Syllabus

1	**Title of the course (L-T-P-C)**	**Software Development for Scientific Computing (3-0-0-6)**
2	**Pre-requisite courses(s)**	Exposure to Data Structures and Algorithms, C / C++ / Java / Matlab
3	**Course content**	Algorithmic Patterns in Scientific Computing: dense and sparse linear algebra, structured and unstructured grid methods, particle methods (N-body, Particle-Particle, Particle-in-cell, Particle-in-a- mesh), Fast Fourier Transforms, Implementing PDEs, C++ standard template library (STL), Introduction to debugging using GDB, GMake, Doxygen, Version Control System, Profiling and Optimization, asymptotic analysis and algorithmic complexity. Mixed-language programming using C, Fortran, Matlab, and Python, Performance analysis and high-performance code, Data locality and auto tuning, Introduction to the parallel programming world.
4	**Texts/References**	Stroustrup C++ Language Reference (https://www.stroustrup.com/4th.html) Suely Oliveira, David Steward: Writing Scientific Software: A Guide to Good Style. Cambridge University Press, 2006 Web references to GNU Make, GDB, Git, GProf, and Gcov. Code Complete: A Practical Handbook of Software Construction https://www2.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.html

1	**Title of the course** (L-T-P-C)	**Approximation algorithms (3-0-0-6)**
2	**Pre-requisite courses(s)**	Data Structures and Algorithms (CS201)
3	**Course content**	Introduction, approximation schemes, design and analysis of approximation algorithms - combinatorial algorithms, linear programming based algorithms. Hardness of approximation.
4	**Texts/References**	*Textbook:* *(1) Approximation algorithms. Vazirani, Vijay V. Berlin: springer, 2001.* *Reference:* *(1) The design of approximation algorithms. Williamson, David P., andDavid B. Shmoys. Cambridge university press, 2011.*

1	**Title of the course** **(L-T-P-C)**	**Parametrized Algorithms and Complexity** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Data Structures and Algorithms, Design and Analysis of Algorithms
3	**Course content**	Introduction. Kernelization, Bounded Search Trees, Iterative Compression, Treewidth, Advanced kernelization algorithms. Lower bounds: Fixed- parameter intractability, lower bounds based on ETH, lower bounds for kernelization. Parameterized Algorithms, Kernelization, and Complexity of Graph Modification Problems
4	**Texts/References**	Textbook: (1) Parameterized Algorithms, Marek Cygan, FedorV. Fomin, Lukasz Kowalik. Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Sourabh. Springer. 2015 Reference: (1) Parameterized Complexity, R. G. Downey, and M. R. Fellows. Springer Science and Business Media. 2012

1	**Title of the course** (L-T-P-C)	**Reinforcement Learning** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Basic Probability and Linear Algebra
3	**Course content**	Bandit Algorithms -- Regret based - UCB, Thomson Sampling, PAC Based - Median Elimination, Markov Decision Process Modeling - Bellman Equation, Dynamic Programming Solutions - Value and Policy Iteration, Linear Programming, Model free methods - Monte Carlo and Temporal Difference Methods - Q-learning, Value function Approximation - State Aggregation, Critic Only/Value Based Methods Methods - TD methods, Q- Learning, SARSA, Actor Only/Policy Based methods - Reinforce, Actor-Critic Methods - Policy Gradient, Natural Actor Critic, Deep RL - DQN, A3C, Model Based RL, Integrating Learning and Planning, Case-studies
4	**Texts/References**	1.Richard S. Sutton and Andrew G. Barto, Introduction to Reinforcement Learning, 2nd Edition, MIT Press. 2017. 2. Dimitri Bertsekas and John G. Tsitsiklis, Neuro Dynamic Programming, Athena Scientific. 1996. 3. Bertsekas, Dimitri P. Dynamic Programming and Optimal Control. Vol. 1 and 2. 4th edition, 2012. 4.Algorithms for Reinforcement Learning, Csaba Szepesvári, Morgan & Claypool, 2009. 5.Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems, Sébastien Bubeck and Nicolò Cesa-Bianchi, Foundations and Trends in Machine Learning, Volume 5, Number 1, 2012.

1	**Title of the course** (L-T-P-C)	**Advanced topics in Embedded Computing** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	CS 301
3	**Course content**	Introduction to systems software in embedded platforms Boot loader Embedded Linux kernel (Processes, Threads, Interrupts) Device Drivers Scheduling Policies (including Real Time) Memory Management Optimizations (Data level and Memory level) Embedded Systems Security Introduction to Embedded GPUs and Accelerators Embedded Heterogenous Programming with Open CLApplication Case Study on Embedded Platforms – eg. Neural Network inferencing on Embedded Platforms, Advanced Driver Assistance Systems
4	**Texts/References**	1. Building Embedded Linux Systems, 2nd Edition by Gilad Ben-Yossef, Jon Masters, Karim Yaghmour, Philippe Gerum, O'Reilly Media, Inc. 2008 2. Linux Device Drivers, Third Edition By Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman, O'Reilly Media, Inc. 2005 3. Embedded Systems: ARM Programming and Optimization by Jason D Bakos, Elsevier, 2015 4. Learning Computer Architecture with Raspberry Pi by Eben Upton, Jeff Duntemann, Ralph Roberts, Tim Mamtora, Ben Everard, Wiley Publications, 2016 5. Real Time Systems by Jane S. Liu, 1 edition, Prentice Hall; 2000 6. Practical Embedded Security: Building Secure Resource-Constrained Systems by Timothy Stapko, Elsevier, 2011

1	**Title of the course** (L-T-P-C)	**Advanced Computer Networks** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	
3	**Course content**	1. Circuit, Packet and Virtual Circuit Switching, MPLS 2. Switch Architectures, Buffering Strategies, Input and Output Queuing, IP Buffer Sizing 3. Quality of Service and Scheduling Algorithms 4. IP Address Lookup and IP Packet Classification algorithms 5. Software Defined Networking 6. Next Generation Network Architectures, Network Provisioning and Design, and "Green" (Energy- Efficient) Networking 7. Data Driven Networking *8.* Wireless Networks - MANETs, Sensor Networks, Cellular Networks, Personal Area Networks *9. Content Based Delivery Networks - Principles of data dissemination, aggregation and caching that are applied to sensor networks, Internet of Things and other content-based paradigms. Students will survey recent research publications on opportunistic networks and next generation content-based networking ideas.* *10. Delay tolerant Networks* *11. Network security - authentication, access control, privacy preservation, intrusion detection and prevention* *12. Performance analysis of new Networking ideas using simulation (such as Network Simulator (ns3), GENI testbed, Simulink, Open LTE and Open C-RAN frameworks)*
4	**Texts/References**	Textbook: Computer Networks: A Systems Approach, Larry Peterson and Bruce Davie, 2011. Performance Evaluation of Computer Systems, by Raj Jain, Wiley, 1991. Computer Networking, Kurose and Ross, Addison-Wesley, 2012. Reference: 1.An Engineering Approach to Computer Networking by S. Keshav, 1997, Addison-Wesley Professional Series. 2.Network Routing, by Deepankar Medhi and Karthikeyan Ramasamy, Morgan Kaufmann, 2007. 3.SDN: Software Defined Networks, by Thomas D.Nadeau, Ken Gray, O'Reilly Media, 2013. 4.High Performance Switches and Routers, By H.Jonathan Chao and Bin Liu, Wiley, 2007. Network Algorithmics, by George Varghese, Morgan Kaufmann, 2005

1	**Title of the course** (L-T-P-C)	**FPGA for communication networks prototyping** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	EE 224 Digital System Exposure on Computer Network
3	**Course content**	History and evaluation of FPGAs; FPGA architecture; Introduction to Quartus Prime (vendors and design tools; vendors and programmable logic); Exploiting Simulation tools (e.g., ModelSim); Exploiting FPGAs for multi-domain technologies; Introduction to radio access networks-fronthaul (e.g., common public radio interface); optical network; metro and core networks; Cross-layer design; The role of FPGA in the specified network segments and use case scenarios; In and Out; Clocks and Registers; State Machines; Modular Design; Memories Managing Clocks; I/O Flavors; Exploiting Qsys and Nios II tools
4	**Texts/References**	1.C. Maxfield, "The Design Warrior's Guide to FPGAs: Devices, Tools and Flows", Jun. 2004, eISBN 9780080477138 2.FPGAs For Dummies, 2nd Intel Special Edition. Published by. John Wiley & Sons, Inc 3.William J. Dally, R. Curtis Harting, "Digital Design: A Systems Approach 1st Edition", Cambridge University Press, September 2012, ISBN 9780521199506 4.Verilog by Example: A Concise Introduction for FPGA Design, Blaine C. Readler 5.Course materials: Slides; Notes; Tutorials from Altera website https://www.altera.com/support/training/university/materials-tutorials.html 6.R. Ramaswami, K. Sivarajan, G. Sasaki; "Optical Networks: A Practical Perspective," 3rd Ed., Morgan Kaufmann, ISBN: 9780123740922

1	**Title of the course** (L-T-P-C)	**Software Defined Networking (SDN) and Network Function Virtualization (NFV)** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Exposure to Computer Networks
3	**Course content**	History and evolution of SDN; SDN Architecture (Application, Control, Infrastructure Layer); SDN Interfaces (East/West/North/South-bound interfaces); SDN Security; SDN routing; SDN standards; SDN Controllers; Network Operating Systems and Languages; OpenFlow; Software Switches (e.g. OpenVSwitch); SDN Simulation/Emulation Platforms (e.g. Mininet); Federated SDN networks; SDN Applications and Use Cases; Programming assignment/project; Need for NFV; NFV and SDN Relationship; Virtual Network Functions; Service Function Chaining; NFV Specifications; NFV Architecture; NFV Use Cases; NFV Management and orchestration (MANO); Open-source NFV; Hands-on exercises based on OpenStack/Docker.
4	**Texts/References**	1.Software Defined Networks: A Comprehensive Approach by Paul Goransson and Chuck Black, Morgan Kaufmann Publications, 2014 2.SDN – Software Defined Networks by Thomas D. Nadeau & Ken Gray, O'Reilly, 2013 Software Defined Networking with OpenFlow, By Siamak Azodolmolky, Packt Publishing, 2013 3.Gray, Ken, and Thomas D. Nadeau. Network function virtualization. Morgan Kaufmann, 2016. 4.Zhang, Ying. Network Function Virtualization: Concepts and Applicability in 5G Networks. John Wiley & Sons, 2018. 5.Foundations of modern networking- SDN, NFV, QoE, IoT, and Cloud, William Stallings James Kurose and Keith Ross, "Computer Networking, A Top-Down Approach"

1	**Title of the course** (L-T-P-C)	**Advanced Distributed Systems** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Operating Systems, Data Structures and Algorithms, Programming in C++
3	**Course content**	Synchronization, Global Snapshot and Distributed Mutual Exclusion, Consensus & Agreement, Checkpointing & Rollback Recovery, Deadlock Detection, Termination Detection, Message Ordering & Group Communication, Fault Tolerance and Self-Stabilization, Peer to Peer Systems Mining Data Streams in a distributed systems: filtering data streams, queries on streams, pattern detection Key-Value Storage: Cassandra, HBase Virtualization and Cloud Computing: virtual machines containers Message oriented communication, Publish Subscribe Systems (use case Apache Kafka) Security: Distribution of security mechanisms, access control, and security management.
4	**Texts/References**	Distributed Computing: Principles, Algorithms, and Systems- Ajay D. Kshemkalyani and Mukesh Singhal Mining Massive data sets- Jure Leskovec, Anand Rajaraman, Jeff Ullman Distributed Algorithms – An Intuitive Approach (The MIT Press) by Wan Fokkink Distributed Algorithms-Nancy Lynch

1	**Title of the course** (L-T-P-C)	**Reinforcement Learning Laboratory** **(0-0-3-3)**
2	**Pre-requisite courses(s)**	Currently taking reinforcement learning theory course
3	**Course content**	The lab will closely follow the theory course. The idea is to have the students implement the basic algorithms on different topics studied in the reinforcement learning theory course.
4	**Texts/References**	1.Richard S. Sutton and Andrew G. Barto, Introduction to Reinforcement Learning, 2nd Edition, MIT Press. 2017. 2.Dimitri Bertsekas and John G. Tsitsiklis, Neuro Dynamic Programming, Athena Scientific. 1996. 3.Bertsekas, Dimitri P. Dynamic Programming and Optimal Control. Vol. 1 and 2. 4th edition, 2012. 4.Algorithms for Reinforcement Learning, Csaba Szepesvári, Morgan & Claypool, 2009. 5.Regret Analysis of Stochastic and Nonstochastic Multi- armed Bandit Problems, Sébastien Bubeck and Nicolò Cesa-Bianchi, Foundations and Trends in Machine Learning, Volume 5, Number 1, 2012.

1	**Title of the course** (L-T-P-C)	**Statistical Pattern Recognition Laboratory** **(0-0-3-3)**
2	**Pre-requisite courses(s)**	Currently taking statistical pattern recognition theory course
3	**Course content**	The lab will closely follow the theory course. The idea is to have the students implement the basic algorithms on different topics studied in the statistical pattern recognition theory course.
4	**Texts/References**	1.R.O.Duda, P.E.Hart and D.G.Stork, Pattern Classification, John Wiley, 2001. 2.C.M.Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

1	**Title of the course** (L-T-P-C)	**Statistical Pattern Recognition** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Multivariate Calculus and Linear Algebra, Probability, Programming
3	**Course content**	Bayesian Decision Making and Bayes Classifier, Parametric and Non Parametric Estimation of Densities, General Linear Models, Discriminative Learning based Models, Dimensionality Reduction Techniques, Empirical and Structural risk minimization, Ensemble Methods - Bagging, Boosting, Pattern Clustering, Graphical Models, Statistical Learning Theory
4	**Texts/References**	1.R.O.Duda, P.E.Hart and D.G.Stork, Pattern Classification, John Wiley, 2001. 2.C.M.Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

1	**Title of the course** (L-T-P-C)	**Special Topics in Hardware Systems** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Digital Systems Computer Architecture
3	**Course content**	Introduction Processors Memories Special Processors and Accelerations Architecture for Machine Learning Detailed Syllabus is attached in the appendix
4	**Texts/References**	1.J .L. Hennessy, D. A. Patterson: Computer Architecture :A Quantitative Approach(The Morgan Kaufmann Series in Computer Architecture and Design),2011 2.J. P. Shen, M. H. Lipasti: Modern Processor Design: Fundamentals of Superscalar Processors , Waveland Press, 2013 3.B. Reagan, R. Adolf, P. Whatmough, G.Y-Wei, D.Brooks:Deep Learning for Computer Architects Synthessis Lectures on Computer Architecture, Morgan & Claypool,2017

1	**Title of the course** (L-T-P-C)	**Logic and Applications** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Discrete Mathematics, Theory of computation
3	**Course content**	Module 1 : Propositional Logic: Natural deduction, semantics, soundness, completeness, compactness, normal forms,Horn clauses and satisfiability. Module 2: Predicate Logic: Natural deduction, resolution, undecidability, expressiveness. Module 3: Some decidable fragments of first-order logic and their decision procedures: propositional logic, equality with uninterpreted functions, linear arithmetic, Presburger logic ,bit vectors, arrays, pointer logic. Module 4: SAT and SMT solvers: theory and practice: Decision procedures for combinations of first-order theories: Nelson-Oppen, Shostak, Satisfiability Modulo Theories (SMT) Combination with SAT solvers: eager, lazy approaches. *Student is required to do a small project using a SAT/SMT solver.*
4	**Texts/References**	1.Logic in Computer Science, Michael Huth and Mark Ryan,Cambridge University Press. 2.Mathematical Logic for Computer science,Mordechai Ben-Ari, Springer. 3. Logic for Computer Scientists, Uwe Schoning, Birkhauser. 4.SAT/SMT by example, Dennis Yurichev.

1	**Title of the course** (L-T-P-C)	**Special Topics in Automata and Logics** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Discrete Mathematics, Theory of computation, Logic and its applications.
3	**Course content**	This course aims at giving an introduction to the theory of automata working on infinite words and infinite trees and connections thereof to logics. These automata and related logics are of fundamental importance in the areas of formal specification and verification of reactive systems. If time permits, we will also discuss some basic results in finite model theory. Below is a list of topics, which will be discussed in this course. Automata on finite words - equivalence of MSO and automata; Automata on infinite words â€“ different acceptance conditions; Closure properties and equivalence of different acceptance conditions and related translations Determinization and complementation results; Equivalence of automata and MSO and decidability of MSO; Automata on infinite trees - different acceptance conditions; Closure properties and comparison of expressive power of different acceptance conditions and related translations Complementation result for tree automata via parity games; Equivalence of MSO and tree automata; Decidability of MSO over tress; Parity games and determinacy; Ehrenfeucht-Frasse games in logics and applications
4	**Texts/References**	1.Wolfgang Thomas: Automata on infinite objects, Handbook of theoretical computer science (vol B): formal methods and semantics, Elsevier. 2.Wolfgang Thomas: Languages, automata, and logic, Handbook of formal languages, vol3: beyond words, Springer-Verlag. 3.Dominique Perrin, Jean-Eric Pin: Infinite words,Elsevier 4.Erich Gradel, Wolfgang Thomas, Thomas Wilke: Automata, logics, and infinite games: a guide to current research. LNCS, Springer-Verlag. (1)Leonid Libkin: Elements of finite model theory, Springer-Verlag.

1	**Title of the course** (L-T-P-C)	**Compilers - Principles and Implementation (3-0-0-6)**
2	**Pre-requisite courses(s)**	Exposure to Data Structures and Algorithms, Computer Architecture, Automata Theory
3	**Course content**	Structure of a compiler, the compiled and interpreted execution models. Lexical analysis and parsing using lex and yacc. Scope and visibility analysis. Data layout and lifetime management of data. Runtime environment. Dynamic memory allocation and Garbage collection. Translation of expressions, control structures, and functions. Code generation and local optimizations, Lattice theory, register allocation, instruction scheduling, optimizations - dataflow, control flow, reaching definitions, and liveness analysis, code transformation-tiling, fusion.
4	**Texts/References**	1. Alfred V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D.Ullman: Compilers: Principles, Techniques, and Tools, 2/E, AddisonWesley 2007. 2. Andrew Appel: Modern Compiler Implementation in C/ML/Java, Cambridge University Press, 2004 3. Dick Grune, Henri E. Bal, Cerial J.H. Jacobs and Koen G. Langendoen: Modern Compiler Design, John Wiley & Sons, Inc. 2000. 4. Michael L. Scott: Programming Language Pragmatics, Morgan Kaufman Publishers, 2006. 5. Fisher and LeBlanc: Crafting a Compiler in C.

1	**Title of the course** (L-T-P-C)	**Topics in Stochastic Control and Reinforcement Learning (3-0-2-8)**
2	**Pre-requisite courses(s)**	Probability, Linear Algebra and Multi-variable Calculus
3	**Course content**	Bandit Algorithms -- Regret based - UCB, Thomson Sampling, PAC Based - Median Elimination, Optimality of Bandit Algorithms, Finite Horizon Problems, Infinite Horizon Problems - Total Cost Criterion, Discounted Cost Criterion, Average Cost Criterion, Markov Decision Process (MDP), Partially observable MDP (POMDP), and Dynamic Programming Solutions - Value and Policy Iteration, Model free methods - Monte Carlo and Temporal Difference Methods - Q-learning, SARSA, on/off-policy learning, Stochastic Approximation: Single and multi-timescale stochastic approximation, ordinary differential equation based convergence results. Convergence of SARSA and Q- learning, Value function Approximation - State Aggregation, Critic Only/Value Based Methods - TD methods, gradient temporal difference learning, Actor Only/Policy Based methods - Reinforce, Actor-Critic Methods - Policy Gradient, Natural Actor Critic, Deep RL - DQN, A3C, Regret Based RL - Upper Confidence Reinforcement Learning, Posterior sampling Reinforcement Learning
4	**Texts/References**	1.Bertsekas, Dimitri P. Dynamic Programming and Optimal Control. Vol. 1 and 2. 4th edition, 2012. 2.Algorithms for Reinforcement Learning, Csaba Szepesvári, Morgan & Claypool, 2009. 3.Dimitri Bertsekas and John G. Tsitsiklis, Neuro Dynamic Programming, Athena Scientific. 1996. 4.Richard S. Sutton and Andrew G. Barto, Introduction to Reinforcement Learning, 2nd Edition, MIT Press. 2017. 5.Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems, Sébastien Bubeck and Nicolò Cesa-Bianchi, Foundations and Trends in Machine Learning, Volume 5, Number 1, 2012.

1	**Title of the course** (L-T-P-C)	**Topics in Data Structures and Algorithms** **(2-0-2-6)**
2	**Pre-requisite courses(s)**	No prerequisites
3	**Course content**	Data structures - arrays, linked lists, stacks and queues, heap and binary search tree. Algorithm design techniques - divide and conquer, greedy, and dynamic programming. Algorithms for graph problems. Asymptotic notations, Complexity lower bounds and NP-completeness.
4	**Texts/References**	Textbook: Cormen, Leiserson, Rivest and Stein, **Introduction to Algorithms**, 3rd edition, by, MIT Press, 2009. Reference: 1.Sanjoy Dasgupta, Christos Papadimitriou and Umesh Vazirani, **Algorithms**, McGraw Hill Education, 2008. 2.Kleniberg and Tardos, **Algorithm Design**, 1st edition, Pearson, 2006.

1	**Title of the course** (L-T-P-C)	**Data Structures** **(3-0-0-3)**
2	**Pre-requisite courses(s)**	None
3	**Course content**	Asymptotic notations Theta, Big-oh, Big-omega, little-oh, little-omega Data structures-arrays,linked lists, stacks and queues, heap and binary search tree, hash tables
4	**Texts/References**	**Textbook**: 1.Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to algorithms. MIT press; 2009. **Reference**: 1.Brass P. Advanced data structures. Cambridge: Cambridge University Press; 2008.

1	**Title of the course** (L-T-P-C)	**Algorithms** **(3-0-0-3)**
2	**Pre-requisite courses(s)**	
3	**Course content**	Algorithm design techniques - divide and conquer, greedy, and dynamic programming; Algorithms for graph problems. Complexity, lower bounds and NP-completeness.
4	**Texts/References**	**Textbook**: Cormen TH, Leiserson CE,Rivest RL,Stein C.Introduction to algorithms. MIT press; 2009. **Reference**: Dasgupta S, Papadimitriou CH, Vazirani UV. Algorithms. New York: McGraw-Hill Higher Education; 2008..

1	**Title of the course** (L-T-P-C)	**Introduction to Reinforcement Learning** **(2-0-2-6)**
2	**Pre-requisite courses(s)**	Exposure to Probability Models, Stochastic Process and its Applications, Math for Data Science I and II
3	**Course content**	andit Algorithms -- Regret based - UCB, Thomson ampling, Markov Decision Process Modeling - Bellman quation, Dynamic Programming Solutions - Value and olicy Iteration, Model free methods - Monte Carlo and emporal Difference Methods - Q-learning, Value unction Approximation - State Aggregation, Critic nly/Value Based Methods Methods - TD methods, -Learning, SARSA, Actor Only/Policy Based methods - einforce, Actor-Critic Methods - Policy Gradient, Deep L - DQN, A3C,
4	**Texts/References**	1.Richard S. Sutton and Andrew G. Barto, Introduction to Reinforcement Learning, 2nd Edition, MIT Press. 2017. 2.Dimitri Bertsekas and John G. Tsitsiklis, Neuro Dynamic Programming, Athena Scientific. 1996. 3.Bertsekas, Dimitri P. Dynamic Programming and Optimal Control. Vol. 1 and 2. 4th edition, 2012. 4.Algorithms for Reinforcement Learning, Csaba Szepesvári, Morgan & Claypool, 2009. 5.Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems, Sébastien Bubeck and Nicolò Cesa-Bianchi, Foundations and Trends in Machine Learning, Volume 5, Number 1, 2012.

1	**Title of the course** (L-T-P-C)	**Statistical Machine Learning** **(2-0-2-6)**
2	**Pre-requisite courses(s)**	Exposure to Probability Models, Stochastic Process and its Applications, Math for Data Science I and II
3	**Course content**	Bayesian Decision Making and Bayes Classifier, Parametric Estimation of Densities, General Linear Models, EM Algorithm, Discriminative Learning based Models, Dimensionality Reduction Techniques, Empirical risk minimization, Ensemble Methods - Bagging, Boosting, Clustering
4	**Texts/References**	1.R.O.Duda, P.E.Hart and D.G.Stork, Pattern Classification, John Wiley, 2001. 2.C.M.Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

1	**Title of the course** (L-T-P-C)	**Runtime Verification** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Not-applicable
3	**Course content**	1. Overview of Runtime Verification, and its comparison with other Formal Verification approaches. 2. Fundamentals: Propositional and First-Order Logic, Temporal Logics (Linear and Metric) 3. Propositional LTL and its variants: specification of properties, runtime verification strategies, expressibility, and monitorability. 4. First Order LTL and its variants: specification of properties, runtime verification strategies, expressibility, and monitorability. 5. Discussion of various state-of-the-art tools and case studies.
4	**Texts/References**	1.K. Havelund, D. Peled, "Runtime Verification: From Propositional to First Order Temporal Logic", Tutorial at International Conference on Runtime Verification, 2018. 2.Ezio Bartocci, Yliès Falcone. "Lectures on Runtime Verification". Springer, 2018. ISBN: 978-3-319-75632-5 3.Michael Huth, Mark Ryan, "Logic in Computer Science: Modelling and Reasoning about Systems", Cambridge University Press, 2004. ISBN: 978-0521543101 4.Research publications on Runtime Verification

1	**Title of the course** (L-T-P-C)	**Systems Bootcamp for ML** **(1-0-2-4)**
2	**Pre-requisite courses(s)**	Exposure to basics of computer programming.
3	**Course content**	Systems Programming lab focuses on programming principles and skills for building systems software. It also focuses on how to build and optimize Al/ML models from a systems perspective. ● Introduction to UNIX, shell programming, version control and management (SVN/Git) ● Introduction to Libraries (Boost, STL) ● Introduction to Profiling (Performance analysis at system level and application level), ML perf ● Introduction to various DL frameworks and DL Inference engines, Hardware backends ● Database fundamentals and programming ● GPU Programming (CUDA, OpenCL)/ ● Automatic Code Generation - TVM Stack ● Mobile-edge cloud computing (Computational offloading decisions) ● Programming assignments/projects will be given related to the above topics
4	**Texts/References**	1.Unix Man Pages for all unix tools, **Advanced Bash Scripting Guide from the Linux Documentation Project** (www.tldp.org) 2.Loeliger, Jon and MCullough, Matthew**, Version Control with Git: Powerful tools and techniques for collaborative software development**, O'Reilly, 2012. 3.Sommerville, Ian. **Engineering Software Products: An Introduction to Modern Software Engineering**, Pearson, May 2019. 4.Rodriguez Andres : **Deep Learning Systems: Algorithms, Compilers, and Processors** for Large- scale Production, Morgan and Claypool publishers, 2020

1	**Title of the course** (L-T-P-C)	**Topics in Design and Analysis of Algorithms** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Discrete Mathematics, Design and Analysis of algorithms, Data structures and Algorithms.
3	**Course content**	**Module 1:** Iterated Improvement Paradigms- Computational and Algorithmic Thinking, Matching Algorithms, Flow Algorithms (16 hours). **Module 2:** Approximation Algorithms- Greedy Approximation, Local Search, Linear Programming, Duality Techniques (16 hours) **Module 3:** Randomized Algorithms- Monte Carlo and Las Ve-gas types, Randomized Attrition, Randomized In- cremental Design, Sampling, Chernoff type bounds and High Confidence Analysis, Abundance of witness for Monte Carlo algorithms, Number theoretic Algorithms (16 hours).
4	**Texts/References**	1. [**OA]** James B. Orlin, Ravindra K. Ahuja, and Thomas L. Magnanti, “Network Flows”, Prentice Hall, 1993. 2. [**WS]** David P. Williamson and David B. Shmoys, “The Design of Approximation Algorithms”, CambridgeUniversity Press, 2011. 3. [**MR**] Rajeev Motwani and Prabhakar Raghavan, “Ran-domized Algorithms”, Cambridge University Press, 1995.

1	**Title of the course** (L-T-P-C)	**Advanced Algorithms** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Exposure to Discrete Mathematics, Design and Analysis of algorithms, Data structures and Algorithms
3	**Course content**	**Module 1:** Iterated Improvement Paradigms- Computational and Algorithmic Thinking, Matching Algorithms, Flow Algorithms (16 hours). **Module 2:** Approximation Algorithms- Greedy Approximation, Local Search, Linear Programming, Duality Techniques (16 hours) **Module 3:** Randomized Algorithms- Monte Carlo and Las Ve-gas types, Randomized Attrition, Randomized In- cremental Design, Sampling, Chernoff type bounds and High Confidence Analysis, Abundance of witness for Monte Carlo algorithms, Number theoretic Algorithms (16 hours).
4	**Texts/References**	1. [**OA]** James B. Orlin, Ravindra K. Ahuja, and Thomas L. Magnanti, "Network Flows", Prentice Hall, 1993. 2. **[WS]** David P. Williamson and David B. Shmoys, "The Design of Approximation Algorithms", CambridgeUniversity Press, 2011. 3. [**MR**] Rajeev Motwani and Prabhakar Raghavan, "Ran-domized Algorithms", Cambridge University Press, 1995.

1	**Title of the course** (L-T-P-C)	**Topics in Graph Theory** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Discrete Structures
3	**Course content**	Recap: fundamental concepts. Topics in factors, covering and packing, cuts, connectivity, coloring, planarity, perfect graphs, Ramsey theory, and random graphs.
4	**Texts/References**	Textbook: 1 . Introduction to Graph Theory (2nd Edition), Douglas B. West. Prentice Hall. References: 1.Algorithmic Graph Theory and Perfect Graphs (2nd Edition), Martin Charles Golumbic. Elsevier. 2.Graph Theory (Graduate Texts in Mathematics, 5th Edition), Reinhard Diestel. Springer.

1	**Title of the course** (L-T-P-C)	**Topics in Parameterized Algorithms and Complexity (3-0-0-6)**
2	**Pre-requisite courses(s)**	Data Structures and Algorithms, Design and Analysis of Algorithms
3	**Course content**	Introduction. Kernelization, Bounded Search Trees, Iterative Compression, Treewidth, Advanced kernelization algorithms. Lower bounds: Fixed-parameter intractability, lower bounds based on ETH, lower bounds for kernelization. Parameterized Algorithms, Kernelization, and Complexity of Graph Modification Problems
4	**Texts/References**	Textbook: Parameterized Algorithms, Marek Cygan, Fedor V. Fomin, Lukasz Kowalik. Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Sourabh. Springer. 2015 Reference: Parameterized Complexity, R. G. Downey, and M. R. Fellows. Springer Science and Business Media. 2012

1	**Title of the course** (L-T-P-C)	**Power Aware Computing (3-0-2-8)**
2	**Pre-requisite courses(s)**	Exposure to Computer Architecture, Operating Systems
3	**Course content**	Introduction to Power and Energy, Power consumption modeling and estimation, Dynamic power management and DVFS, Leakage reduction techniques, circuit-level and Micro-architecture techniques, Power states and ACPI support, Memory/cache power optimizations. Software level techniques, GPU power modeling and optimizations
4	**Texts/References**	1.S. Kaxiras, M. Martonosi, Computer Architecture Techniques for Power- Efficiency, Synthesis Lectures on Computer Architecture. Morgan &C laypool publishers 2. Siva G. Narendra, Anantha Chandrakasan 3.P. Leakage in Nanometer CMOS Technologies, Series on Integrated Circuits and Systems 4.Rakesh Chadha, J Bhasker an ASIC Low Power Primer: Analysis, Techniques and Specification

1	**Title of the course** (L-T-P-C)	**Dataflow Processor Architecture (Guided Study)** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Computer Architecture
3	**Course content**	The philosophy of dataflow; static and dynamic approaches; contrast with conventional out-of-order pipelines; understanding different granularities of operations, appreciating the performance and power possibilities; understanding the caveats; studying particular example architectures; analyzing the fundamental concept in the light of modem trends in the semiconductor industry; analyzing the fundamental concept in the context of particular application classes
4	**Texts/References**	Papers(list not exhaustive); 1 Exploring the potential of heterogeneous von Neumann/Dataflow execution models, Nowatzkietal., ISCA2015. 2 Dataflow Machine Architecture. AH Veen,ACM Computing Surveys.1986. 3 Dataflow Architecture: Are dataflow computers commercially viable?, Kavi et al.,IEEE Pontentials 1992. 4 Synchronous Dataflow Architectures for Network processors, Carlstrom et al., IEEE Micro,2004. 5 Dataflow architectures,Culler.Annual review of Computer Science 1986. 6 An architectural comparison of dataflow systems .Srini. dataflow Computing: Theory and Practice 1992. 7 The Machester Prototype Dataflow Computer.Gurd et al.ACM.1985. 8 Monsoon: an Explicit Token-store Architecture.Papadopolous et al. ACM SIGARCH 1990.