COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

PROGRAMME STRUCTURE & SYLLABUS
[2018 ADMISSIONS ONWARDS]

- MTECH COMPUTER AND INFORMATION SCIENCE
- MTECH SOFTWARE ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE
PROGRAMME STRUCTURE AND SYLLABUS (2018 ADMISSIONS)

M TECH COMPUTER AND INFORMATION SCIENCE

Semester - I

Sl. No.	Course code	Course Title	Core /Elective	Credits	Lec	Lab/ Tutorial	Marks
1	CSC 3101	Mathematical Concepts for Computer Science	C	4	4	3	100
2	CSC 3102	Machine Learning Algorithms	C	4	4	3	100
3	CSC 3103	Design and Analysis of Algorithms	C	3	4	1	100
4	CSC 3104	Algorithms Lab	C	1		3	50
4	-	Elective I	E	3	3	1	100
5	-	Elective II	E	3	3	1	100
Total for Semester I				18	18	12	550

Electives

CSC 3105: Virtualized Systems

CSC 3106: Computational Linguistics

CSC 3107: Advanced Optimization Techniques

CSC 3108: Algorithms for Modern Data Models

CSC 3109: Wireless Communications & Networking

CSC 3110: Digital Image and Video Processing

CSC 3111: Mathematics for Machine Learning

Semester - II

Sl. No.	Course code	Course Title	Core /Elective	Credits	Lec	Lab/ Tutorial	Marks
1	CSC 3201	Algorithms for Massive Datasets	C	4	4	3	100
2	CSC 3202	Probabilistic Graphical Models	C	4	4	3	100
3	CSC 3203	Seminar	C	1		4	50
4	-	Elective III	E	3	3	1	100
5	-	Elective IV	E	3	3	1	100
6	-	Elective V	E	3	3	1	100
Total for Semester II				18	17	13	550

Electives

CSC 3204: Bioinformatics

CSC 3205: Programming Massively Parallel Processors

CSC 3206: Computer Vision

CSC 3207: Modelling Cyber Physical Systems

CSC 3208: Number Theory and Cryptography

CSC 3209: Algorithmic Game Theory

CSC 3210: Deep Learning

CSC 3211: Image and Video Coding

CSC 3212: Visual Quality Assessment

CSC 3213: Reinforcement Learning

CSC 3214: Natural Language Processing with Deep Learning

CSC 3215: Deep Learning for Computer Vision

Semester - III							
1	CSC 3301	Project & Viva Voce	C	18	0	15	400
Semester - IV							
1	CSC 3302	Project & Viva Voce	C	18	0	25	500
Total credits for Degree: **72**							

CSC3101: MATHEMATICAL CONCEPTS FOR COMPUTER SCIENCE

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

Course Description:

This course introduces the study of mathematical structures that are fundamentally discrete in nature. It introduces linear algebra, graph theory and probability. The course is intended to cover the main aspects which are useful in studying, describing and modeling of objects and problems in the context of computer algorithms and programming languages.

Course Objectives:

- To understand Linear systems using Linear Algebra
- To get deep understanding of stochastic processes and their applications
- To study graph theory and its applications using matrix formulation

Course Content:

1. Introduction – proofs – propositions – predicates and quantifiers – truth tables – first order logic – satisfiability – pattern of proof – proofs by cases – proof of an implication – proof by contradiction – proving iff – sets – proving set equations – Russell's paradox – well-ordering principle – induction – invariants – strong induction – structural induction – Pigion hole principle – parity – number theory – divisibility – gcd – Euclid's algorithm – primes

2. Graph theory – simple graphs – isomorphism – subgraphs – weighted graphs – matching problems – stable marriage problem – graph coloring – paths and walks – shortest paths – connectivity – Eulerian and Hamiltonian tours – travelling salesman problem – trees – spanning trees – planar graphs – Euler's formula – directed graphs – strong connectivity – relations – binary relations – surjective and injective relations symmetry, transitivity, reflexivity, equivalence of relations – posets and dags – topological sort

3. Sums and asymptotics – arithmetic, geometric and power sums – approximating sums – harmonic sums – products – Stirling's approximation for finding factorial – asymptotic notations – recurrences – towers of Hanoi – solving recurrences – master theorem – linear recurrences – infinite sets – countable and uncountable sets – cantor's continuum hypothesis

4. Finite automata – regular expressions – pushdown automata – context free grammar – pumping lemmas – Turing machines – Church-Turing thesis – decidability – halting problem – reducibility – recursion theorem – time and space measures – complexity classes – NP – reductions

5. Probability – events and probability spaces – conditional probability – tree diagrams for computing probability – sum and product rules of probability – A posteriori probabilities – identities of conditional probability – independence – mutual independence – birthday paradox – random variables – indicator random variables – probability distribution functions – Bernoulli, Uniform, Binomial distributions – Expectation – linearity of expectations – sums of indicator random variables – expectation of products – variance and standard deviation of random variables – Markov's and Chebyshev's theorems – Bounds for the sums of random variables – random walks

REFERENCES:

1. Eric Lehman, F Thomson Leighton, Albert R Meyer, Mathematics for Computer Science, 1e, MIT, 2010

2. Susanna S. Epp, Discrete Mathematics with Applications, 4e, Brooks Cole, 2010
3. Gary Chartrand, Ping Zhang, A First Course in Graph Theory, 1e, Dover Publications, 2012
4. MichaelSipser, Introduction to Theory of Computation, 2e, Cengage, 2012
5. A First Course in Probability, 9e, Pearson, 2013

ONLINE RESOURCES

1. Tom Leighton, and Marten Dijk. *6.042J Mathematics for Computer Science.*Fall 2010. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu.

2. John Tsitsiklis. *6.041SC Probabilistic Systems Analysis and Applied Probability.* Fall 2013. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu.

3. Igor Pak. *18.315 Combinatorial Theory: Introduction to Graph Theory, Extremal and Enumerative Combinatorics.* Spring 2005. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu.

4. Albert Meyer. *6.844 Computability Theory of and with Scheme.* Spring 2003. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu.

5. Shai Simonson , Theory of Computation, http://www.aduni.org/courses/theory/

CSC3102: MACHINE LEARNING ALGORITHMS

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

Course Description:

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

Course Objectives:

- To understand basics to advanced concepts of Machine Learning
- To attain certain amount of statistical and mathematical sophistication to deal with the subject
- To gain confidence in building Machine Learning algorithms and applications
- To understand the multi-disciplinary aspect of the subject

Course Content:

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

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

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

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

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

REFERENCES:

1. Ethem Alpaydin, Introduction to Machine Learning, 3e, MIT Press, 2014
2. Tom M. Mitchell, Machine Learning, McGraw Hill Education; 1e, 2017
3. Stephen Marsland, Machine Learning, An Algorithmic Perspective, 2e, CRC Press, 2015
4. Giuseppe Bonaccorso, Machine Learning Algorithms, 1e, Packt Publishing Limited, 2017
5. Ethem Alpaydin, Machine Learning- The New AI, MIT Press, 1e, 2016

ONLINE RESOURCES

1. Rohit Singh, TommiJaakkola, and Ali Mohammad.*6.867 Machine Learning.* Fall 2006. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu
2. Andrew Ng, https://www.coursera.org/learn/machine-learning

CSC3103: DESIGN AND ANALYSIS OF ALGORITHMS

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

Course Description:

The course covers the foundational algorithms in depth.

Course Objectives:

- To understand the working and complexity of the fundamental algorithms.
- To develop the ability to design algorithms to attack new problems.

Course Content:

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

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

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

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

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

REFERENCES:

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

CSC3104: ALGORITHMS LAB

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

Course Description:

This lab aims to give hands-on practice at implementing the algorithms taught in the associated course.

Course Objectives:

- To develop the ability to efficiently implement the fundamental algorithms.
- To have the ability to analyze the efficiency of implementations pragmatically.

Course Content:

Session 1

- Introduction to the programming language and the version control system.
- How to add, commit and push changes.
- How to randomly generate inputs and test programs.
- Solve a problem using insertion sort and merge sort.

Session 2

- How to profile programs and how to plot the empirical performance curves.
- Solve a problem using Heaps.
- Plot the running time verses the size of the input.

Session 3

- Solve a problem using Binary Search Trees and Tree balancing.

Session 4

- Solve a problem related to graph traversal and shortest paths.

Session 5

- How to generate complete graphs with random weights.
- Solve a problem related to minimum spanning trees.

Session 6

- Solve a problem using the Union-Find data structure.
- Solve a Maximum flow problem.

Session 7

- Solve a problem using Monte Carlo simulation.
- Solve a problem related to Indicator random variables.

Session 8

- Solve a problem using randomized and non-randomized algorithms. Compare the average running times.

Session 9

- Solve problem related to hashing. Use both hashing using chaining and hashing using open addressing and compare them both.

• Solve a problem related to Perfect hashing. Empirically verify the running time.

Session 10

• Solve a problem using dynamic programming.

Session 11

• Solve a problem using greedy approach and compare with the corresponding dynamic programming solution.

Session 12

• Solve one NP-Complete problem and use the solution to solve another NP-Complete problem.

REFERENCES:

1. Thomas H. Cormen et al, Introduction to Algorithms, 3e, MIT Press, 2009
2. Jon Kleinberg, Eva Tardos, Algorithm Design, 1e, Pearson Education, 2013
3. Robert Sedgewick, Kevin Wayne, Algorithms, 4e, AW Professional, 2011
4. Steven S. Skiena, The Algorithm Design Manual, 2e, Springer, 2011

CSC3105: VIRTUALIZED SYSTEMS

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

Course Description:

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

Course Objectives:

- The course introduces the concepts and principles of virtualization, the mechanisms and techniques of building virtualized systems, as well as the various virtualization-enabled computing paradigms.

Course Content:

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

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

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

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

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

REFERNCES:

1. Jim Smith, Ravi Nair , Virtual Machines: Versatile Platforms for Systems and Processes, 1e, Morgan Kaufmann, 2005
2. Sean Campbell, Applied Virtualization Technology -Usage models for IT professionals and Software Developers, 1e, Intel Press, 2006
3. Matthew Portnoy , Virtualization Essentials, 1e, JW, 2012
4. George Trujillo, Charles Kim, Steve Johnes, Rommel Gracia, Justin Murray , Virtualizing Hadoop, VM Press, 2015

CSC3106: COMPUTATIONAL LINGUISTICS

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

Course Description:

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

Course Objectives:

- To familiarise the fundamentals of speech and written language processing
- To study the applications of these techniques in real world problems like spell-checking, Parts-of Speech Tagging, Corpus development, Wordnet, speech recognition, pronunciation modelling, dialogue agents, document retrieval etc
- To gather information about widely used language processing resources

Course Content:

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

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

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

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

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

REFERNCES:

1. James Pustejovsky, Amber Stubbs, Natural language annotation for machine learning, 1e, O'Reilly, 2013

2. Alexander Clark and Chris Fox, The hand book of Computational linguistics and natural language processing, 1e, Willey-Blackwell, 2012

3. Grant S Ingersoll, Thomas Morton, Andrew L Farris, Taming Text: How to Find, Organize, and Manipulate It, 1e, Manning Publications 2013

4. Daniel Jurafsky and James Martin , Speech and Language Processing, 2e, Pearson, 2013

5. Christopher D. Manning and HinRichSchutze, Foundations of statistical natural language processing, 1e, MIT press, 1999

CSC3107: ADVANCED OPTIMIZATION TECHNIQUES

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

Course Description:

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

Course Objectives:

- To study concepts of Population based Optimization techniques
- To understand the mathematical foundations for various advanced optimization techniques
- To apply the algorithms to various inter disciplinary applications

Course Content:

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

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

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

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

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

REFERENCES:

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

CSC3108: ALGORITHMS FOR MODERN DATA MODELS

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

Course Description:

This course describes the randomization and probabilistic techniques for modern computer science, with applications ranging from combinatorial optimization and Machine learning to communication networks. The course covers the core material to advanced concepts. Also the emphasis is on methods useful in practice.

Course Objectives:

- To know problem solving techniques for modern problems
- To understand probabilistic techniques and analysis of algorithms
- To be able to design algorithms for new problems with volume of data

Course Content:

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

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

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

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

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

REFERENCES:

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

CSC3109: WIRELESS COMMUNICATIONS & NETWORKING

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

Course Description:

This course focuses on imparting knowledge about the practical aspects of wireless network systems with the required basic principles behind them, along with some practical assignments. The course examines the conceptual framework for specifying a wireless network and the related protocols.

Course Objectives:

- Comprehend and demonstrate command in the principles of wireless networking.
- Describe the networking technologies including Cellular networks, WLANs and WWANs.
- Understand the functions of TCP/IP and the organization of the Internet.
- Design and evaluate a wireless network in terms of cost, performance, privacy and security.
- Plan and design a small and practical network for home or small business applications under a specified set of constraints
- To understand new trends and emerging technologies

Course Content:

1. Overview of wireless systems – Tele traffic engineering – Radio propagation – Path loss models – Digital communication over radio channels – Modelling of a Wireless Channel - Capacity of wireless channels – AWGN channel -Fading channels

2. Cellular concepts – Multiple access and interference management- Narrowband and Wideband systems- GSM, CDMA and OFDM - Channel reuse analysis- spread spectrum and CDMA systems – Random access and Wireless LANs – Data and voice sessions over 802.11 – Association in WLANs

3. Wide-Area Wireless Networks – GSM evolution for data – UMTS architecture – QoS in UMTS – HSDPA – FOMA - CDMA evolution

4. Design of a wireless network – radio design for a cellular network – Link budget for GSM and CDMA

5. Beyond 3G – HSPA+, WiMAX and LTE – Cognitive radio networks

REFERENCES:

1. Cory Beard and William Stallings, Wireless Communication Networks and Systems, 1e, Pearson, 2015
2. Vijay K Garg, Wireless Communications & Networking, 1e, Morgan Kaufmann, 2008
3. Anurag Kumar, D. Manjunath, Joy Kuri , Wireless Networks:, 1e, Morgan Kaufman, 2008
4. Christopher Cox , An Introduction to LTE: LTE, LTE-Advanced, SAE and 4G Mobile Communications, 2e, Wiley, 2012
5 Web Resources: http://standards.ieee.org

CSC 3110: DIGITAL IMAGE AND VIDEO PROCESSING

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

Course Description:

The aim of this course is to inculcate a comprehensive knowledge about various Digital Image and Video Processing techniques.

Course Objectives:

- Give an in-depth knowledge about the basic theory and algorithms related to Digital Image and Video Processing.
- Provide awareness about the current technologies and issues specific to Digital Image and Video Processing.
- Provide hands-on experience in using computers to process digital images and Videos.
- Expose students to Python and OpenCV library to do image and video processing tasks.

Course Content:

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

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

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

4. Point and line detection - Hough Transform. Image Segmentation: Fundamentals – Thresholding – Otsu's optimum global thresholding - Region-based segmentation: Region growing - Region Splitting and Merging - Segmentation using Morphological Watersheds.

5. Color video processing: Video display - Composite versus component video - Progressive and interlaced scan. Motion estimation: Optical flow - pixel based motion estimation - block matching algorithm - deformable block matching algorithm - Global and region based motion estimation - multi-resolution motion estimation - Feature based

motion estimation. Stereo and multi-view sequence processing: Depth perception - Stereo imaging principle - Disparity estimation.

REFERENCES:

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

CSC3201: ALGORITHMS FOR MASSIVE DATASETS

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

Course Description:

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

Course Objectives:

- To understand emphasis on the algorithms to be applied on large amounts of data
- To develop hands-on experience on the distributed file systems and MapReduce as a tool for creating parallel algorithms
- To explore streaming data and some of the techniques and algorithms specifically extended for mining on stream data

Course Content:

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

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

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

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

5. Computational model for data mining – storage, cost model, and main memory bottleneck. Hash based algorithm for mining association rule – improvements to a-priori, park-chen-yu algorithm, multistage algorithm, approximate algorithm, limited-pass algorithms – simple randomized algorithm, Savasere, Omiecinski, and Navathe algorithm, Toivonen algorithm

REFERENCES:

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

CSC3202: PROBABILISTIC GRAPHICAL MODELS

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

Course Description:

Probabilistic graphical models (PGM) is one of the most advanced techniques in machine learning to represent data and models in the real world with probabilities. PGM present a general framework for constructing and using probabilistic models of complex systems that would enable a computer to use available information for making decisions. This course is for anyone who has to deal with lots of data and draw conclusions from it, especially when the data is noisy or uncertain. Data scientists, machine learning enthusiasts, engineers, and those who curious about the latest advances in machine learning will find PGM interesting

Course Objectives:

- Understand the concepts of PGM and which type of PGM to use for which problem
- To understand techniques for representation, inference and learning from graph based models
- To apply Bayesian networks and Markov networks to many real world problems

Course Content:

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

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

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

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

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

REFERENCES:

1. Daphne Koller, Nir Friedman, Probabilistic Graphical Models- Principles and Techniques, 1e, MIT Press, 2009
2. Christian Borgelt, Rudolf Kruse and Matthias Steinbrecher, Graphical Models- Methods for data analysis and Mining, 2e, Wiley, 2009
3. David Bellot, Learning Probabilistic Graphical Models in R, Packt Publishing, 1e, 2016
4. Luis Enrique Sucar, Probabilistic Graphical Models, 1e, Springer Nature, 2015

CSC3203: SEMINAR

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

Course Description:

The student has to prepare and deliver a presentation on a research topic suggested by faculty member before the peer students and staff. They also have to prepare a comprehensive report of the seminar presented

Course Objective:

- Review and increase their understanding of the specific topics tested.
- Inculcating presentation and leadership skills among students
- Offering the presenter student an opportunity of interaction with peer students and staff

CSC3204: BIOINFORMATICS

Core/Elective: Elective**Semester:**2**Credits:**3

Course Description:

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

Course Objectives:

- To familiarize computational problems in biology
- To understand models of DNA and DNA mapping
- To study structure prediction

Course Content:

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

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

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

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

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

REFERENCES:

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

CSC3205: PROGRAMMING MASSIVELY PARALLEL PROCESSORS

Core/Elective: Elective **Semester:** 2 **Credits:** 3

Course Description:

It used to be the case that parallel computing was confined to giant supercomputers. But nowadays it is literally everywhere - even in the small mobile handsets that most of us carry around. This course introduces parallel computing with a strong emphasis on programming.

Course Objectives:

- To understand the basics of parallel computing
- To develop programming skills required for parallel computing.
- To learn about strategies for how algorithms that were originally developed for single-processor systems can be converted to run efficiently on parallel computers
- To know about current practical implementations of parallel architectures

Course Content:

1. Introduction - parallel computing - more speed or parallelism - languages and models - sequential vs parallel - concurrent, parallel, distributed - parallel hardware architecture - modifications to the von Neumann Model.

2. Evolution of GPU - GPGPU - introduction to data parallelism - CUDA program structure - vector addition kernel - device global memory and data transfer

3. CUDA thread organization - mapping threads to multi-dimensional data - assigning resources to blocks - synchronization and transparent scalability - thread scheduling and latency tolerance

4. Memory access efficiency - CUDA device memory types - performance considerations - global memory bandwidth - instruction mix and thread granularity -floating point considerations

5 Parallel programming patterns - convolution - prefix sum - sparse matrix and vector multiplication - application case studies - strategies for solving problems using parallel programming

REFERENCES:

1. David B. Kirk, Wen-mei W Hwu, Programming Massively Parallel Processors, 2e, Morgan Kaufmann, 2012
2. Peter Pacheco, Introduction to Parallel Programming, 1e, Morgan Kaufmann, 2011
3. Shane Cook, CUDA Programming: A Developer's Guide to Parallel Computing with GPUs, 1e, Morgan Kaufmann, 2012
4. Jason Sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, 1e, AW Professional, 2010

CSC3206: COMPUTER VISION

Core/Elective: Elective **Semester:** 2 **Credits:** 3

Course Description:

This course introduces concepts and applications in computer vision. Starting with image formation the course covers image processing methods such as filtering and edge detection, segmentation and classification. It includes vision tasks like; object detection, recognition and human motion detection. The content of the course also includes practical exercises to help the students formulating and solving computer vision problems.

Course Objectives:

- To understand processing of digital images
- To familiarise different mathematical structures
- To study detailed models of image formation
- To study image feature detection, matching, segmentation and recognition
- To understand classification and recognition of objects.
- To familiarize state-of-the-art problems in computer vision

Course Content:

1. Low-level vision: Images and imaging operations – Convolutions and point spread functions – Image filtering and morphology- filters – mathematical morphology – thresholding – edge detection.

2. Corner, interest point, and invariant feature detection – Local invariant feature detectors and descriptors: Harris, Hessian, SIFT, SURF and HOG – Texture analysis

3. Intermediate-level vision: Binary shape analysis – Boundary pattern analysis – Line, Circle, and Ellipse detection – generalized Hough transform – Object segmentation

4. Machine learning and deeplearning networks: Classifications concepts – Probabilistic methods – Deep learning networks: AlexNet, VGGNet, SegNet, and Recurrent Neural Networks

5. Motion: Optical flow – Kalman Filter – Surveillance: Foreground-background separation- Particle filters- combining views from multiple cameras – In-vehicle vision systems

REFERENCES:

1. E.R. Davies, Computer Vision: Principles, Algorithms, Applications, Learning, 5e, AP, 2017
2. Ponce Jean & Forsyth David , Computer Vision: A Modern Approach, 2e, PHI, 2014
3. Richard Szeliski, Computer vision: Algorithms and Applications, 1e, Springer, 2010
4. J. R. Parker, Algorithms for Image Processing and Computer Vision, 2e, Wiley, 2010
5. Maria Petrou and Costas Petrou, Image Processing: The Fundamentals, 2e, Wiley, 2010

CSC3207: MODELING CYBER PHYSICAL SYSTEMS

Core/Elective: Elective **Semester:** 2 **Credits:** 3

Course Description:

The course examines wireless cellular, ad hoc and sensor networks, covering topics such as wireless communication fundamentals, medium access control, network and transport protocols, unicast and multicast routing algorithms, mobility and its impact on routing protocols, application performance, quality of service guarantees, and security. Energy efficiency and the role of hardware and software architectures may also be presented for sensor networks.

Course Objectives:

- To know problem solving techniques
- To understand techniques for the design and analysis of efficient algorithms
- To be able to design algorithms for new problems with volume of data

Course Content:

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

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

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

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

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

REFERENCES:

1. Edward Ashford Lee, Sanjit Arunkumar Seshia, Introduction to Embedded Systems - A Cyber-Physical Systems Approach, 2e, MIT Press, 2017
2. Rajeev Alur , Principles of Cyber-Physical Systems, 1e, MIT Press, 2015
3. Raj Rajkumar, Dionisio de Niz, Mark Klein, Cyber-Physical Systems, 1e, AW Professional, 2017
4. Peter Marwedel, Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems, and the Internet of Things, 3e, Springer, 2017

CSC3208: NUMBER THEORY AND CRYPTOGRAPHY

Core/Elective: Elective **Semester:** 2 **Credits:** 3

Course Description:

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

Course Objectives:

- To understand the number theoretic foundations of modern cryptography
- To implement and analyze cryptographic and number theoretic algorithms
- To understand public key cryptosystems
- To understand modern cryptographic techniques

Course Content:

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

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

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

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

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

REFERENCES:

1. James S. Kraft and Lawrence C. Washington, An Introduction to Number Theory with Cryptography, 1e, CRC Press, 2013
2. Jill Pipher, Jeffrey Hoffstein, Joseph H. Silverman, An Introduction to Mathematical Cryptography, 2e, Springer, 2014
3. Christof Paar and Jan Pelzl, Understanding Cryptography, 1e, Springer, 2010
3. G.H.Hardy and Edward M Wright, An Introduction to theory of numbers, 1e, Oxford, 2008
5. : Song Y.Yan, Computational Number Theory & Modern Cryptography, 1e, Wiley, 2013

CSC3209: ALGORITHMIC GAME THEORY

Core/Elective: Elective **Semester:** 2 **Credits:** 3

Course Description:

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

Course Objectives:

- To get a practical understanding of game theory
- To be able to solve computer science problems using the concepts of game theory

Course Content:

1. Introduction to game theory – strategies, costs, payoffs – solution concepts – finding equilibria – games with sequential moves – games with simultaneous moves – discrete strategies, continuous strategies – mixed strategies – games with incomplete information – expected payoffs – Prisoner's dilemma and repeated games – Nash equilibrium – Computational complexity of Nash equilibrium

2. Games on networks – congestion games – selfish routing – Nash and wardrop equilibria for networks – price of anarchy – pricing network edges – network design with selfish agents – economic aspects of internet routing

3. Epistemic game theory – Modeling knowledge – rationality and belief – common belief in rationality – game strategies and perfect recall – cryptography and game theory – modeling cryptographic algorithms as games – multi-party computations – MPC and games

4. Mechanism design – general principles – social choice – incentives – algorithms mechanism design – distributed aspects – cost-sharing mechanisms – mechanism design without money – house allocation problem – stable matchings

5. Voting – evaluation of voting systems – strategic manipulation of votes – auctions – types of auctions – winner's curse – bidding strategies – fairness in auctions

REFERENCES:

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

CSC3210: DEEP LEARNING

Core/Elective: Elective **Semester:** 2 **Credits:** 3

Course Description:

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

Course Objectives:

- To develop a clear understanding of the motivation for deep learning
- To get a practical understanding of machine learning methods based on learning data
- To design intelligent systems that learn from complex and/or large-scale datasets
- To apply deep learning to practical problems

Course Content:

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

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

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

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

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

REFERENCES:

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

CSC3211: IMAGE AND VIDEO CODING

Core/Elective: Elective **Semester:** 2 **Credits:** 3

Course Description:

This course aims to give a rigorous introduction into the fundamental concepts of data compression with strong emphasis on the mathematical techniques and its applications to image and video coding.

Course Objectives:

- To understand how digital data can be compressed using either lossless or lossy techniques.
- To provide a strong mathematical background in the field of coding theory
- To expose the students to the standard compression techniques used in various coding standards.
- To expose the students to the latest image and video coding standards.

Course Content:

1. Introduction: Compression Techniques - Modeling and Coding. Mathematical Preliminaries for Lossless compression: Information Theory – Models - Coding: Uniquely decodable codes - Prefix codes - Kraft-McMillan Inequality. Huffman Coding: Minimum Variance Huffman Codes - Length of Huffman Codes - Adaptive Huffman Coding - Golomb codes - Rice codes - Tunstall codes. Arithmetic Coding: Integer Arithmetic Coding.
2. Dictionary Techniques: Static Dictionary - Digram coding - Adaptive Dictionary - LZ77 - LZ78 - LZW. Context-based Compression: Prediction with partial match - Burrows-Wheeler Transform – CALIC - Run-Length Coding – JBIG - JBIG2.
3. Mathematical Preliminaries for Lossy Coding: Distortion Criteria - Rate Distortion Theory. Scalar Quantization: Quantization problem - Uniform Quantizer - Lloyd-Max Quantizer - Adaptive Quantization - Non-uniform Quantization - Entropy-Coded Quantization. Vector Quantization: LBG Algorithm - Tree Structured and Structured Vector Quantizers. Differential Coding: Basic algorithm – DPCM. Transform Coding.

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

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

REFERENCES:

1. Khalid Sayood, "Introduction to Data Compression", Morgan Kaufmann Publishers, 4th Ed., 2012.
2. David Salomon, "Data Compression – The Complete Reference", Springer, 4th Ed., 2006.

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

CSC3212: VISUAL QUALITY ASSESSMENT

Core/Elective: Elective **Semester:** 2 **Credits:** 3

Course Description:

The aim of this course is to provide the foundation knowledge in various quantitative metrics used for assessing the quality of image/video data.

Course Objectives:

- To learn and implement basic and advanced image quality assessment metrics.
- To learn and implement basic and advanced video quality assessment metrics.
- To get knowledge about the significance of reference and no-reference based image/video quality assessment metrics.
- To get an insight into the latest research development in the domain of image/video quality assessment metrics.

Course Content:

1. Fundamentals of Human Vision and Vision Modeling: Color Vision – Luminance - Perception of Light Intensity - Spatial Vision - Contrast Sensitivity - Temporal Vision and Motion - Visual Modeling.

2. Video Quality Testing: Subjective Assessment Methodologies - Selection of Test Materials - Selection of Participants - Experimental Design - International Test Methods - Objective Assessment Methods. Perceptual Video Quality Metrics: Quality Factors - Metrics Classification - Pixel-Based Metrics -Psychophysical Approach - Engineering Approach - Metric Comparisons.

3. Philosophy of Picture Quality Scale: PQS and Evaluation of Displayed Image - Construction of a Picture Quality Scale - Visual Assessment Tests - Results of Experiments - Key Distortion Factors - Applications of PQS. Signal-to-noise ratio (SNR) – Peak signal-to-noise Ratio (PSNR) - Structural Similarity Based Image Quality Assessment: Structural Similarity (SSIM) Index – Universal Image Quality Index (UIQI) – Feature Similarity Index (FSIM).

4. Vision Model Based Digital Video Impairment Metrics: Vision Modeling for Impairment Measurement - Perceptual Blocking Distortion Metric - Perceptual Ringing Distortion Measure. Computational Models for Just-Noticeable Difference: Single-Stimulus JNDT Tests - JND with DCT Sub-bands - JND with Pixels - JND Model Evaluation.

5. No-Reference Quality Metric for Degraded and Enhanced Video: State-of-the-art for No-Reference Metrics - Quality Metric Components and Design - No-Reference Overall Quality Metric - Performance of the Quality Metric - Video Quality Experts Group.

REFERENCES:

1. H. R. Wu and K. R. Rao. “Digital Video Image Quality and Perceptual Coding”, CRC Press, 2006. ISBN-13: 978-0-8247-2777-2.
2. Stefan Winkler. "Digital Video Quality: Vision Models and Metrics", Wiley Publishers, 1st Ed., 2005. ISBN-13: 978-0470024041.
3. Oleg S. Pianykh. "Digital Image Quality in Medicine - Understanding Medical Informatics", Springer, 2014 Ed., 2013. ISBN-13: 978-3319017594.
4. Shahriar Akramullah. "Digital Video Concepts, Methods, and Metrics: Quality, Compression, Performance, and Power Trade-off Analysis", Apress; 2014 Ed., 2014. ISBN-13: 978-1430267126.
5. Chenwei Deng, Lin Ma, Weisi Lin, King Ngi Ngan. "Visual Signal Quality Assessment: Quality of Experience (QoE)", Springer; 2015 Ed., 2014. ISBN-13: 978-3319103679.
6. Madhuri A. Joshi, Mehul S. Raval, Yogesh H. Dandawate, Kalyani R. Joshi, Shilpa P. Metkar. "Image and Video Compression: Fundamentals, Techniques, and Applications", CRC Press, 1st Ed., 2014. ISBN-13: 978-1482228229.

CSC3213: MATHEMATICS FOR MACHINE LEARNING

Core/Elective: Elective **Semester:** 2 **Credits:** 3

Course Description:

This course covers the mathematics necessary to understand machine learning algorithms.

Course Objective:

- To have a working knowledge of mathematical ideas required for understanding topics of machine learning.

Course Contents:

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

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

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

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

5. Applications: linear regression – parameter estimation – Bayesian Linear Regression – PCA – Maximum Variance Projections – Low-Rank Approximations – Gaussian mixture models – Parameter learning via maximum likelihood – EM Algorithm – Support Vector Machines – Separating Hyperplanes – Primal and Dual forms – The Kernel Trick

References:

1. Gilbert Strang, Linear Algebra and Learning from Data, Wellesley-Cambridge Press, 2019
2. Marc Peter Deisenroth et al., Mathematics for Machine Learning (draft), Ebook: https://mml-book.com

3. Mehryar Mohri et al., Foundations of Machine Learning, 2nd Edition, The MIT Press, 2018
4. Gilbert Strang, Introduction to Linear Algebra, 5th Edition, Wellesley-Cambridge Press, 2016
5. James Stewart, Multivariable Calculus, 7th Edition, Cengage Learning, 2011
6. Dimitri P. Bertsekas, John N. Tsitsiklis, Introduction to Probability, 2nd Edition, Athena Scientific, 2008.
7. Morris H. DeGroot, Mark J. Schervish, Probability and Statistics, 4th Edition, Pearson, 2011

CSC3214: NATURAL LANGUAGE PROCESSING WITH DEEP LEARNING

Core/Elective: Elective **Semester:** 2 **Credits:** 3

Course Description:

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

Course Objectives:

- To learn the foundations of speech and written language processing
- To look at various NLP tasks through the lens of deep learning
- To gain practical insights in tools and techniques to implement an NLP project effectively
- To outline and analyze various research frontiers of NLP in the deep learning era

Course Content:

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

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

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

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

5. Natural language generation – Coreference resolution – Constituency parsing

References:

1. Dan Jurafsky and James H. Martin. Speech and Language Processing, Prentice Hall, 3rd ed, 2019
2. Jacob Eisenstein. Natural Language Processing, The MIT Press, 2019

3. Yoav Goldberg, Graeme Hirst.. Neural Network Models for Natural Language Processing, Morgan and Claypool Life Sciences, 2017
4. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning, The MIT Press, 2017
5. Charu C Aggarwal. Neural Networks and Deep Learning, Springer, 2018
6. Eugene Charniak. Introduction to Deep Learning, The MIT Press, 2019

Online Courses

1. Natural Language Processing in Deep Learning by Stanford (Coursera)
2. Deep Learning Specialization by deeplearning.ai
3. Applied AI with Deep Learning by IBM (Coursera)
4. Introduction to pyTorch and Machine Learning (Udemy)
5. Practical Deep Learning with pyTorch (Udemy)

CSC3215: DEEP LEARNING FOR COMPUTER VISION

Core/Elective: Elective **Semester:** 2 **Credits:** 3

Course Description:

Deep learning has fundamentally changed the landscapes of a number of areas in artificial intelligence, including vision, NLP, and game playing. Intersection of deep learning and CV have emerged with interesting results serving a benchmark for the advances in one of the most important tasks in artificial intelligence. This course introduces the state of the art of deep learning research and its successful applications to major CV tasks, including image classification, object detection, image captioning, video classification, and image generation.

Course Objectives:

- To learn the foundations of computer vision.
- To look at various CV tasks through the lens of deep learning.
- To gain practical insights in tools and techniques to implement an CV projects effectively.
- To outline and analyze various research frontiers of CV in the deep learning era.

Course Content:

1. Neural Networks – Stochastic Gradient Descent – Back Propagation – Logistic Regression –Softmax. Implementation: Image Classification problem.

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

3. Transfer Learning – Pre-trained Models – Recurrent Neural Networks (RNN) – LSTM – GRU –Autoencoders and its variants.Implementation: Hyperspectral Image Classification using RNN.

4. Generative Adversarial Network (GAN) – Attention Mechanism – YOLO. Implementation: Object Detection in Image using YOLO.

5. Video Classification: Approaches – Parallel CNN – Streaming CNN for action recognition – 3D convolution for temporal learning – Segmenting and captioning videos
Implementation: Video Classification/Video Summarization/Anomaly Detection using CNN

References:

1. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning, The MIT Press, 2017
2. Charu C Aggarwal. Neural Networks and Deep Learning, Springer, 2018
3. Eugene Charniak. Introduction to Deep Learning, The MIT Press, 2019
4. Linda G. Shapiro, George C. Stockman, “Computer Vision”, Prentice Hall, 1st Ed., 2001.

5. Richard Szeliski, “Computer Vision: Algorithms and Applications”, Springer, 1st Ed., 2010.
6. David A. Forsyth, Jean Ponce, “Computer Vision: A Modern Approach”, 2nd Ed., 2011.
7. Simon J. D. Prince, “Computer Vision: Models, Learning, and Inference”, Cambridge University Press, 1st Ed., 2012.
8. Ramesh Jain, Rangachar Kasturi, Brian G. Schunck, “Machine Vision”, McGraw-Hill, 1st Ed., 1995.

Online Courses

1. Deep Learning Specialization by deeplearning.ai
2. Applied AI with Deep Learning by IBM (Coursera)
3. Introduction to pyTorch and Machine Learning (Udemy)
4. Practical Deep Learning with pyTorch (Udemy)

DEPARTMENT OF COMPUTER SCIENCE PROGRAMME STRUCTURE AND SYLLABUS (2018 ADMISSIONS)							
M TECH SOFTWARE ENGINEERING							
Semester - I							
Sl. No.	Course code	Course Title	Core/ Elective	Credits	Lec.	Lab/ Tutorial	Marks
1	CSS3101	Mathematical Concepts for Computer Science	C	4	4	3	100
2	CSS3102	Advances in Databases	C	4	4	3	100
3	CSS3103	Design and Analysis of Algorithms	C	3	4	1	100
4	CSS3104	Algorithms Lab	C	1		3	50
5	-	Elective I	E	3	3	1	100
6	-	Elective II	E	3	3	1	100
Total for Semester I				18	18	12	550
Electives							
CSS 3105: Artificial Intelligence							
CSS 3106: Human Computer Interaction							
CSS 3107: Information Retrieval and Web search							
CSS 3108: Functional Programming							
CSS 3109: Software Quality Management							
CSS 3110: Patterns in Software Engineering							
Semester - II							
Sl. No.	Course code	Course Title	Core/ Elective	Credits	Lec.	Lab/ Tutorial	Marks
1	CSS 3201	Software Architecture	C	4	4	3	100
2	CSS 3202	Agile Software Engineering	C	4	4	3	100
3	CSS 3203	Seminar	C	1	0	4	50
4	-	Elective III	E	3	3	1	100
5	-	Elective IV	E	3	3	1	100
6	-	Elective V	E	3	3	1	100
Total for Semester II				18	17	13	550
Electives							
CSS 3204: Design of Real Time/Embedded Software							
CSS 3205: Software Agent Systems							
CSS 3206: Network Forensics							
CSS 3207: Enterprise Application Integration & Business Process Management							
CSS 3208: Advanced Data Mining							
CSS 3209: Fuzzy Set Theory: Foundations & Applications							
CSS 3210: Complex Networks: Theory & Applications							
CSS 3211: Data Science & Big Data Analytics							
Semester - III							
1	CSS 3301	Project & Viva Voce	C	18	0	15	400
Semester - IV							
1	CSS 3302	Project & Viva Voce	C	18	0	25	500

Total credits for Degree: **72**

CSS3101: MATHEMATICAL CONCEPTS FOR COMPUTER SCIENCE

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

Course Description:

This course introduces the study of mathematical structures that are fundamentally discrete in nature. It introduces linear algebra, graph theory and probability. The course is intended to cover the main aspects which are useful in studying, describing and modelling of objects and problems in the context of computer algorithms and programming languages.

Course Objectives:

- To understand Linear systems using Linear Algebra
- To get deep understanding of stochastic processes and their applications
- To study graph theory and its applications using matrix formulation

Course Content:

1. Linear Systems: Vector Spaces, Linear Independence and Rank, Basis, Quadratic forms and Semi Definite matrices, Eigen values and Eigen Vectors, LU Decomposition, Orthogonality, Least squares problem, QR decomposition, SVD, Basic Tensor Concepts

2. Probability axioms, Bayes theorem, Random variables and distributions, Expectation and Variance, Covariance and correlation, Moment generating functions, Inequalities: Markov, Chebyshev, Chernoff bound, Laws of large numbers, Central limit theorem

3. Multivariate Distribution, Point estimation: EM algorithm, Mean-square Error, Sufficiency-Completeness, Testing hypotheses, Stochastic Processes and Markov Chains: Markov process, Kolmogorov-Chapman equations, Parameter Estimation

4. Matrix representation of graphs, Hypergraphs, Bipartite graphs, Components, Independent paths, connectivity and cut sets, Graph Laplacian - Random walks - Measures and metrics on graphs

5. Random Graphs : Models, Ramsey theory, Random graphs with general degree distributions Connectivity and Matchings, Diameter of Random Graphs

REFERENCES:

1. Eric Lehman, F Thomson Leighton, Albert R Meyer, Mathematics for Computer Science, 1e, MIT, 2010
2. Susanna S. Epp, Discrete Mathematics with Applications, 4e, Brooks Cole, 2010
3. Gary Chartrand, Ping Zhang, A First Course in Graph Theory, 1e, Dover Publications, 2012
4. MichaelSipser, Introduction to Theory of Computation, 2e, Cengage, 2012
5. A First Course in Probability, 9e, Pearson, 2013

ONLINE RESOURCES

1. Tom Leighton, and Marten Dijk. *6.042J Mathematics for Computer Science.*Fall 2010. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu.

2. John Tsitsiklis. *6.041SC Probabilistic Systems Analysis and Applied Probability.* Fall 2013. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu.

3. Igor Pak. *18.315 Combinatorial Theory: Introduction to Graph Theory, Extremal and Enumerative Combinatorics.* Spring 2005. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu.

4. Albert Meyer. *6.844 Computability Theory of and with Scheme.* Spring 2003. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu.

5. Shai Simonson , Theory of Computation, http://www.aduni.org/courses/theory/

CSS3102: ADVANCES IN DATABASES

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

Course Description:

This is a second course in database systems which cover advanced aspects of database systems touching upon the theoretical advancements to handle the new areas and challenges related to the management of data. The course introduces the students to the frontiers of the classical database systems and takes them to the multidimensional data and the associated processing techniques. Later, a large multitude of specialty databases are introduced. This course consolidates the theory and practices pertaining to big data storages and cloud databases.

Course Objectives:

- The course introduces the emerging theories and practices in the field of database systems management.
- The orientation of the contents are towards the more challenging data management areas like big data, cloud computing and data science.

Course Content:

1. Physical Database Design: The Physical Database Design Process - Data Volume and Usage Analysis - Controlling Data Integrity - Missing Data – Denormalization –Partitioning - File Organizations – Heap-Sequential-Indexed-Hashed – Non-unique indexing.

2. Online Analytical Processing: Recent Enhancements and Extensions to SQL - Analytical and OLAP Functions–Multidimensional Analysis - New Data Types- New Temporal Features in SQL- Other Enhancements. Need for Data Warehousing – Architectures- Data Mart and Data Warehousing Environment - Real-Time Data Warehouse Architecture - Enterprise Data Model-Status/Event/Transient/Periodic Data - Derived Data - Star Schema and variations - Fact Tables - Dimension Tables - Normalization - Surrogate Key - Hierarchies - Unstructured Data.

3. Object-Based Databases: Complex Data Types - Structured Types and Inheritance in SQL - Table Inheritance - Array and Multiset Types in SQL - Object-Identity and Reference Types in SQL - Implementing O-R Features - Persistent Programming Languages - Object-Relational Mapping. Object-Oriented Databases: Motivation – Concepts and Features – Object Modelling – Indexing – Design Considerations- Object-Oriented versus Object-Relational. XML Databases: Motivation - Structure of XML Data - XML Document Schema - Querying and Transformation – XPath – XQuery – XSLT - Application Program Interfaces to XML - Storage of XML Data - XML Applications.

4. Spatial and Temporal Data: Motivation - Time in Databases - Spatial and Geographic Data - Multimedia Databases -Mobility and Personal Databases - Active Databases, Time series Databases. Advanced Transaction Processing : Transaction-Processing Monitors - Transactional Workflows - E-Commerce - Main-Memory Databases - Real-Time Transaction Systems - Long-Duration Transactions.

5. Classification of NoSQL Database Management Systems, Key-Value Stores- Document Stores - Wide-Column Stores - Graph-Oriented Databases–Redis, MongoDB, Cassandra, Neo4j – Hadoop data storage – Pig, Hive, HBase – Introduction to Integrated Data Architecture.
Cloud Databases: Database as a service (DBaS), Amazon SimpleDB, DynamoDB – EnterpriseDB - Google Cloud SQL, Google BigQuery – Microsoft Azure SQL.

REFERENCES:

1. A. Hoffer Jeffrey, V. Ramesh, Topi Heikki, Modern database management, 12e, Pearson, 2015.

2. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts, 6e, McGraw-Hill, 2013
3. SherifSakr, Big Data 2.0 Processing Systems: A Survey, 1e, Springer: Briefs in Computer Science, 2016
4. Lee Chao, Cloud Database Development and Management, 1e, CRC Press, 2013
5. Rini Chakrabarti and Shilbhadra Dasgupta, Advanced Database Management System, 1e, Dreamtech Press, 2011
6. Michael Stonebreaker, Paul Brown, and Dorothy, Object-Relational DBMSs, 2e , Morgan Kaufman, 1998
7. PhilipeRigaux, et. al, Spatial Databases (with applications to GIS), 1e, Morgan Kaufmann, 2001
8. Jennifer Widom, Stefano Ceri, Active Database Systems: Triggers and Rules for Advanced Database Processing,1e, Morgan Kaufmann, 1995

CSS3103: DESIGN AND ANALYSIS OF ALGORITHMS

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

Course Description:

The course covers the foundational algorithms in depth.

Course Objectives:

- To understand the working and complexity of the fundamental algorithms.
- To develop the ability to design algorithms to attack new problems.

Course Content:

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

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

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

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

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

REFERENCES:

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

CSS3104: ALGORITHMS LAB

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

Course Description:

This lab aims to give hands-on practice at implementing the algorithms taught in the associated course.

Course Objectives:

- To develop the ability to efficiently implement the fundamental algorithms.
- To have the ability to analyze the efficiency of implementations pragmatically.

Course Content:

Session 1

- Introduction to the programming language and the version control system.
- How to add, commit and push changes.
- How to randomly generate inputs and test programs.
- Solve a problem using insertion sort and merge sort.

Session 2

- How to profile programs and how to plot the empirical performance curves.
- Solve a problem using Heaps.
- Plot the running time verses the size of the input.

Session 3

- Solve a problem using Binary Search Trees and Tree balancing.

Session 4

- Solve a problem related to graph traversal and shortest paths.

Session 5

- How to generate complete graphs with random weights.
- Solve a problem related to minimum spanning trees.

Session 6

- Solve a problem using the Union-Find data structure.
- Solve a Maximum flow problem.

Session 7

- Solve a problem using Monte Carlo simulation.
- Solve a problem related to Indicator random variables.

Session 8

- Solve a problem using randomized and non-randomized algorithms. Compare the average running times.

Session 9

- Solve problem related to hashing. Use both hashing using chaining and hashing using open addressing and compare them both.

- Solve a problem related to Perfect hashing. Empirically verify the running time.

Session 10

- Solve a problem using dynamic programming.

Session 11

- Solve a problem using greedy approach and compare with the corresponding dynamic programming solution.

Session 12

- Solve one NP-Complete problem and use the solution to solve another NP-Complete problem.

REFERENCES

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

CSS3105: ARTIFICIAL INTELLIGENCE

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

Course Description:

Artificial Intelligence (AI) is a field that has a long history but is still constantly and actively growing and changing. In this course basics of modern AI as well as some of the representative applications of AI along with huge possibilities in the field of AI, which continues to expand human capability beyond our imagination are taught.

Course Objectives:

- Introduce the foundational principles of AI that drive real world complex applications and practice implementing some of these systems
- Equip students with the tools to tackle new AI problems that they may encounter in life

Course Content:

1. Introduction: Overview and Historical Perspective-Intelligent Agents-Problem Solving by searching-State Space Search: Depth First Search, Breadth First Search, DFID-Informed search & exploration-Heuristic Search- Best First Search-Hill Climbing-Beam Search-Tabu Search-Randomized Search:Simulated Annealing, Genetic Algorithms-Constraint Satisfaction Problems.

2. Finding Optimal Paths: Branch and Bound, A*, IDA*, Divide and Conquer approaches-Beam Stack Search-Problem Decomposition: Goal Trees, AO*, Rule Based Systems -Game Playing: Minimax Algorithm, Alpha-Beta Algorithm, SSS*.

3. Knowledge and reasoning: Propositional Logic- First Order Logic-Soundness and Completeness- Forward and Backward chaining-Resolution-semantic networks-Handling uncertain knowledge – Probabilistic Reasoning – making simple and complex decisions.

4. Planning : Planning problems - Planning with state space search - Partial order planning - Planning Graphs – Planning with Propositional logic-Hierarchical planning - Multi agent planning.

5. Learning: Forms of learning - Inductive learning - Learning decision trees - Explanation based learning - Statistical learning - Instantance based learning – Neural networks-Reinforcement learning.

REFERNCES:

1 Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach, 3e, Prentice Hall, 2009
2. Deepak Khemani. A First Course in Artificial Intelligence, 1e, McGraw Hill Education, 2017
3. Stefan Edelkamp and Stefan Schroedl. Heuristic Search: Theory and Applications, 1e, Morgan Kaufmann, 2011
4. Zbigniew Michalewicz and David B. Fogel. How to Solve It: Modern Heuristics. Springer; 2e, 2004
5. Elaine Rich and Kevin Knight. Artificial Intelligence, 3e, Tata McGraw Hill, 2017
6. Patrick Henry Winston. Artificial Intelligence, 1e, Pearson, 2002

CSS3106: HUMAN-COMPUTER INTERACTION

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

Course Description:

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

Course Objectives:

- To understand basic HCI concepts and definitions
- To understand the different types of interfaces
- To study and design multimodal interfaces
- To design & develop interfaces for diversified users

Course Content:

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

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

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

4. Domain specific design – HCI in healthcare – games – older adults – kids – Physical disabilities – Perpetual Impairments – Deaf and Hard of Learning users.

5. Developments process – requirement specification – User experiences and HCI – Usability Engineering life cycle – Task analysis – prototyping tools and techniques – scenario based design – Participatory design – Testing and evaluation – Usability testing – Inspection based evaluation – Model based evaluation.

REFERNCES:

1. Andrew sears, Julie A Jacko, Lawrence, The human computer interaction hand book: fundamentals, evolving technologies and emerging applications, 1e, Erlbaum Associates, 2008
2. Alan Dix, Janet Finlay, Gregory D Abowd, Russell Beale, Human - Computer Interaction, 3e, Pearson, 2012
3. Helen Sharp, Yvanno Rogers and Jenny Preece, Interaction Design: Beyond human Computer Interaction, 1e, John Wiley, 2011
4. Jan Noyes, Chris Baber, User centred design of systems, 1e, Springer, 2013

CSS3107: INFORMATION RETRIEVAL AND WEB SEARCH

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

Course Description:

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

Course Objectives:

- Basic and advanced techniques for text-based information systems: efficient text indexing;
- Boolean and vector space retrieval models; evaluation and interface issues
- Web search including crawling, link-based algorithms, and Web metadata
- Understand the dynamics of the Web by building appropriate mathematical models.
- Build working systems that assist users in finding useful information on the Web

Course Content:

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

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

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

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

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

REFERENCES:

1. Ricardo Baezce Yates, BerthierRibeiro-Neto , Modern Information Retrieval: The Concepts and Technology behind Search, 2e, ACM Press, 2011
2. Christopher D. Manning, PrabhakarRaghavan and HinrichSchütze , Introduction to Information Retrieval, 1e, Cambridge University Press, 2008
3. Bruce Croft, Donald Metzler and Trevor Strohman, Search Engines: Information Retrieval in Practice, 1e, AW, 2009

CSS3108: FUNCTIONAL PROGRAMMING

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

Course Description:

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

Course Objectives:

- To have a theoretical understanding of functional programming.
- To develop the ability to design and implement functional programs.

Course Content:

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

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

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

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

5. Programming imperatively in Haskell – The IO Monad – Why Monads are Necessary – The State Monad– ST Monad – Mutable and Immutable Arrays – Parsing using Monads – Applications – Fault-tolerant systems – Financial analysis – Comparison to other functional languages.

REFERENCES

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

CSS3109: SOFTWARE QUALITY MANAGEMENT

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

Course Description:

This course discusses basic software project quality management principles and techniques as they relate to software project planning, monitoring and control. This course describes the basics of software verification and validation planning with an emphasis on software peer reviews and software testing. The course also covers software configuration management, technical metrics for software.

Course Objectives:

- Understand the basics and benefits of software quality engineering
- Plan, implement and audit a Software Quality Management program for their organization
- Select, define, and apply software measurement and metrics to their software products, processes and services
- Understand the fundamentals of the configuration management process to include configuration identification, configuration control, status accounting, and audits

Course Content:

1. Introduction to software quality: Software Quality - Hierarchical models of Boehm and McCall - Quality measurement - Metrics measurement and analysis - Gilb's approach -GQM Model

2. Tools for Quality - Ishikawa's basic tools - CASE tools - Defect prevention and removal - Reliability models - Rayleigh model - Reliability growth models for quality assessment

3. Testing for reliability measurement Software Testing - Types, White and Black Box, Operational Profiles - Difficulties, Estimating Reliability, Time/Structure based software reliability - Assumptions, Testing methods, Limits, Starvation , Coverage, Filtering, Microscopic Model of Software Risk

4. Software reliability and availability - standards and evaluation of process - ISO 9000 - SEI Capability Maturity Model (CMM) - Software configuration management -

5. Technical metrics for software - metrics for the analysis model - metrics for design model - metrics for source code - metrics for testing - metrics for maintenance - technical metrics for object oriented systems - distinguishing characteristics - class oriented metrics -operation oriented metrics - testing metrics - project metrics

REFERENCES:

1. Allan C. Gillies, Software Quality: Theory and Management, 3e, Cengage, 2003
2. Ron S Kenett, E. R Baker, Software Process Quality- Management and Control, 1e, CRC, 1999
3. Stephen H. Kan , Metrics and Models in Software Quality Engineering, 1e, AW, 2014
4. Patric D. T.O connor , Practical Reliability Engineering, 5e, John Wesley & Sons, 2011
5. Roger S. Pressman, Software Engineering - A practitioners approach, 8e, McGraw Hill,2014

CSS3110: PATTERNS IN SOFTWARE ENGINEERING

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

Course Description:

Software architecture and design requires to be warned against subtle issues that can cause major problems during implementation. Often, people only understand how to apply certain software architecture and design techniques to certain problems. Formatting and applying these techniques to a broader range of problems is, by itself, a complex problem. Patterns in the areas of software architecture and design provide general solutions, documented in a format that doesn't require specifics tied to a particular problem.

In addition, patterns give a more educated vocabulary to the software architects and designers while expressing the various scenarios of software interactions.

Course Objectives:

- To give a comprehensive overview of recurring patterns in software development
- To impart the technical details on various patterns
- To provide insight into pattern based development

Course Content:

1. Patterns – Patterns category – Relationships between patterns – Patterns and Software Architecture – Architectural patterns – Idioms – Pattern systems – Documentation of Patterns

2. Analysis Patterns – Patterns in analysis – business patterns – Support patterns – Patterns for typed models – Association patterns

3. Design Patterns – Catalog of design patterns –Case study implementation– Creational – Structural Patterns –Behavioural patterns

4. Patterns of Enterprise application architecture – Object relational patterns - Web presentation patterns – Distribution patterns – Concurrency patterns

5. Business Process Improvement Patterns – Pedagogical patterns – Pattern languages –Anti-patterns – Major criticisms

REFERENCES

1. Frank Buschmann et al, Pattern-Oriented Software Architecture Volume 4 & 5, Wiley, 2007
2. Bernd Bruegge, Allen H Dutoot, Object-oriented Software Engineering using UML, Patterns & Java, 3e, Pearson, 2013
3. Martin Fowler , Analysis Patterns: Reusable Object Models, 1e, Addison-Wesley, 1997
4. Eric Gamma et al, Design Patterns: Elements of Reusable Object-Oriented Software, 1e, Pearson, 2008
5. Martin Fowler, Patterns of Enterprise Application Architecture, 1e, AW, 2002
6. Hohpe, Gregor, Bobby Woolf, Enterprise Integration Patterns, 1e, Addison-Wesley, 2005
7. Kircher, Michael, Markus Völter and UweZdun, Remoting Patterns: Foundations of Enterprise, Internet and Realtime Distributed Object Middleware , 1e , John Wiley & Sons, 2004
8. Kaplan, Jonathan, William C. R. Crawford, J2EE Design Patterns, 1e, O'Reilly, 2003

CSS3201: SOFTWARE ARCHITECTURE

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

Course Description:

This course introduces the essential concepts of software architecture. Software architecture is an abstract view of a software system distinct from the details of implementation, algorithms, and data representation. Architecture is, increasingly, a crucial part of a software organization's business strategy.

Course objectives:

- To understand the relationships between system qualities and software architectures.
- To study software architectural patterns and their relationship to system qualities
- To know software architecture documentation and reuse

Course Content:

1. The architecture Business Cycle (ABC) – Roots of Software architecture - Software architecture definitions and importance – Architectures and quality attributes -Architectural Styles - Architectural views: Need for multiple views – Some representative views – Conceptual View – Module view – Process view – Physical view – Relating the views to each other – The Software Architecture analysis Method (SAAM).

2. Life cycle view of architecture design and analysis – Eliciting quality attributes - QAW – Design of architecture - the ADD method – Evaluating architecture -ATAM method

3. Architecture-based development Product lines – cost and benefits of product line approach – product line activities – practice areas – patterns – PLTP – phased approach for adopting product lines

4. Software Architecture for Big Data Systems - Big Data from a software Architecture Perspective-Horizontal Scaling Distributes Data - Big Data – A complex software engineering problem-Software Engineering at Scale -Enhancing Design Knowledge for Big Data Systems-QuA Base – A Knowledge Base for Big Data System Design

5. Case study of J2EE/EJB - Future of software architecture

REFERENCES:

1. Len Bass, Software Architecture in Practice, 3e, Pearson, 2013
2. G.Zayaraz , Quantitative approaches for Evaluating Software Architectures: Frame Works and Models, 1e, VDM Verlag, 2010
3. Klaus Pohl et.al, Software Product Line Engineering: Foundations, Principles and Techniques, 1e, Springer, 2011
4. Web resource: Ian Gorton, Software Architecture for Big Data Systems https://www.sei.cmu.edu/webinars/view_webinar.cfm?webinarid=298346

CSS 3202: AGILE SOFTWARE ENGINEERING

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

Course Description:

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

Course Objectives:

- To introduce the changing concepts of software product development following the agile methods
- To equip the students with recent advances in software testing and refactoring that makes the product development more streamlined and efficient
- To provide an understanding of project management and its principles in a contemporary iterative, incremental Agile project environment

Course Content:

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

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

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

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

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

REFERENCES

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

CSS3203: SEMINAR

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

Course Description:

The student has to prepare and deliver a presentation on a research topic suggested by faculty member before the peer students and staff. They also have to prepare a comprehensive report of the seminar presented

Course Objective:

- Review and increase their understanding of the specific topics tested.
- Inculcating presentation and leadership skills among students
- Offering the presenter student an opportunity of interaction with peer students and staff

CSS3204: DESIGN OF REAL-TIME/EMBEDDED SOFTWARE

Core/Elective: Elective **Semester:** 2 **Credits:** 3

Course Description:

This course describes software-engineering techniques to develop software for embedded systems. This course examines requirements analysis, the definition of object structure and behaviour, architectural and mechanistic design, and more detailed designs that encompass data structure, operations, and exceptions. The object-based Unified Modeling Language (UML) is used to describe the structural and behavioral aspects critical to real-time systems

Course Objectives:

- To understand the principles of software design for resource constrained devices
- To understand real-time/embedded software modelling with UML
- To understand and apply real-time design patterns

Course Content:

1. Embedded / Real-Time Systems: Definitions and Issues - Object-Oriented Methods and the Unified Modeling Language – Basic concepts of Real-Time Systems - Safety critical systems – Object oriented process for embedded systems.

2. Real Time Operating Systems: Case studies of QNX, VxWorks, Windows CE

3. Requirement Analysis of Real-time systems: Use cases – Heuristics for good Requirement analysis diagrams – Structural Object Analysis: Key strategies for object identification - Heuristics for good class diagrams – Behavioural Object analysis: UML State charts – State chart heuristics

4. Architectural Design: Tasking Model, Component Model, Deployment Model, Safety / Reliability Model – Mechanistic Design – Detailed Design – Performance Analysis of Real Time Systems: Real Time Scheduling Theory

5. Dynamic Modelling - UML and Design Patterns – Real Time Design Patterns – Debugging and Testing – Real Time Frameworks – Design Automation Tools: Rhapsody OXF

REFERENCES:

1. Hassan Gomaa, Real-Time Software Design for Embedded Systems, 1e, Cambridge University Press, 2016
2. Hassan Gomaa, Designing Concurrent, Distributed, and Real-Time Applications with UML –, 1e, AW, 2013
2. Bruce Powel Douglass, Real Time UML: Advances in the UML for Real-Time Systems, 3e, Pearson, 2016
3. Bruce Powel Douglass, Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems, 1e, AW Object Technology Series, 2002
4. Sebastian Gerard and Marte:BranSelic, Modeling and Analysis of Real-Time and Embedded Systems with UML,1e, Elsevier, 2013

CSS3205: SOFTWARE AGENT SYSTEMS

Core/Elective: Elective **Semester:** 2 **Credits:** 3

Course description:

This course provides a thorough understanding of agent related system development. Software agents are finding their way into areas such as environmental security, climate change, seismic safety, epidemic prevention, detection and response, computer emergency response and human and societal dynamics

Course objectives:

- To understand Agent development
- To gain Knowledge in Multi agent and intelligent agents
- To Understand Agents and security
- To gain Knowledge in Agent Applications

Course Content

1. The agent landscape – The smart agent framework: Introduction – Initial concepts – Entities-Objects – Agents – Autonomy – Tropistic agent – Specification structure of SMART. – Agent relationships – An operational analysis of Agent relationships.

2. Sociological Agents - Autonomous Interaction - Contract Net as a global directed system – Computational Architecture for BDI agents – Evaluating social dependence networks – Normative agents.

3. Intelligent Agents –Deductive Reasoning Agents – Practical reasoning agents - Reactive agents – Hybrid Agents – Understanding Each other – Communicating – Methodologies

4. Modeling multi agent system with AML – JADE:Java Agent development frame work – wireless sensor networks and software Agents – Multi agent Planning Security and anonymity in agent systems.

5. Multi Agent system: Theory approaches and NASA applications – Agent based control for multi-UAV information collection- Agent based decision support system for Glider pilots – Multi agent system in E- Health Territorial Emergencies – Software Agents for computer network security- Multi-Agent Systems, Ontologies and Negotiation for Dynamic Service Composition in Multi-Organizational Environmental Management.

REFERENCES:

1. Mohammad Essaaidi, Maria Ganzha, and MarcinPaprzycki, Software Agents, Agent Systems and Their Applications, 1e, IOS Press, 2012.
2. Mark D .Inverno and Michael Luck, Understanding Agent Systems, 1e, Springer,2010.
3. Michael Wooldridge, An Introduction to Multi Agent Systems, 1e, John Wiley & Sons Ltd., 2009.
4. Lin Padgham, Michael Winikoff, Developing Intelligent Agent Systems: A Practical Guide, 1e, John Wiley & Sons Ltd., 2004.
5. Bradshaw, Software Agents, 1e, MIT Press, 1997.
6. Richard Murch, Tony Johnson, Intelligent Software Agents, 1e, Prentice Hall, 2000.

CSS3206: NETWORK FORENSICS

Core/Elective: Elective **Semester:** 2 **Credits:** 3

Course Description:

This course will introduce the student to the essential aspects of information security and network forensics. The student will be provided with the tools, techniques and industry accepted methodologies so that upon completion of the course the student will be able to describe key concepts of network security and forensics and how those concepts apply to themselves and their organization.

Course Objectives:

- Describe key principles, such as defense in depth and demilitarized zones (DMZ)
- Provide an overview of the requirement for intrusion detection systems (IDS) and their implementation
- Provide an overview of network security devices and infrastructures, including proxy servers and firewalls
- Describe the methodologies used in network forensics
- Describe data hiding and obfuscation and outline obfuscation methods

Course Content:

1. Introduction to Security -CIA and AAA - protecting against Intruders - Users, Systems, and Data -Services, Role-Based Security, and Cloud Computing - Security and Forensic Computing - ISO 27002 - Risks -Risk Management/Avoidance - Security Policies -Defining the Policy - Example Risks - Defense-in-Depth - Gateways and DMZ (Demilitarized Zones) - Layered Model and Security - Encryption - Layered Approach to Defense

2. Intrusion Detection Systems Types of Intrusion - Attack Patterns - Host/Network-Based Intrusion Detection - Placement of the IDS - Snort - Example Rules - Running Snort -User, Machine, and Network Profiling – Honey pots - In-Line and Out-of-Line IDSs - False and True - Customized Agent-Based IDS

3. Network Security Elements Objectives - Introduction - Router (Packet Filtering) Firewalls - Network Address Translation - PIX/ASA Firewall - Proxy Servers

4. Network Forensics Key Protocols - Ethernet, IP, and TCP Headers - TCP Connection – ARP – SYN - Application Layer Analysis - FTP - ICMP - DNS - Port Scan - SYN Flood - Spoofed Addresses - Application Layer Analysis - HTTP - Network Logs on Hosts – Tripwire

5. Data Hiding and Obfuscation Obfuscation Using Encryption - Obfuscation through Tunneling - Covert Channels - Watermarking and Stenography - Hiding File Contents - File Contents

REFERENCES:

1. Buchanan, William J, Introduction to Security and Network Forensics, 1e, CRC Press, 2011
2.EC-Council, Computer Forensics: Investigating Network Intrusions and Cyber Crime, 2e, Cengage, 2017
3. Michael E. Whitman, Herbert J. Mattord, Principles of Information Security, 2e, Cengage Learning Pub., 2012
4. Eoghan Casey, Digital Evidence and Computer Crime Forensic science, Computers and Internet, 2e, Elsevier Academic Press, 2011
5. ChristofPaar, Jan Pelzl, Understanding Cryptography: A Textbook for Students and Practitioners, 2e, Springer, 2010

CSS3207: ENTERPRISE APPLICATION INTEGRATION & BUSINESS PROCESS MANAGEMENT

Core/Elective: Elective **Semester:** 2 **Credits:** 3

Course description:

The course will introduce the major design, implementation and deployment issues regarding system integration, data-oriented cross-platform integration, e-business applications implementation and the security considerations in enterprise level multi-location systems integration. Business Process Management (BPM) is the set of concepts, methods, and tools that help organizations define, implement, measure and improve their end-to-end processes

Course Objectives

The course will introduce the concepts and techniques related to the service-oriented as well as the data-oriented application integration approaches. Reasonable emphasis will be given for the middleware technologies and large-scale application integration standards. The course will also focus on the methods and techniques required to analyze, design, implement, automate, and evaluate business processes and workflows related to process-aware information systems.

Course Content

1. Application Integration Overview: Problems in large-scale application integration, Business & Service Oriented Integration: XSLT Processing, EnterpriseServiceBus, Web services introduction, Second generation web services –messaging –security –metadata.

2.Middleware: Basics and types, Distributed Transactions, Two Phase Commit, Message-oriented Middleware (MoM), Java middleware, Integration Servers, XML and other standards. Commercial examples.

3. Data-orientated Application Integration: Loosely couples systems, Data oriented programming, Data flow architecture, Event driven architecture. Integration with Business systems: Legacy systems integration –challenges, External system integration standards –RosettaNet –ebXML –UCCNet.

4. Integration standards: SOAP, XML-RPC, REST. Vertical Application Integration.The Application Integration Process.Reliability and Fault-tolerance.Ontologies.Data integration patterns.

5. Business Process Analysis and Design: Workflows & BPMS, Introduction to BPMN, Managing Processes, Components of process models, Process Management Maturity, Rules, Integrating rules with processes, Process dashboards. Commercial solutions.

REFERENCES:

1. Thomas Erl, Service Oriented Architecture: A field guide to Integrating XML and Web Services, 1e, Prentice Hall, 2004
2. G.Hohpe and B. Woolf Enterprise Integration Patterns: Designing, Building and Deploying Messaging Solutions, 1e, AW Professional, 2003
3. D. Linthicum, Next Generation Application Integration: From Simple Information to Web Services, 1e, Addison Wesley, 2003
4. Michael Havey , Essential Business Process Modeling, 1e, O'Reilly Media, 2005

CSS3208: ADVANCED DATA MINING

Core/Elective: Elective **Semester:** 2 **Credits:** 3

Course Description:

Data mining is the science of extracting hidden information from large datasets. This course offers clear and comprehensive introduction to both data mining theory and Practice. All major data mining techniques will be dealt with and how to apply these techniques in real problems are explained through case studies.

Course Objectives:

- Introduce the fundamental concepts of data and data analysis
- Case based study of specific data mining tasks like Clustering, Classification, Regression, Pattern Discovery and Retrieval by Content.
- Introduce algorithms for temporal data mining and spatial data mining.

Course Content:

1. Statistical descriptions of data-data visualization-measuring data similarity and dissimilarity-data pre-processing-data cleaning-data integration-data reduction-data transformation-data warehouse modeling-design-implementation-data cube technology-queries by data cube technology-multidimensional data analysis in Cube space

2. mining frequent patterns, associations and correlations – pattern mining in multidimensional space- colossal patterns- approximate patterns- applications- Mining data streams-Mining Sequence patterns in transactional databases- mining sequence pattern in Biological Data

3. Classification and prediction- decision tree induction-Bayesian classification-rule-based classification- neural networks-support vector machines-lazy learners-genetic algorithms-model evaluation-Cluster analysis- portioning methods- hierarchical methods- density based methods-grid based-probabilistic model based clustering- clustering high dimensional data-constraint based clustering- clustering high dimensional data-graph clustering methods

4. Outlier detection- outliers and outlier analysis- outlier detection methods-statistical approaches-proximity based approaches- clustering based approaches- classification based approaches-mining contextual and collective outliers- outlier detection in High-Dimensional data

5. Time series representation and summarization methods-mining time series data -Spatial data mining-spatial data cube construction-mining spatial association and co-location patterns-spatial clustering and classification methods-spatial trend analysis- Multimedia data mining-text mining- mining world wide web- trends in Data mining

REFERNCES:

1. TheophanoMitsa, Temporal Data mining, 1e, CRC Press, 2010
2. Jiawei Han &MichelineKamber, Jian Pei, Data mining concepts and techniques, 1e, Elsevier, 2014
3. A B M Showkat Ali, Saleh A Wasimi, Data mining methods and Techniques, Cengage

Learning, 2009
4. G.K Gupta, Introduction to Data mining with case studies, 1e, PHI, 2008

CSS3209: FUZZY SET THEORY: FOUNDATIONS AND APPLICATIONS

Core/Elective: Core **Semester:** 2 **Credits:** 3

Course Description:

This course concentrates on fuzzy set theory and its application. This includes the concepts, and techniques from fuzzy sets and fuzzy logic to enhance machine learning techniques.

Course Objectives:

- To review the concepts of fuzzy set theory.
- To understand the fuzzy set theory techniques
- To use the theory in optimization problems
- To apply the theory to enhance machine learning techniques

Course Content:

1. Crisp sets and Fuzzy sets - Introduction - crisp sets an overview-the notion of fuzzy sets-basic concepts of fuzzy sets- membership functions - methods of generating membership functions-Defuzzification methods-operations on fuzzy sets- fuzzy complement- fuzzy union- fuzzy intersection- combinations of operations-General aggregation operation

2. Fuzzy arithmetic and Fuzzy relations-Fuzzy numbers-arithmetic operations on intervals-arithmetic operations on fuzzy numbers-fuzzy equations- crisp and fuzzy relations-binary relations- binary relations on a single set - equivalence and similarity relations- compatibility or tolerance relation

3. Fuzzy measures - Fuzzy measures - belief and plausibility measure - probability measures - possibility and necessity measures- possibility distribution- relationship among classes of fuzzy measures.

4. Fuzzy Applications-Fuzzy approximate reasoning- Fuzzy Expert System-Fuzzy systems-Fuzzy controllers-Fuzzy Neural Networks- Fuzzy automata-Fuzzy Dynamic systems

5. Fuzzy Clustering-Fuzzy Pattern Recognition-Fuzzy image processing - Fuzzy data bases and information retrieval-Fuzzy Decision making - Fuzzy systems and Genetic algorithms - Fuzzy regression.

REFERNCES:

1. George J Klir and Tina AFolger: Fuzzy Sets, Uncertainty and Information, Fuzzy Sets, Uncertainty and Information, 1e, Pearson Education, 2015
2. George J Klir and Bo Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, 1e, Pearson Education, 2015.
3. Timothy J Ross: Fuzzy logic with Engineering Applications, 3e, Wiley, 2011.
4. H. J. Zimmerman: Fuzzy Set theory and its Applications,4e, Springer,2001.

CSS3210: COMPLEX NETWORKS: THEORY AND APPLICATIONS

Core/Elective: Core **Semester:** 2 **Credits:** 3

Course description:

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

Course Objectives

- Representation and analysis of complex networks

Course Content

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

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

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

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

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

REFERENCES:

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

CSS 3211: DATA SCIENCE & BIG DATA ANALYTICS

Core/Elective: Elective **Semester:** 2 **Credits:** 3

Course Description:

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

Course Objectives:

- To examine fundamental statistical techniques for data analytics
- To attain certain amount of statistical and mathematical sophistication to deal with the subject
- To gain thorough understanding on specific technologies for advanced analytics with Big Data
- To provide a guidance on operationalizing Big Data analytics projects

Course Content:

1. Big data ecosystem – Data analytics life cycle – Data preparation – Model planning – Model building – Communicate results – Operationalize – Case studies

2. Basic data analytics using R: R data structures - Exploratory data analysis - Statistical methods for evaluation – Data visualization using R: Visualization of categorical data, time series, distributions – Plots and Maps - Visualization packages

3. Analytical theory and methods: Clustering: K Means – K Medoids – CLARA – Hierarchical clustering: Agglomerative clustering – Cluster validation – Association rules – Regression – Case studies

4. Classification – Time series analysis: Characteristics – ARIMA models – Text analysis: Processing text – Categorizing and tagging – Text summarization & classification - Case studies

5. Advanced analytics: Technology and tools – MapReduce and Hadoop – Analytics with RDBMS -NoSQL – Big data analytics stream data - In-database analytics

REFERENCES

1. EMC Education Services, Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data, Wiley; 1st edition (2015)
2. Hadley Wickham, Garrett Grolemund, R for Data Science: Import, Tidy, Transform, Visualize, and Model Data, Shroff/O'Reilly; First edition (2017)
3. Joel Grus, Data Science from Scratch, Shroff (2015)
4. James D. Miller, Statistics for Data Science, Packt Publishing Limited (2017)
5. Thomas Rahlf, Data Visualisation with R: 100 Examples, Springer, 1st ed. (2017)
6. Alboukadel Kassambara, Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning (Volume 1), CreateSpace Independent Publishing Platform (2017)