerance techniques
CO3	Analyze cost dependability trade-offs and the limits of computer system dependability
CO4	Gain knowledge in sources of faults and their prevention and forecasting
CO5	Analyze fault-tolerant or non-fault-tolerant on the basis of dependability requirements.

Course Articulation Matrix:

	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		

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to Fault Tolerant Systems, Fault Classification, Types of Redundancy, Traditional and Network Measures, Hardware Fault Tolerance, Rate of Hardware Failures, Failure Rate, Reliability, Availability, Mean Time To Failure, Canonical and Resilient Structures: Series and Parallel Systems, Non-Series/Parallel Systems, M of N Systems, Triple Modular Redundant Structure, Voters, Variations on N-Modular Redundancy and Duplex Systems. Fault Tolerant Processor Level Techniques: Watchdog Processor and Simultaneous Multithreading, Byzantine Failures. Introduction to Information Redundancy, Coding: Parity, Checksum, M of N, Berger, Cyclic, Arithmetic, Resilient Disk Systems: RAID Level 1, Level 2, Level 3, Level 4 and Level 5, Data Replication: Non-Hierarchical, Hierarchical Organization and Primary-Backup Approach, Algorithm Based Fault Tolerance. Introduction to Software Fault Tolerance, Acceptance Tests, Single Version Fault Tolerance: Wrappers, Software Rejuvenation, Data Diversity and Software Implementable Hardware Fault Tolerance, N Version Programming: Consistent Comparison Problem and Version Independence, Recovery Block Approach, Preconditions, Postconditions and Assertions, Exception Handling, Software Reliability Models: Jelinski-Moranda, Littlewood-Verrall, Musa-Okumoto and Schneidewind Model, Remote Procedure Cells. Introduction to Checkpointing, Level, Latency, Overhead, Optimal Checkpointing: Reducing Overhead and Reducing Latency, Cache Aided Rollback Error Recovery, Checkpointing in Distributed Systems: Domino Effect and Livelock, Coordinated Checkpointing Algorithm, Time Based Synchronization, Diskless Checkpointing and Message Logging, Checkpointing in Shared Memory Systems: Bus-Based Coherence and Directory-Based Protocol, Checkpointing in Real Time Systems, Other Uses of Checkpointing.

Text Books/Reference Books/Online Resources:

1. I. Koren and C. M. Krishna, Fault Tolerant Systems, Morgan Kauffman, 2007.
2. D. K. Pradhan, Editor, Fault Tolerant Computer System Design, Prentice Hall, 1996.
3. L. L. Pullum, Software Fault Tolerance Techniques and Implementation, Artech House Computer Security Series, 2001.
4. M. L. Shooman, Reliability of Computer Systems and Networks Fault Tolerance Analysis and Design, Wiley, 2002.

Course Code: CS 5185	FOG AND EDGE COMPUTING	Credits 3-0-0: 3

Pre-Requisites: None

Course Outcomes:

At the end of the course, the student will be able to

CO1	Understand the basic requirements of fog and edge computing.
CO2	Understand the key architectures and applications in fog and edge computing.
CO3	Perform fog and edge computing services.
CO4	Implement software using standard open-source fog and edge computing software for data analytics.

Course Articulation Matrix:

	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

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to Fog Computing, Limitation of Cloud Computing, Differences between Cloud and Fog Computing, Advantages, Business Models, Architecture, Opportunities and Challenges, Challenges in Fog Resources: Taxonomy and Characteristics, Resource Management Challenge, Optimization Challenges, Miscellaneous Challenges, IoT and Fog: Programming Paradigms, Research Challenges and Research Directions, Fog Protocols, Management and Orchestration of Network Slices in 5G, Fog, Edge and Clouds, Data Management and Analysis in Fog Computing, Case Studies. Introduction to Edge Computing, Origins of Edge, Edge Helping Low-End IoT Nodes, Architecture, Edge Helping Higher-Capability Mobile Devices: Mobile Offloading, Edge Helping the Cloud, Edge for Augmented Reality, Data Processing on the Edge, Dispersed Learning with Edge/Fog Computing, Video Analytics on the Edge, Edge Computing Applications.

Text Books/Reference Books/Online Resources:

1. Rajkumar Buyya, Satish Narayana Srirama, Fog and Edge Computing, Wiley Publications, 2019.
2. Wei Change and Jie Wu, Fog/Edge Computing for Security, Privacy and Applications, Springer, 2021.

Course Code: CS 5186	MOBILE SECURITY	Credits 3-0-0: 3

Pre-Requisites: None

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

CO1	Understand vulnerabilities at mobile devices and wireless infrastructure
CO2	Analyse vulnerability mitigation techniques
CO3	Analyse mobile computing software-related threats and vulnerabilities, and evaluate methodologies and best practices for application security
CO4	Design mobile computing access control models
CO5	Design secure communication frameworks for communication between different mobile computing devices.

Course Articulation Matrix:

	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		

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to Mobile Security, Building Blocks – Basic security and cryptographic techniques. Security of GSM Networks, Security of UMTS Networks, LTE Security, WiFi and Bluetooth Security, Wireless Communications Infrastructure Vulnerabilities Mitigation Techniques, Mobile Device Vulnerabilities, Mobile Device Vulnerabilities Mitigation Techniques and Organizational Mobile Device Security Policy Requirements, Mobile computing access control models.
Mobile Malware and App Security, Security Models of different OS, SMS, Mobile Geolocation and Mobile Web Security, Security of Mobile VoIP Communications, Emerging Trends in Mobile Security. Security in next generation mobile networks.

Text Books/Reference Books/Online Resources:

1. Hacking Exposed Mobile: Security Secrets & Solutions, Bergman, N., Stanfield, M., Rouse, J., Scambray, J., et al. (2013). McGraw Hill, 2013.
2. Mobile Application Security, Himanshu Dviwedi, Chris Clark and David Thiel, McGraw-Hill, 2010.
3. Security of Mobile Communications, Noureddine Boudriga, CRC Press, 2010.
4. Mobile Phone Security and Forensics: A Practical Approach, Androulidakis, I., Springer, 2012.

Course Code: CS 5187	REINFORCEMENT LEARNING	Credits 3-0-0: 3

Pre-Requisites: None

Course Outcomes:

At the end of the course, the student will be able to

CO1	Formulate Reinforcement Learning problems
CO2	Apply various Tabular Solution Methods to Markov Reward Process Problems
CO3	Apply various Iterative Solution methods to Markov Decision Process Problems
CO4	Comprehend Function approximation methods

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction: Introduction to Reinforcement Learning (RL) – Difference between RL and Supervised Learning, RL and Unsupervised Learning. Elements of RL, Markov property, Markov chains, Markov reward process (MRP).
Evaluative Feedback - Multi-Arm Bandit Problem: An n-Armed Bandit Problem, Exploration vs Exploitation principles, Action value methods, Incremental Implementation, tracking a non-stationary problem, optimistic initial values, upper-confidence-bound action selection, Gradient Bandits.
Introduction to and proof of Bellman equations for MRPs
Introduction to Markov decision process (MDP), state and action value functions, Bellman expectation equations, optimality of value functions and policies, Bellman optimality equations.
Dynamic Programming (DP): Overview of dynamic programming for MDP, principle of optimality, Policy Evaluation, Policy Improvement, policy iteration, value iteration, asynchronous DP , Generalized Policy Iteration.
Monte Carlo Methods for Prediction and Control: Overview of Monte Carlo methods for model free RL, Monte Carlo Prediction, Monte Carlo estimation of action values, Monto Carlo Control, On policy and off policy learning, Importance sampling.
Temporal Difference Methods: TD Prediction, Optimality of TD(0), TD Control methods - SARSA, Q-Learning and their variants.
Eligibility traces: n-Step TD Prediction, Forward and Backward view of TD(λ), Equivalence of forward and backward view, Sarsa(λ),, Watkins's Q(λ), Off policy eligibility traces using importance of sampling.
Function Approximation Methods: Value prediction with function approximation, gradient descent methods, Linear methods, control with function approximation.

Text Books/Reference Books/Online Resources:

1. Richard S. Sutton and Andrew G. Barto, Reinforcement Learning: An Introduction", 2nd Edition, The MIT Press.
2. Csaba Szepesvari – Algorithms for Reinforcement Learning – Morgan & Claypool, 2010.