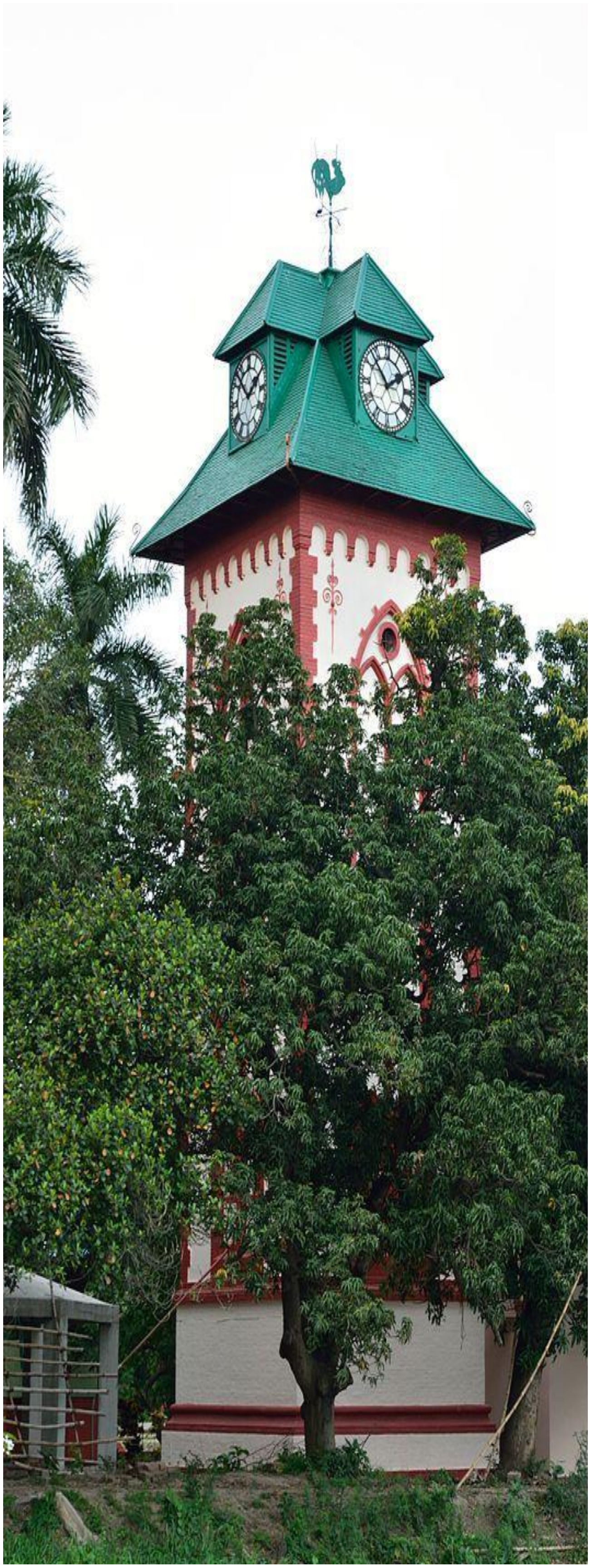

Department of Computer Science and Technology

Postgraduate Programmes

Course Structure and Syllabus

(Effective from 2025-26 admitting batch onwards)

Indian Institute of Engineering Science and Technology (IIEST), Shibpur

Botanic Garden, Howrah

Table of Contents

Course Structure

M.Tech in Computer Science and Engineering

First Semester									
Sl. No.	**Type**	**Course Name**	**Course code**	**Class Load/Week**			**Credit**	**Class load/ week**	**Marks**
				L	**T**	**P**			
1	PC	Advanced Algorithms	CS5101N	3	0	0	3	3	100
2	PC	Real Time Systems	CS5102N						
3	PC	High Performance Computing	CS5103N						
4	PSE	Machine Learning	CS5121N	3	0	0	3	3	100
		Information and Coding Theory	CS5122N						
		VLSI System Design	CS5123N						
		Digital Signal Processing	CS5124N						
		Cognitive Science	CS5125N						
		Cryptography and Network Security	CS5126N						
		Blockchain Technologies	CS5127N						
5	OE	Soft Computing	CS5161N	3	0	0	3	3	100
		Data Structures and Design of Algorithms	CS5162N						
		Introduction to AI and ML	CS5163N						
		Theory Sub-total		**15**	**0**	**0**	**15**	**15**	**500**
6	PC	Sessional on Advanced Algorithms	CS5171N	0	0	3	2	3	50
7	PC	Sessional on Real Time Systems	CS5172N						
8	PC	Sessional on High Performance Computing	CS5173N						
		Practical Sub-total		**0**	**0**	**9**	**6**	**9**	**150**
		First Semester Total		**15**	**0**	**9**	**21**	**24**	**650**

Second Semester	-								
Sl. No.	**Type**	**Course Name**	**Course code**	**Class Load/Week**			**Credit**	**Class load/ week**	**Marks**
				L	**T**	**P**			
1	PC	Mathematics for Computer Science	CS5201N	3	0	0	3	3	100
2	PC	Advanced Database Management Systems	CS5202N	3	0	0	3	3	100
3	PC	Advanced Networking	CS5203N	3	0	0	3	3	100
4	PSE	Embedded Systems for Secured Hardware Design	CS5221N	3	0	0	3	3	100
		Natural Language Processing	CS5222N						
		Computational Geometry	CS5223N						
		Internet of Things	CS5224N						
		Quantum Computing	CS5225N						
		Graph Algorithms	CS5226N						
		Cyber Security and Forensics	CS5227N						
		Deep Learning	CS5228N						
		In-Memory Computation	CS5229N						

		Reconfigurable Computing	CS5230N						
5	OE	Data mining and Knowledge Discovery	CS5261N	3	0	0	3	3	100
		Information Security and Cryptography	CS5262N						
		Database Management Systems	CS5263N						
		Computer Control of Industrial Processes	CS5264N						
		Mobile and Pervasive Computing	CS5265N						
		Theory Sub-total		**15**	**0**	**0**	**15**	**15**	**500**
6	P	M.Tech. project/Term-paper	CS5291N	0	0	3	2	3	50
7	O	Seminar/Viva Voce	CS5292N	0	0	3	2	3	50
		Practical Sub-total		**0**	**0**	**6**	**4**	**6**	**100**
		Second Semester Total		**15**	**0**	**6**	**19**	**21**	**600**

Third Semester	**-**								
Sl. No.	**Type**	**Course Name**	**Course code**	**Class Load/Week**			**Credit**	**Class load/ week**	**Marks**
				L	**T**	**P**			
1	**VAC**			3	0	0	3	3	100
2	P	M. Tech. Thesis	CS6191N				12	24	300
3	O	Progress Seminar and Viva-voce	CS6192N				6		100
4	I	Summer internship (6-8 weeks) evaluation					2		50
		Third Semester Total					**23**		**550**

Fourth Semester	**-**								
Sl. No.	**Type**	**Course Name**	**Course code**	**Class Load/Week**			**Credit**	**Class load/ week**	**Marks**
				L	**T**	**P**			
1	P	M. Tech. final thesis	CS6291N				22	30	400
2	O	Thesis Seminar and Viva-voce	CS6292N				8		200
		Fourth Semester Total					**30**		**600**

M.Tech in Computer Science and Engineering

Detailed Syllabus

1st Semester Courses Syllabi

M.Tech in Computer Science and Engineering

Course Code	CS5101N	Course Name	Advanced Algorithms	Course Category	PC1	L	T	P
						3	0	0

Pre-requisite Courses	*Discrete Structure, Data structure, Theory of Computation*	**Co-requisite Courses**	*None*	**Progressive Courses**	
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	To introduce advanced ideas in design of algorithms; To study the performance guarantees of algorithms; to understand complexity classes, To introduce methods for coping with NP-hard problems.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Design Paradigms:** Overview: Divide and conquer, Dynamic Programming, Greedy Algorithms, Graph search and traversal techniques, Backtracking, Branch and Bound.	4	To explain different problem solving strategy
2	**Network Flow, Matrix Operations:** Network flow – Augmenting paths, Ford-Fulkerson method, Maximum bipartite matching Matrix Operations – Strassen's algorithm, Solving system of linear equations, Inverting matrices.	5	To explain how design strategy come into play in the specified problems
3	**String Matching Problem**: Introduction to string-matching problem, Naïve algorithm, Rabin Karp, Knuth Morris Pratt, BoyerMoore algorithms and complexity analysis.	5	To explain how design strategy come into play in the specified problems
4	**Computational Geometry:** Line-segment properties, Sweep algorithms, Finding convex hull.	4	To introduce Geometric algorithm
5	**NP – Completeness, Approximation Algorithms:** Theory of NP- Hard and NP-Complete Problems: P, NP and NP-Complete complexity classes; A few NP-Completeness proofs; Other complexity classes. **Approximation Algorithms:** Performance bounds (ratio bound, relative error bound) of the problems, Approximation scheme, A few Approximation Algorithms.	8	To introduce the concept of NP completeness and explain proof strategy for NP-complete and eventually explain approximation algorithm for such type of problem
6	**Probabilistic Algorithms & Randomized Algorithms:** Numerical probabilistic algorithms, Las Vegas and Monte Carlo algorithms, Game-theoretic techniques, Applications on graph problems	6	To introduce probabilistic and randomized algorithm with few examples

7	**Parallel Algorithms:** Introduction, Models, speedup and efficiency, Some basic techniques, Examples from graph theory, sorting, Parallel sorting networks. Parallel algorithms and their parallel time and processor complexity.	8	To briefly explain parallel algorithm

Course Outcome	CO1: To comprehend and select algorithm design approaches in a problem specific manner. CO2: To develop sound theoretical understanding of advanced algorithms and practical problem solving skills using them CO3: To understand the necessary mathematical abstraction to solve problems. CO4: To come up with analysis of efficiency and proofs of correctness

Learning Resources	**References:** 1. Introduction to Algorithms, 3rd Edition,Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, MIT Press 2009, ISBN 978-0-262-03384-8. 2. Algorithm Design. Kleinberg, Jon, and Éva Tardos, Addison-Wesley, 2006.

Course Code	CS5102N	Course Name	Real Time Systems	Course Category	PC2	L	T	P
						3	0	0

Pre-requisite Courses	*(i)Operating Systems (ii)Data structures, and algorithms (iii) C Programming*	**Co-requisite Courses**		**Progressive Courses**	
Course Offering Department		**CST**		**Data Book / Codes/Standards**	

Course Objective	(i) To comprehend real-time concepts (ii) To get acquainted with different non-programmable and programmable approaches for implementing an Real Time System (RTS) (iii) To study the available architectural patterns for designing an RTS along with their associated issues and how to resolve them. (ii) To explore the role of Real Time Operating Systems in RTS with case studies

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction - Characteristics and Constraints, Hard Versus Soft Real-Time Systems	2	
2	Hardware for Real-Time Systems - Processor Architecture, Memory Technologies, Architectural Advancements, IO Interfacing, Microprocessor versus Microcontroller, Distributed Architectures	3	
3	Temporal Modeling and specification of real time systems: State diagram, finite automata model, petri-net, state chart and mode chart, Q-model, formal methods.	3	
4	Real-Time Operating System - Kernel Design Alternatives	5	
5	Real-time Scheduling (Uniprocessor scheduling): Basic concepts. Scheduling anomalies. Aperiodic task scheduling. Clock-based and priority-based scheduling of periodic tasks. Resource access protocols.	6	
6	Real-time Scheduling: Multiprocessor/Multicore scheduling	4	
7	Real-time Scheduling: Distributed scheduling	3	
8	Real-time synchronization: uniprocessor protocols	4	
9	Real-time synchronization: multiprocessor/multicore protocols	4	
10	Time/utility function real-time resource management	8	
11	Real-Time Operating System and Internet of Things	2	
12	Real time systems analysis. Safety and reliability. Fault tolerance techniques. Performance analysis. Execution time prediction.	3	
13	Real-Time Programming	4	

Course Outcome	(i) To identify the real-time nature of a system (ii) To analyze and specify a real-time system (RTS) (iii) Given the specification of an RTS, to choose the most appropriate approach for its design and implementation. (iv) To choose appropriate hardware and software platform and algorithms for implementation of an RTS (v) To choose and use from the available patterns for design and implementation of an RTS.

Learning Resources	**References /Books:** 1. Real-Time Systems by Jane Liu 2. Real-Time Systems Design Principles for Distributed Embedded Applications by H. Kopetz 3. Real Time Systems Design and Analysis (Tools for the Practitioner) by P. A. Laplante and S. J. Ovaska 4. The Concise handbook of Real-Time Systems (version 1.3) by TimeSys Corporation.

Course Code	CS5103N	Course Name	High Performance Computing	Course Category	PC3	L	T	P
						3	0	0

Pre-requisite Courses	Digital Logic Design, Computer Organization and Computer Architecture	Co-requisite Courses	*None*	Progressive Courses	Embedded Systems, Reconfigurable Computing
Course Offering Department		**CST**		**Data Book / Codes/Standards**	

Course Objective	
	1. To provide a comprehensive understanding of parallel computing models and architectures. 2. To introduce parallel programming paradigms such as CUDA and Verilog HDL. 3. To explore the design and implementation of scalable algorithms. 4. Highlight current trends and challenges in HPC

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Development of computer systems, von Neumann machine, performance measure, CISC and RISC processors, Parallel Processing Concepts, Memory hierarchy, Virtual memory system, CAM	6	Able to understand the fundamental concepts of parallel processing and its importance in modern computing
2	Pipelining, Pipeline hazards - structural, data and control hazards, minimizing hazard - stalls, data forwarding, dynamic scheduling, latency analysis, MAL Instruction-level parallelism, superscalar, super-pipelined and VLIW architectures Array processors, Vector processors, Centralized and distributed shared-memory architectures	6	Able to understand the concept of pipelining, instruction-level parallelism and cache coherence issues and solutions in parallel systems. Able to differentiate between centralized and distributed shared-memory architectures
3	Parallel computers: Flynn's taxonomy - SISD, SIMD – shared memory, interconnection network; MISD and MIMD machines, Parallel and scalable architectures; multiprocessor system interconnects –shared bus, multi-port, crossbar, multistage networks; vector processors; distributed shared-memory architectures; Cluster Computer Architecture Program flow mechanisms; dataflow processors	7	Able to classify computer architectures using Flynn's taxonomy, understand various parallel and scalable architectures, and explain multiprocessor interconnects.

4	Cache design: Cache write – write-back and write-through; Improving cache performance, reducing cache miss rate, miss penalty and hit time; cache coherence - data consistency, enforcing cache coherence, Snoopy protocol, Cache models - MESI, MOESI, Directory based protocol, Cache coherence in CMPs	8	Able to learn techniques to improve cache performance and gain knowledge of cache coherence issues, protocols like MESI, MOESI, and directory-based methods, and how coherence is maintained in Chip Multiprocessors.
5	Design issues in HPC: Load balancing, scheduling, synchronization and resource management; Operating systems for scalable HPC	4	Able to understand key design challenges in HPC systems, analyze and apply techniques for load balancing, scheduling, synchronization, and resource management.

6	Scalable storage systems: RAID, SSD cache, SAS, SAN HPC based on cluster, cloud, and grid computing	4	Will gain knowledge of scalable storage technologies such as RAID, SSD cache, SAS, and SAN, and understand their integration in high-performance computing (HPC) environments
7	Accelerated HPC: architecture, GPU, FPGA, Xeon Phi, Cell BE programming, Power-aware HPC Design Beta scale computing; big data processing, optics in HPC, quantum computers.	7	Able to understand accelerated HPC architectures and programming using GPU, FPGA along with concepts in power-aware design

Course Outcome	CO1: To be able to understand the fundamentals of parallel computing models and system architectures. CO2: To be able to analyze the performance of parallel programs using speedup, efficiency, and scalability metrics. CO3: To be able to apply parallel programming models to develop efficient code. CO4: To be able to evaluate trade-offs in memory usage, computation, and communication in high performance systems. CO5: To be able to design and implement scalable parallel algorithms for real-world scientific or engineering problems.

Learning Resources	**References:** 1. Georg Hager and Gerhard Wellein. Introduction to High Performance Computing for Scientists and Engineers (1st ed.). CRC Press, Chapman; Hall/CRC Computational Science, India 2. Vipin Kumar , Ananth Grama , Anshul Gupta , George Karypis. Introduction to Parallel Computing (2nd ed.). Pearson India . 3. John L. Hennessy and David A. Patterson. Computer Architecture: A Quantitative Approach (5th ed.). Morgan Kaufmann. 4. David B. Kirk and Wen-mei W. Hwu. Programming Massively Parallel Processors: A Hands-On Approach (2nd ed.). Morgan Kaufmann 5. Kai Hwang, Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill. 6. M. J. Flynn, Computer Architecture: Pipelined and Parallel Processor Design, Narosa Publishing House.

Course Code	CS5121N	Course Name	Machine Learning	Course Category	PSE1	L	T	P
						3	0	0

Pre-requisite Courses	Linear Algebra, Probability and Statistics, Programming for Data Science / Python Programming, Data Structures and Algorithms	Co-requisite Courses	Optimization Methods, Data Analytics and Visualization, Image Processing, Natural Language Processing	Progressive Courses	Advanced Machine Learning, Reinforcement Learning, Natural Language Processing, Computer Vision
Course Offering Department		**CST**		**Data Book / Codes/Standards**	

Course Objective	(i) The objective of this course is to equip postgraduate students with a solid foundation in machine learning by covering both theoretical concepts and practical implementations. (ii) Students will learn core topics such as supervised and unsupervised learning, optimization techniques, deep learning architectures, reinforcement learning, and ensemble methods. (iii) The course aims to develop students' ability to model, analyze, and solve real-world problems using data-driven approaches, while also building the mathematical and computational skills necessary to understand and implement advanced algorithms.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Mathematical Foundations:** Linear Algebra - Vector spaces, matrices, eigenvalues; Probability and Statistics - Bayes' theorem, distributions; Optimization Techniques - Gradient descent, convexity	3	i) Develop a strong understanding of linear algebra concepts. ii) Apply principles of probability and statistics for modeling uncertainty and making inferences. iii) Understand and implement fundamental optimization techniques. iv) Build the mathematical foundation required to analyze, design, and optimize machine learning algorithms effectively.
2	**Supervised Learning:** generative/discriminative learning, parametric/nonparametric learning, Naive Bayes Model, Logistic regression, Decision Trees, and support vector machines, Model Evaluation, Case study	5	i) Understand key concepts in supervised learning, including generative vs. discriminative models and parametric vs. nonparametric approaches. ii) Gain practical knowledge of core algorithms. Learn techniques for model evaluation and validation. iii) Apply supervised learning methods to case studies, developing the ability to solve practical classification and prediction problems.
3	**Unsupervised Learning:** clustering, Kohonen Network, SOFM, dimensionality reduction, kernel methods; Advanced discussion on clustering and Gaussian Mixture Models, Expectation Maximization	5	i) Understand the principles of unsupervised learning. Gain knowledge of neural network-based approaches. ii) Apply advanced clustering methods, including Gaussian Mixture Models and the Expectation-Maximization algorithm. iii) Explore kernel methods and their role in high-dimensional data analysis, enhancing the ability to uncover hidden patterns in unlabeled data.

4	**Optimization Techniques:** Convex vs. Non-convex optimization, Local vs. global optimization, Gradient-Based Optimization, Adaptive Optimization Algorithms, Constrained Optimization, **Regularization Techniques**	5	i) Understand the differences between convex and non-convex optimization. ii) Gain proficiency in gradient-based optimization methods and adaptive algorithms. iii) Apply constrained optimization techniques to handle problems with specific limitations or requirements. iv) Utilize regularization methods to improve model generalization and prevent overfitting in learning algorithms.
5	**Deep Learning Architecture:** Feedforward Neural Networks, Convolutional Neural Networks, Recurrent Neural Networks, Transformer Architectures, Autoencoders, Generative Adversarial Networks, Graph Neural Networks, Case study	10	i) Understand the core architectures of deep learning. ii) Gain knowledge of advanced models such as transformer architectures, autoencoders, generative adversarial networks (GANs), and graph neural networks (GNNs). iii) Apply deep learning models to real-world case studies
6	**Reinforcement learning:** Key elements, Types of RL, Exploration vs. Exploitation trade-off, Markov Decision Processes, Case study	4	i) Understand the fundamental concepts and key elements of reinforcement learning (RL). Explore different types of RL approaches. ii) Analyze the exploration vs. exploitation trade-off and its impact on learning efficiency and performance. iii) Apply Markov Decision Processes (MDPs) to model sequential decision-making problems, and implement RL techniques in real-world case studies.

7	**Ensemble methods:** Bagging, Boosting, Stacking, Voting Ensembles, Case study	4	i) Understand the principles and motivation behind ensemble learning. ii) Gain proficiency in key ensemble techniques, including bagging, boosting, stacking, and voting ensembles. iii) Analyze the strengths and trade-offs of different ensemble methods in various learning scenarios. iv) Apply ensemble techniques to real-world case studies, enhancing the ability to build and evaluate powerful predictive models.

Course Outcome	1. Develop a deep understanding of core machine learning concepts, including supervised and unsupervised learning, optimization, and model evaluation techniques. 2. Gain proficiency in implementing various machine learning models, such as decision trees, support vector machines, and deep learning architectures using tools like Python and popular libraries. 3. Analyze and apply optimization techniques, including gradient-based methods, regularization, and adaptive algorithms, to improve model performance. 4. Implement and evaluate advanced deep learning models, such as convolutional and recurrent neural networks, transformers, and autoencoders. 5. Gain expertise in reinforcement learning and ensemble methods, applying them to solve complex, real-world problems across domains like robotics, natural language processing, and computer vision. 6. Develop the ability to critically assess machine learning models, interpret results, and apply ethical considerations when deploying AI systems in various applications.

Learning Resources	**References /Books:** **Books:** 1. Hal Daumé III, A Course in Machine Learning 2. Chris Bishop, Pattern Recognition and Machine Learning 3. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Francis Bach, Deep Learning (Adaptive Computation and Machine Learning series) 4. Tom Mitchell, Machine Learning **Reference Books:** 1. Trevor Hastie, Robert Tibshirani and Jerome Friedman, Elements of Statistical Learning 2. Charu Aggarwal, Neural Networks and Deep Learning: A Textbook 3. Maxim Lapan, Deep Reinforcement Learning Hands-On 4. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning 5. Andreas C. Müller, Sarah Guido, Introduction to Machine Learning with Python 6. François Chollet, Deep Learning with Python

Course Code	CS5122N	**Course Name**	Information and Coding theory	**Course Category**	PSE1	L	T	P
						3	0	0

Pre-requisite Courses	Abstract Algebra, Probability and Statistics, Basic communication theory	**Co-requisite Courses**	*NIL*	**Progressive Courses**	*NIL*
Course Offering Department		***CST***		**Data Book / Codes/Standards**	*NIL*

Course Objective	to equip the postgraduate students with mathematical foundations of information theory and coding theory

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Revision of probability theory:** Bayesian theorem, conditional probability, joint probability, expectation and variance of random variables.	2	To refresh ideas needed for the course
2	**Concept of Information:** Zero memory source, Markov source, extensions of information source, source coding, channel coding, uncertainty, information entropy	3	Understand information source and metrics
3	**Source coding:** Shannon's first theorem, Shannon Fano coding scheme, Huffman coding scheme, Data compression, Kraft McMillan inequality	3	Ability to design source codes
4	**Channel coding:** Channel coding theorem, error at receiver, channel equivocation, mutual information, cascading of channels, Differential entropy, channel capacity theorem, Shannon Limit.	4	Ability to design channel codes
5	**Message transmission:** Concept of message rate, reliable transmission of information over an unreliable channel, Shannon–Hartley theorem	4	Understand principles of comm
6	**Introduction to algebraic coding theory:** Groups, fields, GF(p^m), vector space, matrices, representation of message and code in vector and polynomial form	3	Understand the theory behind coding
7	**Error Control coding:** Linear block codes, concept of generator matrix, parity check matrix, error correction techniques, syndrome based decoding, standard array based decoding	4	Ability to design and explore linear block codes
8	**Cyclic codes:** Definition, generator polynomial, parity check polynomial, systematic form, random error correction, burst error correction, multiple error correcting cyclic codes, generalized BCH codes, design distance	4	Ability to design cyclic codes and their usage
9	Ideas of non-binary codes, Convolutional codes, LDPC codes	5	Appreciate recent development
10	Applications of coding theory in data security and cryptography	4	Appreciate recent applications

Course Outcome	ability to present information in the form of channel admissible codes ability to mathematically model the communication channel ability to model channel noise for different channels ability to design error correcting codes for different channels ability to encrypt and decrypt information using coding theory concept

Learning Resources	**References/Books:** 1. Lin and Costello, Error control Coding 2. Moon, Error correction coding 3. Hamming, Coding and Information theory 4. Cover and Thomas, Elements of information theory

Course Code	CS5123N	**Course Name**	VLSI System Design	**Course Category**	PSE1	L	T	P
						3	0	0

Pre-requisite Courses	*Digital Logic, Basic Electronics*	**Co-requisite Courses**		**Progressive Courses**	
Course Offering Department		**CST**		**Data Book / Codes/Standards**	

Course Objective	To provide an understanding of VLSI design principles, tools, and methodologies, enabling students to design and analyze Very Large Scale Integration (VLSI) technology, enabling them to develop innovative solutions for various applications.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Introduction to VLSI System Design:** MOS Devices, Circuits and Fabrication, Design Principles and Characteristics of MOS Devices in Logic Circuits, Size and Complexity of Integrated Circuits, Feature Size, Impact of Shrinking, Clocking, Scaling, PLA Minimization and Folding, Inverters and Logic Gates, Design Rules and Layouts, Stick Diagram, Transistor Sizing, CMOS Downscaling, Process Technology, Nanotechnology Devices, VLSI / ULSI.	4	Students will gather basic knowledge of MOS-based VLSI design, scaling, layout, and emerging technologies.
2	**ASIC**: Introduction, Advantages, Examples, Classification, VLSI Design style and Design Flow, Economics of ASIC.	2	To know basics of ASICs, their types, design flow, and economic aspects in VLSI design.
3	**Simulation:** Types of Simulation, Logic Simulation, Circuit Simulation, Gate level Simulation, Switch Simulation.	2	Students will be familiar with various types of simulation techniques used in VLSI, including logic, circuit, gate-level, and switch-level simulations.
4	**VLSI Design Process:** System Specification, Functional Design, Logic Design, Circuit Design, Physical Design, Verification, Fabrication and Packaging.	3	To know complete VLSI design process from system specification to fabrication and packaging.

5	**Design Styles:** Custom Design, Standard-Cell Design, Gate-Array Design, FPGA (Architecture and Physical Design) and MCMs.	3	Students will be familiar with different VLSI design styles including custom, standard-cell, gate-array, FPGA, and MCMs.
6	**Physical Design Issues:** Partitioning – Iterative and Constructive Algorithms. Floorplanning – Rectangular Dual and Hierarchical Method. Placement – Force-directed, Simulated Annealing and Genetic Algorithms. Routing – Area Routing, Channel Routing, Switchbox Routing, Performance driven Placement and Routing, Clock Routing. Layout Compaction. Design Rule Verification and Complexity Issues.	8	Students will know the key steps of physical design in VLSI, including partitioning, floorplanning, placement, routing, layout compaction, and design rule verification.
7	**VHDL:** Introduction, Composite Data Type, Processes, Subprograms, Behavioral & Structural VHDL, Packages, Libraries, VHDL in Simulation – Test Bench, VHDL for Combinational, Sequential logic & FMS.	4	To know VHDL concepts and use them to model, simulate, and test combinational, sequential circuits, and finite state machines.

8	Case Studies: Arithmetic Logic Unit, Digital Filter VHDL for testing.	3	Students will be able to apply VHDL to design and test real-world digital systems like an Arithmetic Logic Unit and a Digital Filter.
9	Idea of System Level Design using VHDL / Verilog / MATLAB, System and FPGA Synthesis of the systems.	3	Students will get an overview of system-level design and synthesis using VHDL / Verilog / MATLAB for FPGA implementation.
10	Logic Synthesis: Design Methodology, PLA Based synthesis, Two and Multilevel combinational circuit – OBDD, Synthesis, Delay, Testability.	3	Students will learn logic synthesis methodologies, including PLA-based and multilevel synthesis, with focus on OBDD, delay, and testability.

11	SOC Design: ASIC and SOC, IP-Reuse & Integration, Design Factors, Design Flow, Verification, Low Power Design – Algorithm, Architecture, Optimization.	3	Students will gather SoC design concepts, including IP reuse, design flow, verification, and low-power design techniques at algorithm and architecture leve.
12	Application: DSP and Audio-Video Processor.	2	Students will know VLSI applications in designing DSP and audio-video processor systems.
13	Design for Testability: Fault Types and Models, Ad Hoc Testable Design Techniques, Scan – based Techniques, Built-In Self Test Techniques	2	Students will gather design-for-testability concepts, including fault models, scan-based methods, and built-in self-test techniques.

Course Outcome	CO1: Understand the fundamentals of VLSI design methodologies, design hierarchy, and fabrication process. CO2: Analyze and design combinational and sequential digital circuits using CMOS logic styles. CO3: Develop and simulate digital systems using Hardware Description Languages such as Verilog or VHDL. CO4: Evaluate power, area, and delay of digital VLSI circuits to optimize performance. CO5: Understand and apply concepts of physical design, timing analysis, and testing in VLSI systems.

Learning Resources	**Refernces:** 1. An Introduction to Physical Design – Surrafzadeh, Wong, TMH Inc. 2. Algorithms for VLSI Physical Design Automation – Naveed Sherwani – Kellwer Academic Publisher. 3. Application Specific Integrated Circuit – by Sebastian Smith – Pearson Education Asia. 4. VHDL & FPLD in Digital System Design – by Salcic Zoran – Kellwar Academic Publisher. 5. Logic Synthesis – S. Devadas, A. Ghosh, K. Kellwar, McGrawHill Inc. 6. Logic Design Theory – N. N. Biswas, PHI. 7. FPGA Architecture and CAD tools – by V. Metz & J. Rose, Addison Weseley. 8. Reuse Methodology Manual for System on Chip Designs – Kellwar Academic Publishers. 9. Surviving the SOC Revolution – A Guide to Platform based Design – H. Chang, L. Cooke, M. Hunt, G. Martin, A. McNelly and L. Todd, Norwell, MA: Kellwar Academic Publisher.

Course Code	CS5124N	Course Name	Digital Signal Processing	Course Category	PSE1	L	T	P
						3	0	0

Pre-requisite Courses	*Mathematics, Basic electrical science, Basic communication theory*	**Co-requisite Courses**		**Progressive Courses**	*Image and video processing*
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	To familiarize the students with digital signal processing - basic mathematics, tools available for signal processing, hardware and software implementations.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Introduction:** Review of discrete-time signal and system analysis; Advantages and typical applications of DSP	2	Familiarize with the subject
2	**Sampling and Quantization:** Sampling and discrete-time processing of continuous time signals, Sampling of low-pass and band-pass signals; Uniform and non-uniform quantization, Lloyd-Max algorithm, Log-companding, A-law, μ-law; Adaptive quantization and prediction	6	Understand sampling techniques
3	**Orthogonal transforms:** Properties and applications of DFT, implementing linear time invariant systems using DFT, circular convolution, linear convolution using DFT; Fast Fourier Transform, FFT algorithms: Decimation in time, decimation in frequency; Goertzel algorithm; Application of transform in speech, audio, image and video coding, Karhunen-Loeve Transform, DCT, JPEG and MPEG coding standards	12	Understand different transform domains
4	**Digital Filter design techniques:** IIR and FIR filters, filter design specifications; Design of digital IIR filters: Impulse invariant, and bilinear transformation techniques for Butterworth and Chebyshev filters; **Design of FIR filters:** Windowing, frequency sampling filter design, optimum approximations of FIR filters **Digital Signal Processor architecture**	8	Ability to design digital filters, both finite and infinite response, and appreciate the processor architecture
5	**Multi-rate Signal Processing:** Fundamentals of multirate systems, Decimation and interpolation, application of Multirate DSP in sampling rate conversion; Filter banks; Polyphase structures; Quadrature-mirror filter bank; Wavelet transform and its relation to multi-rate filter banks; applications to speech and audio coding.	10	Understand the concept of handling signals with multiple rates

6	**Basic concept of Adaptive Digital Signal Processing:** Adaptive Wiener filter and LMS algorithm; Applications of adaptive filtering to echo cancellation and equalization	4	Appreciate techniques of of adaptive signal processing

Course Outcome	ability to formulate signal processing problem in digital domain ability to design different signal processing algorithms ability to interface DSP hardware with a given system ability to program the DSP hardware ability to innovate adaptive signal processing algorithms

Learning Resources	**References:** 1. Proakis, J.G. and Manolakis, D.G., "Digital Signal Processing: Principles, Algorithm and Applications", 4th Ed., Pearson Education. 2. Ifeachor, E.C. and Jervis, B.W., "Digital Signal Processing: A Practical Approach", 2nd Ed., Pearson Education. 3. Mitra, S.K., "Digital Signal Processing-A Computer Based Appraoach", 3rd Ed., Tata McGraw-Hil 4. Oppenheim, A.V. and Schafer, R.W. with Buck, J.R., "Discrete TimeSignal Processing", 2nd Ed., Prentice-Hall of India.

Course Code	CS5125N	Course Name	Cognitive Science	Course Category	PSE1	L	T	P
						3	0	0

Pre-requisite Courses	Mathematics, Statistics, Data Analysis and Programming	Co-requisite Courses		Progressive Courses	
Course Offering Department		**CST**		**Data Book / Codes/Standards**	

Course Objective	(i) Interdisciplinary subject aims to integrate knowledge from various fields like psychology, computer science, linguistics, and neuroscience to understand the fundamental principles of cognition.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Origin and History of Cognitive Science, Role of Philosophy, Psychology and Neuroscience, Relation between Logic and the Sciences, Science of Information Processing	4	(i) Understanding basic principles of cognitive science and related fields of processing information
2	Computational Models of Intelligence. Essential Features of Computational Models.	5	(i) Awareness to consciousness, language, and decision making
3	Knowledge Representations: Semantic Nets, Frames, Scripts.	3	(i) Formalize knowledge representation methods by analysing the problem
4	Inference Models of Cognition Generative Models – Conditioning – Causal and statistical dependence – Conditional dependence – Data Analysis – Algorithms for Inference.	10	Overview of Learning, Perception, and Inference generation
5	Learning Models of Cognition Learning as Conditional Inference – Learning with a Language of Thought – Hierarchical Models-Learning (Deep) Continuous Functions – Mixture Models..	10	Knowledge to develop mixture models
6	Simple Word Vector Representations: Language Models, Softmax, Single Layer Networks.	6	Understanding different language models
7	Applications: Language Modeling. Machine Translation. Sentence Classification. Neuropsychology for Computer Scientists.	4	Students should addresses real-world problem solving methods by research

Course Outcome	CO1: Understanding basic principles of cognitive processes, CO2: Develop skills in analyzing, interpreting, and assessing the empirical data CO3: Developing research techniques that contribute to **cognitive science**. CO4: Culminates to building and launching a working AI application that addresses real-world problem solving methods.

Learning Resources	1. Cognitive Science - An Introduction to the Science of the Mind by Jos Luis Bermdez 2. Cognitive science by Jose Luis Bermudez, Second edition 3. How the mind comes into being by Martin V. Butz & Esther F. Kutter, First edition 4. Conscious mind, resonant brain by Stephen Grossberg

Course Code	CS5126N	Course Name	Cryptography and Network Security	Course Category	PSE1	L	T	P
						3	0	0

Pre-requisite Courses	*Number Theory, Modular Arithmetic, Set Theory, Computer Networks.*	**Co-requisite Courses**		**Progressive Courses**	
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	To provide knowledge of cryptographic techniques and network security mechanisms, enabling students to design, analyze, and implement secure communication systems and protect information infrastructure against evolving cyber threats.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Attacks on Computers & Computer Security:** Introduction to Network Security, Motivation, Need for Security, Security approaches, Principles of Security, Types of attack, Threats in Networks, Network Security Controls.	2	Foundational understanding of network security, types of attacks and security principles.
2	**Cryptography: Basic Concepts & Techniques:** Introduction, Plaintext & Cipher text, Substitution Techniques, Transposition Techniques, Encryption & Decryption, Symmetric & Asymmetric key Cryptography, Key Range & Key Size, Mathematical foundations of cryptography.	5	Students will learn basic concepts of cryptography, including encryption techniques, key types, and related mathematical foundations.
3	**Symmetric Key Algorithm:** Introduction, Algorithm types & Modes, Overview of Symmetric Key Cryptography, Diffie-Hellman Key exchange algorithm and its analysis, Fiestel Cipher, Simplified-Data Encryption, Standard (S-DES) algorithm, DES algorithm, Variety of DES, Advanced, Encryption Standard (AES).	7	Students will be familiar with symmetric encryption, key exchange, and block ciphers like DES and AES.
4	**Asymmetric Key Algorithm:** Introduction, Overview of Asymmetric key Cryptography, RSA algorithm, Knapsack Cipher, Elliptic curve cryptography, Message Authentication Codes and Digital Signature, Symmetric & Asymmetric key Cryptography together.	5	Students will be familiar with asymmetric encryption, key algorithms like RSA, and basics of digital signatures and hybrid cryptographic systems.

5	**Message Digests and Hash Functions:** Basic concepts of Message Digest, MD 5 algorithm, Basic concepts of Hash Functions, SHA-1 algorithm.	4	Students will acquire the idea of message digests, algorithms like MD5 and SHA-1.
6	**Key Management and Certification Authority:** Digital Certificate, Certification Authority, Registration Authority, Steps involved in digital certificate creation, revocation, etc.	3	Students will be familiar with Digital certificates, roles of certification and registration authorities, and the process of certificate creation.
7	**IP Security:** Overview of IP Security (IPSec) and Virtual Private Networks (VPN), IP Security Architecture, Modes of Operation, Security Associations (SA), Authentication Header (AH), Encapsulating Security Payload (ESP), Internet Key Exchange.	5	Students will know the fundamentals of IP security, including IPSec architecture, VPNs, security protocols like AH and ESP.
8	**Web Security:** Web Security Threats, Web Traffic Security Approaches, Overview of Secure Socket Layer (SSL) and Transport Layer Security (TLS), Overview of Secure Electronic Transaction (SET). Distributed system security - Kerberos Motivation, Kerberos Version 4 and 5, PGP – Notation and operational description.	5	Students will learn Web security threats and protocols like SSL, TLS, SET, Kerberos, and PGP used for secure communication and authentication.

9	**Network Defence:** Firewall Characteristics, Types of Firewalls, Comparison of Firewall Types, overview of Firewall Configuration	2	Students will be familiar with Firewall types, their characteristics, and basic configuration for network defense.
10	**Wireless Security:** Security threats to wireless network – rogue access point/Ad-hoc networks, Denial-of-service, configuration problems (misconfigurations/incomplete configurations), passive capturing. Protocols – WEP, WPA, WPA2.	4	Students will understand Wireless network threats and security protocols like WEP, WPA, and WPA2.

Course Outcome	CO1: Understand and explain cryptographic concepts and their applications. CO2: Analyze and evaluate symmetric and asymmetric encryption algorithms. CO3: Design and implement secure communication protocols. CO4: Assess network vulnerabilities and propose appropriate countermeasures. CO5: Apply cryptographic techniques to real-world security problems. CO6: Conduct research and development in the field of information and network security.

Learning Resources	**References / Books:** 1. “Cryptography and Network Security”, William Stallings, 2nd Edition, Pearson Education Asia. 2. “Network Security private communication in a public world”, C. Kaufman, R. Perlman and M. Speciner, Pearson. 3. “Cryptography & Network Security”, Atul Kahate, TMH. 4. “Network Security Essentials: Applications and Standards”, by William Stallings, Pearson. 5. “Designing Network Security”, Merike Kaeo, 2nd Edition, Pearson Books. 6. “Practical Unix & Internet Security”, Simson Garfinkel, Gene Spafford, Alan Schwartz, 3rd Edition, Oreilly. 7. Network Security : Kaufman , Perlman, Speciner, Pearson Education.

Course Code	CS5127N	Course Name	Blockchain Technologies	Course Category	PSE1	L	T	P
						3	0	0

Pre-requisite Courses	*(i) Data Structures and Algorithms (ii) Cryptography and Network Security*	**Co-requisite Courses**	*(i) Distributed Systems (ii) Advanced Networking*	**Progressive Courses**	*(i) Federated Learning (ii) Web 3.0-based Decentralized Systems*
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	1. To understand basic cryptographic primitives used in blockchain. 2. To analyze the various versions of blockchain based on usecases. 3. To provide conceptual understanding of how blockchain technology can be used to secure and improve business processes. 4. To familiarize with the technology stack essential for implementation of blockchain-based solutions

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Introduction:** Overview of Blockchain: Definition of blockchain, Advantages of blockchain, Evolution of blockchain, Types of blockchain, Differences between public, private, and consortium blockchains.	6	-Understand the fundamental concept and definition of blockchain technology. -Explain the advantages and key features that make blockchain a transformative technology. -Trace the evolution and development stages of blockchain systems. -Differentiate between various types of blockchains, based on structure, access, and governance.

2	**Public Blockchain and Cryptocurrency:** Block Structure, Block-propagation, Mining, Mining pool, Hash puzzles, Miner, Mining function, and its difficulty, Proof of Work (PoW), and alternatives. Application of public blockchain, Bitcoin, Bitcoin Script, transaction process, coin generation, Double Spending, Handle Double Spending using Blockchain, mining pool reward distribution.	12	-Understand the internal structure and propagation of blocks in a public blockchain, along with the roles of miners and mining pools. -Explain the concept of Proof of Work (PoW), mining functions, hash puzzles, difficulty adjustment, and alternative consensus mechanisms. -Describe the Bitcoin architecture, including transaction processing, Bitcoin Script, and coin generation mechanisms. -Analyze the double-spending problem and evaluate how blockchain technology prevents it.

			-Evaluate mining pool reward distribution strategies and applications of public blockchains in decentralized systems.

3	**Private Blockchain and Smart Contract:** Smart Contract, Smart contract-based transactions, Distributed consensus algorithms (Raft, PAXOS, BFT, p-BFT), Components of a private blockchain. Hyperledger Fabric - Architecture, Details of the major components (like peer nodes, orderer service, channel, etc.) Transaction flow, Endorsement policy.	12	-Understand the concept of smart contracts and their role in automating blockchain-based transactions. -Explain key distributed consensus algorithms such as Raft, PAXOS, BFT, and p-BFT used in private blockchain systems. -Describe the essential components and architecture of private blockchains. -Analyze the architecture of Hyperledger Fabric, including peer nodes, orderer service, channels, and their roles. -Evaluate the transaction flow and endorsement policy in Hyperledger

			Fabric for secure and efficient operation.
4	**Overview on Blockchain Implementations and Case Studies**: Blockchain implementations - Ethereum, IOTA, Hyperledger Sawtooth, Stellar, etc. Usage of blockchain technologies in various enterprise applications: e-Voting, Know Your Customer (KYC), Supply chain management, Asset Records, Trade.	12	-Explore and compare various blockchain platforms in terms of architecture and use cases. -Understand and analyze the application of blockchain technologies in real-world enterprise use cases.

Course Outcome	CO1: Understand the blockchain technology. CO2: Develop blockchain based solutions and write smart contract using Hyperledger Fabric and Ethereum frameworks. CO3: Integrate ideas from various domains and implement them using blockchain technology in different perspectives.

Learning Resources	1. Melanie Swan, “Block Chain: Blueprint for a New Economy”, O’Reilly, 2015 2. Josh Thompsons, “Block Chain: The Block Chain for Beginners- Guide to Block chain Technology and Leveraging Block Chain Programming” 3. Daniel Drescher, “Block Chain Basics”, Apress; 1stedition, 2017 4. Anshul Kaushik, “Block Chain and Crypto Currencies”, Khanna Publishing House, Delhi. 5. Imran Bashir, “Mastering Block Chain: Distributed Ledger Technology, Decentralization and Smart Contracts Explained”, Packt Publishing 6. Ritesh Modi, “Solidity Programming Essentials: A Beginner’s Guide to Build Smart Contracts for Ethereum and Block Chain”, Packt Publishing 7. Salman Baset, Luc Desrosiers, Nitin Gaur, Petr Novotny, Anthony O’Dowd, Venkatraman Ramakrishna, “Hands-On Block Chain with Hyperledger: Building Decentralized Applications with Hyperledger Fabric and Composer”, Import, 2018

Course Code	CS5161N	Course Name	Soft Computing	Course Category	OE1	L	T	P
						3	0	0

Pre-requisite Courses	Mathematical background, Concept of Algorithm, Programming skill and critical thinking to solving problems.	**Co-requisite Courses**		**Progressive Courses**	
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	(i) The limitations of traditional, classical computing methods in solving real-world problems (ii) Introducing the concepts of Soft Computing techniques to solve real world problems.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Overview:** Goal of the subject, Definition of soft computing, How it differs from hard computing, Characteristics of soft computing, Components of soft computing.	3	Understanding when and what are the requirements of applying soft computing methods
2	**Fuzzy Sets and Systems:** Overview of Classical Crisp Sets, Crisp set versus Fuzzy set, Membership functions, Fuzzy rule generation, Operations on fuzzy sets, Properties of fuzzy sets, Fuzzy Relations, Fuzzy Measures, Fuzzy Arithmetic, Composition operations, Fuzzy Numbers, Linguistic Variables, Arithmetic Operations on Intervals & Numbers, Lattice of Fuzzy Numbers, Fuzzy Equations, Fuzzy extension principle, Approximate Reasoning, Fuzzy implication functions, Fuzzy Inference Systems, Type-2 fuzzy sets, Applications, Fuzzy Logic versus Classical Logic	10	(i) Idea about fundamentals of fuzzy sets, logic, etc. in solving engineering problems. (ii) Introducing concepts of modeling in systems using fuzzy based approaches. (iii) Understanding how to use human reasoning power in solving complex problems.
3	**Artificial Neural Networks:** History, overview of biological Neuro-system, Definition and Properties of Artificial Neural Network (ANN),Applications of Artificial Neural Networks, Mc-Pitts Model, ANN architecture, Learning Paradigms-Supervised, Unsupervised and reinforcement Learning, Hebbian learning, Perceptions learning rule, Delta learning, Back Propagation Algorithm, Multilayer Perceptron Model,Competitive learning networks, Kohonen self organizing networks, Hopfield Networks, Associative Memories, The boltzman machine: Working principles and Applications, Radial Basis function network, Recurrent neural network, current topics: Deep Learning, Applications	10	(i) Equip students with the fundamental understanding of ANNs, their architectures, training algorithms and applications. (ii) How to apply these concepts to solve real-world problems, including classification, regression, and pattern recognition.

4	**Optimization methods** Genetic Algorithms: Basic concepts of genetic algorithms, Traditional search method vs. Genetic Algorithm, How it works and where applicable, Encoding, Fitness function, Selecting, crossover, mutation, schema analysis, analysis of selection algorithms; convergence, genetic modeling, Applications, problem solving: Graph coloring, TSP,etc. Particle Swarm Optimization:Basic Concepts, Local Best, Global Best, Velocity Updation, Position Updation, Variant of PSO, Applications. Differential Evaluation: Basic Concept, Initialization of vectors, Target Vector, Donor Vector, Selection, Mutation, Crossover, Control Parameters, Applications Simulated Annealing, Ant colony optimization,Tabu Search, and current topics.	10	Understanding optimization and search problems using biologically inspired operators
5	**Rough Set Theory:** Decision Systems, Indiscernibility Relation, Inconsistency, Lower and Upper Approximation, Rough Set, Attribute dependency, Attribute Reduction, Discernibility matrix, Reduct and Core, Applications of Rough Set Theory: dimensionality Reduction, Feature Selection, Classifier.	5	(i) Enabling students to understand the fundamental concepts, principles, and applications of Rough Set Theory. (ii) Students should be able to apply these concepts to analyze data, identify patterns, and make informed decisions.

6	**Hybrid Systems:** Integration of Artificial neural networks, Fuzzy logic,Rough set Theory and Evolutionary Algorithms, Applications	4	(i) Understanding merits/demerits of each system and how to use synergistic model (ii) Equip with very specific nature to solve problems just like the human brain works.

Course Outcome	CO1: Students should understand which problems are not solved using conventional approaches. CO2: Characterizing the real world problem solving in presence of Uncertain, imprecise, ambiguous and incomplete knowledge. CO3: The concepts of fuzzy logic, Neural Network, Evolutionary Algorithms and other soft computing techniques and their role in solving real world problems. CO4: Creating Intelligent systems in solving problems using human centric reasoning, learning and bio-inspired computing.

Learning Resources	**References /Books:** 1. “Fuzzy Sets & Fuzzy Logic”, G.J. Klir & B. Yuan, PHI. 2. “Neural networks: a comprehensive foundation”, Haykin, Pearson. 3. “Neuro-fuzzy Systems”, Chin-teng-lin, C.S. Lee. 4. “Genetic Algorithms”, Goldberg, Pearson.

Course Code	CS5163N	Course Name	Data Structures and Algorithms	Course Category	OE1	L	T	P
						3	0	0

Pre-requisite Courses	Mathematics, Programming in C	**Co-requisite Courses**		**Progressive Courses**	
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	i) Understand and explain the fundamental concepts of data structures and their role in algorithm development and software design. ii) Analyze time and space complexity of algorithms using Big-O notation to evaluate performance. iii) Implement and apply linear data structures such as arrays, linked lists, stacks, and queues in problem-solving. iv) Understand and use non-linear data structures like trees and graphs, including traversal and searching techniques. v) Select and design appropriate data structures for various computational problems to improve efficiency and scalability.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Data structures:** Abstract data types (ADTs), vector, list, stack, queue, priority queue, trees, graph etc.	2	Demonstrate proficiency in implementing and analyzing abstract data types such as vectors, lists, stacks, queues, priority queues, trees, and graphs to solve complex computational problems.
2	**Algorithms:** Algorithm design, Algorithms and programs, basic idea of pseudo-code. Algorithm efficiency and analysis, time and space analysis of algorithms – order notations.	4	Apply algorithm design principles, express solutions using pseudo-code, and analyze algorithm efficiency using time and space complexity with order notation.

3	**Linear Data Structures** - Array and Linked list: Array : Different representations – row major, column major. Sparse matrix - its implementation and usage. Array representation of polynomials.Linked List: Singly linked list, circular linked list, doubly linked list, linked list representation of polynomial and applications.	6	Develop and implement linear data structures such as arrays and linked lists, including their specialized representations and applications in polynomial and sparse matrix computations.
4	**Linear Data Structures - Stack and Queue:** Stack: Stack and its implementations using array and linked list, applications. Queue: circular queue, dequeue. Implementation of queue- both linear and circular using array and linked list, applications.	5	Design and implement stacks and various types of queues using arrays and linked lists, and apply them to solve real-world computational problems.

5	**Recursion:** Principles of recursion – use of stack, differences between recursion and iteration, tail recursion. Applications - The Tower of Hanoi, Eight Queens Puzzle.	3	Analyze and implement recursive algorithms using stack-based execution models, distinguish recursion from iteration, and apply recursion to solve classic problems like Tower of Hanoi.
6	**Nonlinear Data structure - Tree:** Basic terminologies, forest, tree representation using array and linked list. Binary trees - binary tree traversal, threaded binary tree, expression tree. Binary search tree, Height balanced binary tree (AVL tree) B- Trees.	6	Construct and manipulate various tree structures including binary, threaded, expression, AVL, and B-trees, using array and linked list representations, and perform efficient tree traversals and operations.

7	**Non-linear Data Structure – Graph:** Graph definitions and concepts, Graph representations – adjacency matrix, adjacency list. Graph traversal – Depth-first search (DFS), Breadth-first, search (BFS), applications.	5	Implement graph data structures using adjacency matrices and lists, perform DFS and BFS traversals, and apply graph algorithms to solve practical problems.
8	**Searching and Sorting Algorithms:** Sorting: Bubble sort and its optimizations, insertion sort, shell sort, selection sort, merge sort, quick sort, heap sort, radix sort. **Searching:** Sequential search, binary search, interpolation search.	6	Evaluate and implement various searching and sorting algorithms, analyze their computational complexity, and apply them to organize and retrieve data efficiently.
9	**Hashing:** Hashing functions, collision resolution techniques.	3	Design efficient hashing schemes using appropriate hash functions and collision resolution techniques for effective data storage and retrieval.

Course Outcome	CO1: Ability to select and design data structures and algorithms that are appropriate for a given problem. CO2: Ability to gauge how the choice of data structures and algorithm design methods impacts the performance of programs. CO3: Ability to analyze algorithms and to determine algorithm correctness and complexities. CO4: Skill to identify the scope for improving the performance (in terms of algorithmic betterment) of a given application. CO5: Ability to choose an appropriate algorithm out of different alternative algorithms for a given problem.

Learning Resources	**References:** 1. M. A. Weiss, "Data Structures and Algorithm Analysis in C", 2nd Ed, Pearson Edu. Asia. 2. Y. Langsam, M. J. Augenstein and A. M. Tenenbaum, "Data Structures using C", Pearson Education Asia 3. Richard F. Gilberg, Behrouz A. Forouzan, "Data Structures – A Pseudocode Approach with C", Thomson Brooks / COLE. 4. Aho, J. E. Hopcroft and J. D. Ullman, Data Structures and Algorithms, Pearson Edu. Asia.

Course Code	CS5163N	Course Name	Introduction to AI and ML	Course Category	OE1	L	T	P
						3	0	0

Pre-requisite Courses		Co-requisite Courses		Progressive Courses	
Course Offering Department		***CST***		Data Book / Codes/Standards	

Course Objective	(i) Students should able to know about the history of AI and basic principles of AI (ii) Student also able to know some basic algorithms of ML (iii) Students should understand the different applications of AI/ML.

Module	Syllabus	Duration (class-hour)	Module Outcome
	Introduction: Definition of Artificial Intelligence, Alan Turing and cracking enigma, mark 1 machines, 1956-the birth of the term AI, AI winter of 70's, expert systems of 1980s, skipped journey of present day AI. Distinction between terms AI, Pattern recognition, and Machine Learning, Societal implication of AI	2	(i) Idea about basics of AI & ML (ii) What are the different subfields of AI
	Problem Solving with AI - State Space Search, Expert System. Problem Representations: Problem Representation in State Space, Production System – Components, Advantages, Applications. Uninformed Searches: BFS, DFS, Iterative Deepening (ID) Search with Space and Time Complexities. Informed Searches: Heuristic Functions, Hill Climbing Search, Best First Search, A* Algorithm, Admissibility of A* Algorithm, IDA* Algorithm, Problem Reduction and AO* Algorithm, Means-Ends Analysis. Adversarial Search: Games, Evaluation Function, The Min-Max Algorithm, Alpha-Beta Pruning.	11 (2+3+4+2)	(i) Understanding problem representation methods (ii) Knowledge acquired about different searching algorithms and concepts of heuristic search. (iii) Characteristics of Game search algorithms

	Knowledge Representation and Expert Systems: Propositional logic, FOL, Modus ponen, modus tollens, Semantic network, Frames, slots Rule Based Expert System: Rules as Knowledge Representation Technique, Architecture of Rule Based Expert System, Example, Forward Chaining and Backward Chaining, Advantages and Disadvantages, Uncertainty Management in Rule Based Expert Systems, Certainty Factors Theory, Comparisons of Bayesian Reasoning and Certainty Factors. Frame Based Expert System: Frames as Knowledge Representation Technique, Inheritance in Frame Based System, Methods and Demons, Interaction of Frames and Rules.	6 (2+4)	(i) Understanding different knowledge representation methods. (ii) How to managing uncertainty in real life problem solving
	Introduction of Machine Learning, Mathematical background. Types of Machine Learning	2	(i) Developing mathematical foundation to understand ML algorithms (ii) Learn how to improve model performance through feature processing
	Supervised learning: Regression Models, Support vector machines, Generative/discriminative learning, parametric/nonparametric learning, Multilayer Perceptron neural models, Gradient descent, Backpropagation, Batch processing, Learning rate, Cross validation, Overfitting, Regularization, Radial Basis function neural network, Principal component analysis, Confusion Matrix	8	(i) Different models of ML (ii) Learn how to evaluate and improve model performance (iii) Acquiring Knowledge about Foundation of Artificial Neural network

	Unsupervised learning: clustering, Kohonen Network, SOFM, dimensionality reduction, kernel methods; Advanced discussion on clustering and Gaussian Mixture Models, Expectation Maximization	4	Machine Learning Algorithms for clustering
	Reinforcement learning: Mathematical Formulation, Markov decision process, Policy based, model based, Value based learning Q-learning	5	Acquiring Knowledge about Foundation of RL
	Introduction to Deep Learning Models, Deep Q Network	4	Hybridization of different models

Course Outcome	CO1: To learn what AI is, what can (and can't) be done with AI, CO2: Ideas of a problem that could be solved using AI methods, CO3: Know the basic understanding about data and its analysis to develop predictive models. CO4: Identifying machine learning algorithms for classification and regression problems. CO5. Develop machine learning modes and evaluate performance. CO6. Build the concepts of reinforcement learning. CO7. Recognizing different Deep learning models and used in solving problems.

Learning Resources	1, Machine Learning - Theory and practice M N Murty and Anantharayana V S 2. Alpaydin, Ethem. Introduction to machine learning. MIT press, 2020. 3. Artificial Intelligence, Rich and Knight 4. Explorations in Artificial Intelligence and Machine Learning https://www.routledge.com/rsc/downloads/AI_FreeBook.pdf 5. Artificial Intelligence With an Introduction to Machine Learning By Richard E. Neapolitan, Xia Jiang

2nd Semester Courses

Course Code	CS5201N	**Course Name**	Mathematics for Computer Science	**Course Category**	PC4	L	T	P
						3	0	0

Pre-requisite Courses	Discrete Structures, Theory of Computations, Probability and Statics	**Co-requisite Courses**	Advanced Data Structures and Algorithms , Artificial Intelligence / Machine Learning, Computer Systems	**Progressive Courses**	Advanced Artificial Intelligence / Machine Learning, Advanced Optimization Techniques, Advanced Complexity Theory, Cryptography and Security
Course Offering Department		CST		**Data Book / Codes/Standards**	

Course Objective	The course aims to equip postgraduate students with a strong mathematical foundation essential for advanced computing disciplines. It develops students' analytical and logical reasoning skills through the study of discrete structures, linear algebra, probability, and formal logic. Students will learn to model, analyze, and solve complex computational problems using rigorous mathematical methods, preparing them for both research and real-world applications in computer science.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Discrete Mathematics and Linear Algebra:** Logic & Proof Techniques (direct, contradiction, induction), Set Theory & Relations, Functions and Recursion, Lattice, Counting, Graph Theory (trees, connectivity, coloring), Vectors and Matrices, Eigenvalues and Eigenvectors, Matrix Decompositions (LU, QR, SVD), Applications in machine learning (PCA, LDA, etc.)	10	i) Understand and apply logic and proof techniques. ii) Apply graph theory and combinatorial methods for problem-solving in computer science. iii) Understand and utilize concepts of linear algebra. iv) Apply mathematical tools in data analysis.
2	**Probability and Statistics:** Probability Spaces and Random Variables, Expectation, Variance, Covariance, Distributions (Binomial, Poisson, Normal), Bayesian Inference, Hypothesis Testing, Applications in Artificial Intelligence and machine learning (Naive Bayes Classification, Decision Tree, Ensemble Classification, etc.)	10	i) Understand fundamental concepts of probability theory. Analyze and apply standard probability distributions in solving real-world problems. ii) Apply statistical inference techniques, including Bayesian inference and hypothesis testing. iii) Utilize probabilistic and statistical methods in Data Analysis and predictive modeling.

3	**Mathematical Logic and Computability:** Propositional and Predicate Logic, Formal Systems and Proof Theory, Lambda Calculus, Turing Machines, Reductions, Undecidable problems, Complexity Theory.	10	i) Develop a solid understanding of propositional and predicate logic. ii) Gain proficiency in foundational computational models. iii) Understand the concepts of decidability and undecidability. iv) Explore fundamental aspects of complexity theory.
4	**Optimization Techniques:** Linear Optimization (overview), Nonlinear Optimization (Unconstrained and Constrained), Dynamic Programming, Convex Optimization, Integer Programming, Applications in Neural Network Training, Support Vector Machine, Cryptography, Deep Learning, Reinforcement Learning, etc.	12	i) Understand and apply various optimization techniques. ii) Formulate and solve optimization problems relevant to computer science and engineering domains. iii) Explore the role of optimization in various applications.

Course Outcome	1. **Apply discrete mathematical structures and proof techniques** to analyze algorithms and computational logic relevant to computer science problems. 2. **Demonstrate proficiency in linear algebra concepts**, including matrix operations and eigenvalue analysis. 3. **Analyze and model probabilistic systems using statistical methods** to support decision-making. 4. **Understand and evaluate formal systems, computability, and complexity**, including the design and limitations of Turing Machines and undecidable problems. 5. **Formulate and solve optimization problems** using linear, nonlinear, dynamic, and integer programming techniques. 6. **Integrate mathematical reasoning and computational thinking** to design and evaluate solutions to complex computer science problems across theoretical and applied domains.

Learning Resources	Books Referred: ***Text Books:*** Eric Lehman, Tom Leighton, and Albert Meyer, Mathematics for Computer Science Probability and Statistics for Computer Scientists – Michael Baron **Reference Books:** 1. Kenneth H. Rosen, Discrete mathematics and its applications. 2. *Convex Optimization* – Stephen Boyd and Lieven Vandenberghe 3. Introduction to the Theory of Computation by *Michael Sipser* 4. Convex Optimization by *Stephen Boyd and Lieven Vandenberghe* 5. Numerical Optimization by *Jorge Nocedal and Stephen Wright*

Course Code	CS5202N	Course Name	Advanced Database Management Systems	Course Category	PC5	L	T	P
						3	0	0

Pre-requisite Courses	Database Management System, Data Structure and Algorithms	**Co-requisite Courses**	–	**Progressive Courses**	Big-data Analytics, Cloud Computing
Course Offering Department		Computer Science and Technology		**Data Book / Codes/Standards**	–

Course Objective	Prepare students with the knowledge and proficiency in advanced database mgt. techniques including relational, distributed, object oriented, mobile databases with an objective to make them skilled database designers and successful developers in various domains of applications.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Review of RDBMS:** Normal Forms, FDs, Decomposition and synthesis approaches, File organization, external sorting, Processing of joins, Query optimization, query transformation rules, DB transactions, ACID properties, interleaved executions, schedules, serializability, Recovery techniques, concepts of in-memory database	10	Recapitulation on understanding of database fundamentals, including data modeling, SQL querying, and database design principles
2	**Distributed Database Mgt. Systems:** Features of distributed vs. centralized database; levels of distribution transparency, Reference architecture, Fragmentation, Distribution transparency; Distributed Database Design, Framework, Design of Database fragmentation, Allocation of fragments; Equivalence transformations for queries, Transforming global queries into fragment queries; Query Optimization, Distributed database transaction management, commit protocols, failure handling	16	Efficient management of database that is physically distributed across multiple locations but work as part of a single and centralized system
3	**Object Oriented DBMS:** Creation and modeling of data as objects, relationship, identifiers, classes of objects, Type hierarchies and inheritance, Overview of Object Query language (OQL), Graph database	6	Understanding the principles and implementation of object-oriented databases, including concepts like object identity, encapsulation, inheritance, and polymorphism.

4	**Overview of Bigdata management and MongoDB**	5	Equip students to understand the basic concepts of big data, its characteristics, and the associated challenges. Familiarization with NoSQL databases, especially MongoDB, and its appropriateness to handle big data is also essential.
5	**Overview of Mobile Database Systems**	5	Understand and apply the basic principles of managing data in mobile environments. Learning the associated challenges is also essential.

Course Outcome	CO1: Review traditional DBMS (e.g. Relational) including Query optimization, DB transactions, Recovery techniques and concepts of in-memory database CO2: Understand Distributed DBMS focusing on Reference architecture, Design of distributed Database, Query Optimization, Distributed database transaction management CO3: Ability to differentiate between centralized and distributed databases and judiciously applying in various problem domain CO4: Explain non-relational database (e.g. Graph Database) and its suitability in many real-life applications CO5: Implement applications requiring usage of non-relational database

Learning Resources	Books Referred: *Text Books:* 1. A. Silberschatz, H. Korth, S. Sudarshan, Database system concepts, 5/e, McGraw Hill. 2. S. Ceri, G. Pelagatti, Distributed Databases, Principles & Systems, McGraw Hill. 3. R. Elmasri, S. B. Navathe, Fundamentals of Database Systems, The Benjamin/ Cummings Publishing Company, Latest Edition. *Reference Books:* S. W. Dietrich, S. D. Urban, An Advanced Course in Database Systems; Beyond Relational Databases, PEARSON Education, Latest Edition Vijay Kumar, Mobile Database Systems, WILEY.

Course Outcome	CO1: Understanding of information retrieval concepts, including indexing, querying, and ranking. CO2: Familiarity with various information retrieval models, such as Boolean, vector space, and probabilistic models. CO3:

Learning Resources	**References / Books:** 1. Introduction to Information Retrieval. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schuetze, Cambridge University Press. 2. Search Engines: Information Retrieval in Practice. Bruce Croft, Donald Metzler, and Trevor Strohman, Pearson Education. 3. Modern Information Retrieval. Baeza-Yates Ricardo and Berthier Ribeiro-Neto. 2nd edition, Addison-Wesley. 4. Information Retrieval: Implementing and Evaluating Search Engines. Stefan Buttcher, Charlie Clarke, Gordon Cormack, MIT Press.

Course Code	CS5203N	Course Name	Advanced Networking	Course Category	PC6	L	T	P
						3	0	0

Pre-requisite Courses	*(i)Computer Networks*	Co-requisite Courses	*(i)Cryptography & Network Security (ii) Mathematics for Computer Science*	Progressive Courses	*(i) Internet of Things (ii) Blockchain Technologies*
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	
	1. Deepen understanding of advanced principles, architectures, and mechanisms underpinning modern computer networks. 2. Critically examine the design, operation, and evolution of advanced data-link, transport, routing, and application-layer protocols. 3. Analyze emerging trends such Software Defined Networking (SDN), Network Function Virtualization (NFV), and cloud-scale networking. 4. Explore performance considerations in complex networking environments, including quality of service (QoS), congestion control, and scalability. 5. Develop the ability to evaluate research literature and conduct comparative studies of advanced networking technologies. 6. Understand the role of security, trust, and resilience in advanced network protocol design and deployment.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Introduction:** Data communication Components: Representation of data and its flow Networks; Various Connection Topology; Protocols and Standards; TCP/IP protocol suite; OSI model.	4	This module gives an overview to help brush up basic computer networks knowledge studied in UG level.
2	**Wireless Networks:** Introduction; Wireless Links and Network Characteristics; WiFi:802.11 Wireless LANs - architecture, MAC protocol, IEEE 802.11 frame; Mobility in the same IP Subnet; WiMAX/IEEE 802.16; Personal Area Network - IEEE 802.15.2 Bluetooth.	9	-Understand the fundamentals, characteristics, and challenges of wireless networks. -Explain and illustrate the architecture and operational modes of Wi-Fi (IEEE 802.11) and assess mobility support within the same IP subnet. -Compare and evaluate WiMAX and Wi-Fi in terms of design, architecture, MAC/PHY layers, and Quality of Service (QoS) mechanisms. -Describe the architecture, protocol stack, and communication methods of Bluetooth (IEEE 802.15.1) personal area networks.

3	**Mobile Networks:** Cellular Internet Access: Overview; 3G Cellular Data Networks; 4G LTE; 5G Wireless Networks; Mobility Management - Principles, Mobile IP; Managing Mobility in Cellular Networks - Routing, Handoffs in GSM.	9	-Understand the architecture and evolution of mobile cellular networks including 3G, 4G LTE, and 5G systems. -Explain how mobile devices access the Internet over cellular networks and describe the associated protocols and standards. -Describe and analyze mobility management principles including Mobile IP, and their role in seamless connectivity. -Evaluate mobility handling mechanisms in cellular networks such as routing strategies and handoff techniques in GSM and other mobile systems.

4	**Multimedia Networks:** Multimedia Networking Applications - properties of video, audio, types of multimedia applications; Streaming Stored Video - UDP Streaming, HTTP Streaming, Adaptive Streaming, Content Distribution Network (CDN), Case Studies; Voice-over-IP (VoIP); Protocols for Real-Time Conversational Applications - RTP, SIP.	10	-Understand the characteristics of multimedia data and the requirements of various multimedia networking applications. -Explain and compare different approaches for video streaming, as well as the role of Content Distribution Networks (CDNs). -Describe the architecture and functioning of Voice-over-IP (VoIP) systems. -Analyze the protocols supporting real-time conversational applications, particularly RTP and SIP, and assess their effectiveness in multimedia delivery.

5	**Quality of Service:** Data flow characteristics - definition, sensitivity of applications, flow classes; Flow control to improve QoS - scheduling, traffic shaping or policing, resource reservation; Integrated services (INTSERV) - flow specification, admission, service classes, resource reservation protocol (RSVP); Differentiated services (DIFFSERV) - DS field, per-hop behavior, traffic conditioners.	10	- Understand the characteristics and sensitivity of different data flows and classify network traffic based on application requirements. -Explain QoS enhancement techniques such as scheduling, traffic shaping/policing, and resource reservation. -Describe and analyze the Integrated Services (INTSERV) model, including flow specification, admission control, and the use of RSVP. -Evaluate the Differentiated Services (DIFFSERV) architecture, including DS field configuration, per-hop behavior, and traffic conditioning mechanisms.

Course Outcome	CO1: Analyze the design and performance of advanced protocols, synthesizing disciplinary knowledge through theoretical models. CO2: Design and propose enhancements to network systems (e.g., congestion control, QoS schemes), showcasing innovation and problem- solving skills. CO3: Critically review and present recent research in emerging networking areas CO4:Collaborate in multidisciplinary teams to tackle complex networking challenges, reflecting communication, teamwork, and leadership abilities.

Learning Resources	References: 1. Behrouz A. Forouzan. “Data Communications and Networking”, 5th Edition, McGraw Hill Publishing Co. 2. William Stallings. “Data and Computer Communications”, 10th edition*, Pearson Larry Publication.* 3. Andrew S. Tanenbaum and David J. Wetherall. “Computer Networks”, 5th Edition*, Pearson Larry Publication.* 4. James Kurose and Keith Ross, "Computer Networking: A Top-Down Approach" 6th edition, *Pearson Larry Publication*.

Course Code	CS5221N	Course Name	Embedded Systems for Secured Hardware Design	Course Category	PSE2	L	T	P
						3	0	0

Pre-requisite Courses	*Digital Logic Design, Computer Organization and Computer Architecture*	Co-requisite Courses	*Microprocessor and Microcontroller, Operating System*	Progressive Courses	
Course Offering Department		***CST***		Data Book / Codes/Standards	

Course Objective	1. To introduce the fundamentals of embedded systems, including microcontrollers, real-time operating systems (RTOS), and hardware-software co-design. 2. To explore the security vulnerabilities in embedded systems and the threat landscape affecting hardware platforms. 3. To study cryptographic algorithms and hardware-based countermeasures, including secure boot, hardware encryption, and tamper-resistant design. 4. To develop skills to design and implement secure embedded architectures, considering both hardware and firmware layers. 5. To enable students to work on case studies and projects involving real-world security challenges in embedded system design

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Introduction** Concept of embedded system. Processors and hardware units for embedded,system, embedded software, hardware –software co-design, system on chip	3	Able to understand the fundamentals of embedded systems, including processors, hardware components, embedded software, hardware-software co-design, and system-on-chip (SoC) architectures.
2	**System specification and system analysis** Requirements, analysis, and modelling techniques	3	Able to analyze the requirement analysis of a system
3	**System design techniques** Traditional approach, hardware software partitioning, system architecture,hardware synthesis, software design, optimization issues, POLIS co-design methodology	6	Able to understand the design techniques and optimization issues
4	**Implementation platforms** Microcontrollers, DSP, PIC, FPGA	5	Able to adopt implementation platform for an embedded system
5	**Real time issues** System monitoring, RTOS, clock synchronization, wireless devices, parallelism	5	Able to understand real times issues of embedded systems

6	**Fault tolerance techniques** Fault tolerant design principles, redundancy, testing of embedded systems, time and data. Byzantine algorithm	5	Will gain knowledge of fault tolerant techniques of embedded systems
7	**Hardware security** Issues of hardware security. Cryptographic hardware and their Implementation, optimization of cryptographic hardware on FPGA. Physically Unclonable Functions (PUFs), PUF implementations. Hardware Trojan: Operating modes, countermeasures, logic testing and side-channel analysis based techniques for Trojan detection	8	Able to understand security issues in embedded systems
8	**System design Case studies** Fault-tolerance of cryptographic hardware. Modeling of arbiterPUFs	5	Able to learn some case studies

Course Outcome	CO1: To be able to understand the architecture, components, and operation of embedded systems. CO2: To be able to identify and analyze potential security threats and vulnerabilities in embedded hardware platforms. CO3: To be able to design secure embedded system architectures incorporating both hardware and firmware countermeasures. CO4: To be able to implement and verify secure embedded designs using tools.

Learning Resources	References: 1. Debdeep Mukhopadhyay and Rajat Subhra Chakraborty: Hardware Security: Design, Threats andSafeguards, CRC Press. 2. Ahmad-Reza Sadeghi and David Naccache (eds.): Towards Hardware-intrinsic Security: Theory andPractice, Springer. 3. Ted Huffmire et al: Handbook of FPGA Design Security, Springer. 4. Pak Chan et. al.: Digital Design Using FPGAs, Prentice Hall. 5. Jesse H. Jenkin: Designing with FPGAs and CPLDs, Prentice Hall. 6. Zainalabedin Navabi: VHDL. Analysis and Modelling of Digital Systems, McGraw-Hill

Course Code	CS5222N	Course Name	Natural Language Processing	Course Category	PSE2	L	T	P
						3	0	0

Pre-requisite Courses	*Programming skills, Data structures and algorithms, Machine learning fundamentals*	**Co-requisite Courses**	*Data Structures and Algorithms, Machine learning fundamentals*	**Progressive Courses**	Information Retrieval, *Multimodal NLP,*
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	Upon completing this course, students will be able to: 1. Understand the fundamental concepts and techniques of Natural Language Processing. 2. Design and develop NLP systems for various applications, such as text classification, sentiment analysis, and language translation. 3. Apply machine learning and deep learning algorithms to NLP tasks. 4. Analyze and evaluate the performance of NLP systems. 5. Develop practical skills in using NLP tools and libraries, such as NLTK, spaCy, and transformers.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Introduction to NLP:** Definition, issues and strategies, application domain, tools for NLP, Linguistic organisation of NLP, NLP vs PLP.	2	Understand how NLP processes are structured based on linguistic levels.
2	**Word Classes:**	10	
	Review of Regular Expressions, CFG and different parsing techniques, **Morphology:** Inflectional, derivational, parsing and parsing with FST, Combinational Rules **Phonology:** Speech sounds, phonetic transcription, phoneme and phonological rules, optimality theory, machine learning of phonological rules, phonological aspects of prosody and speech synthesis. Pronunciation, **Spelling and N-grams:** Spelling errors, detection and elimination using probabilistic models, pronunciation variation (lexical,allophonic, dialect), decision tree model, counting words in Corpora, simple N-grams, smoothing (Add-One, Written-Bell, Good-Turing), N- grams for spelling and pronunciation.		Understand word structure,pronunciation, parsing techniques.
3	**Syntax:** **POS Tagging:** Tagsets, concept of HMM tagger, rule based and stochastic POST, algorithm for HMM tagging, ransformation based tagging **Sentence level construction & unification:** Noun phrase, co-ordination, sub-categorization, concept of feature structure and unification.	6	Learn how to apply PoS tagging, sentence structure, unification.

	Semantics : **Representing Meaning:** Unambiguous representation, canonical form, expressiveness, meaning structure of language, basics of FOPC **Semantic Analysis:** Syntax driven, attachment & integration, robustness **Lexical Semantics:** Lexemes (homonymy, polysemy, synonymy, hyponymy), WordNet, internal structure of words, metaphor and metonymy and their computational approaches Word Sense Disambiguation:Selectional restriction based, machine learning based and dictionary based approaches.	9	Analyze meaning, resolve references, generate language.
	Pragmatics: **Discourse:** Reference resolution and phenomena, syntactic and semantic constraints on Coreference, pronoun resolution algorithm, text coherence, discourse structure **Dialogues:** Turns and utterances, grounding, dialogue acts and structures Natural Language Generation: Introduction to language generation, architecture, discourse planning (text schemata, rhetorical relations). Standardization using Unicode and Code Conversion.	10	
	Deep Learning for NLP: Basic and Advanced Models such as RNNs, LSTMs, and transformers	4	Learning how to apply deep learning models in NLP.

Course Outcome	CO1: Understand the theoretical foundations of NLP, including linguistics and machine learning. CO2: Develop practical skills in NLP, including text processing, tokenization, and sentiment analysis. CO3: Apply NLP techniques to real-world problems, such as text classification, language translation, and information retrieval. CO4: Evaluate the performance of NLP systems and identify areas for improvement.

Learning Resources	**References:** 1. Diana Maynard, Kalina Bontcheva, Isabelle Augenstein, "Natural Language Processing for the Semantic Web", A Publication in the Morgan & Claypool Publishers series 2. Steven Bird, Ewan Klein and Edward Loper, "Natural Language Processing with Python", By O'Reilly 3. Christopher Manning and Hinrich Schütze, "Foundations of Statistical Natural Language Processing", The MIT Press

Course Code	CS5223N	**Course Name**	Computational Geometry	**Course Category**	PSE2	L	T	P
						3	0	0

Pre-requisite Courses	*Discrete Structure, Data Structures*	**Co-requisite Courses**	*None*	**Progressive Courses**	
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	
	1. To be familiar with topics in geometric structures and algorithms as listed in the course contents 2. To be familiar with how to design algorithms for geometric problems. 3. Master a subset of algorithms: Convex hull, Delaunay triangulations, and voronoi diagrams etc. 4. to be familiar with how to research the background of a topic in geometric algorithms

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Historical perspective: complexity notions in classical geometry. Towards computational geometry, geometric preliminaries, models of computation.	6	
2	Geometric searching: point location problems, location of a point in a plannar subdivision, the slab method, the chain method, range - searching problems.	10	
3	Convex hulls: problem statement and lower bounds. Graham's scan, Jarvis's march, quick hull technique, convex hulls in two and higher dimensions, extension and applications.	8	
4	Proximity: divide and conquer approach, locus approach; the Voronoi diagram, Delauney triangulation, Arrangement and duality, triangulation of polygon lower bounds, variants and generalizations. Intersections, hidden-line and hidden surface problem.	8	
5	The geometry of rectangles: application of the geometry of rectangles, measure and perimeter of a union of rectangles, intersection of rectangles and related problems.	4	

Course Outcome	PO1: Construct algorithms for simple geometrical problems. PO2: Implement computational geometry algorithms. PO3: Read, understand, analyze and modify a given algorithm. Ability to design algorithmic solutions for given geometric problems. PO4: Carry out independent research

Learning Resources	References: 1. F. P. Preparata and M.I. Shamos: Computational Geometry: An Introduction. 2. M. De Berg, M. Van Kreveld, M. Overmars and 0. Schwarzkopf: Computational Geometry: Algorithms and Applications 3. J. O'Rourke: Computational Geometry in C, Cambridge University Press, London. 4. K. Mehlhom and St. Naher: The LEDA Platform of Combinatorial and Geometric Computing

Course Code	CS5224N	**Course Name**	Internet of Things	**Course Category**	PSE2	L	T	P
						3	0	0

Pre-requisite Courses	*(i) Computer Networks (ii) Operating Systems (iii) Microprocessor-based Systems (iv) Data Structures (v) Fundamental Programming*	**Co-requisite Courses**	*(i) Cryptography and Network Security (ii) Machine Learning*	**Progressive Courses**	*(i) Industrial IoT (ii) IoT Security and Privacy (iii) Blockchain for IoT (iv) Federated Learning on Edge Device*
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	The objective of this course is to impart necessary and practical knowledge of components of Internet of Things and develop skills required to build real-life IoT based projects.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Introduction to IoT:** Architectural Overview, Design principles and needed capabilities, IoT Applications, Sensing, Actuation, Basics of Networking, M2M and IoT Technology Fundamentals- Devices and gateways, Data management, Business processes in IoT, Everything as a Service (XaaS), Role of Cloud in IoT, Security aspects in IoT.	10	-Understand the architecture, design principles, and essential capabilities required for Internet of Things (IoT) systems. -Describe key IoT components including sensors, actuators, devices, gateways, and networking basics, along with M2M communication and data flow. -Analyze IoT data management, business processes, and the concept of Everything as a Service (XaaS). -Evaluate the role of cloud computing in IoT and identify fundamental security considerations for IoT applications.

2	**Elements of IoT:** Hardware Components- Computing (Arduino, Raspberry Pi), Communication, Sensing, Actuation, I/O interfaces. Software Components- Programming API's (using Python/Node.js/Arduino) for Communication Protocols-MQTT, ZigBee, Bluetooth, CoAP, UDP, TCP.	14	-Understand the essential hardware components of IoT systems. -Explain and apply key protocols such as MQTT, ZigBee, Bluetooth, CoAP, UDP, and TCP in IoT applications. -Develop and test basic IoT programs using Python, Node.js, or Arduino for interfacing hardware and implementing communication.

3	**IoT Application Development:** Solution framework for IoT applications-Implementation of Device integration, Data acquisition and integration, Device data storage-Unstructured data storage on cloud/local server, Authentication, authorization of devices.	14	-Understand the solution framework and workflow for developing end-to-end IoT applications. -Implement device integration and data acquisition techniques for real-time IoT systems. -Manage unstructured device data storage on cloud or local servers. -Apply authentication and authorization mechanisms to ensure secure device communication.

4	**Case Studies:** IoT case studies and mini projects based on Industrial automation, Transportation, Agriculture, Healthcare, Home Automation	6	-Analyze real-world IoT applications across domains. -Design and implement mini projects that demonstrate the integration of IoT concepts and technologies to solve domain-specific problems. -Evaluate the effectiveness, challenges, and impact of IoT solutions in various application areas.

Course Outcome	CO1: Understand internet of Things and its hardware and software components CO2: Interface I/O devices, sensors & communication modules CO3: Remotely monitor data and control devices CO4: Develop real life IoT based projects

Learning Resources	**Reference Books:** 1. Vijay Madisetti, Arshdeep Bahga, "Internet of Things - A Hands on Approach", University Press. 2. Cuno Pfister, "Getting Started with the Internet of Things", O Reilly Media. 3. Pethuru Raj and Anupama C. Raman, “The Internet of Things: Enabling Technologies, Platforms, and Use Cases”, CRC Press 4. Adrian McEwen, “Designing the Internet of Things”, Wiley 5. Raj Kamal, “Internet of Things: Architecture and Design”, McGraw Hill

Course Code	CS5225N	**Course Name**	Quantum Computing	**Course Category**	PSE2	L	T	P
						3	0	0

Pre-requisite Courses	*Algorithm, Signals and Systems, Linear Algebra, Information and coding theory*	**Co-requisite Courses**	*NIL*	**Progressive Courses**	*NIL*
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	To familiarize with the concept of quantum computing and quantum communication as an emerging domain

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Introduction:** Introducing quantum mechanics. Quantum kinematics, quantum dynamics, quantum measurements. Single qubit, multiqubits, gates. Density operators, pure and mixed states, quantum operations, environmental effect, decoherence. Quantum no-cloning, quantum teleportation.	5	Understand the basics of quantum mechanics
2	Quantum Gates, Quantum circuits, Universal gates, quantum parallelism, Deutsch-Josza algorithm.	3	Understand basics of quantum circuits
3	**Quantum Entanglement:** Quantum correlations, Bell's inequalities, EPR paradox. Theory of quantum entanglement. Entanglement of pure bipartite states. Entanglement of mixed states. Peres partial transpose criterion. NPT and PPT states, bound entanglement, entanglement witnesses.	8	Understand concepts of quantum physics
4	**Quantum Algorithm:** Introduction to quantum algorithms. Deutsch-Jozsa algorithm, Grover's quantum search algorithm, Simon's algorithm. Shor's quantum factorization algorithm.	10	Ability to explain quantum algorithms

5	**Quantum Error Correction:** Errors and correction for errors. Simple examples of error correcting codes in classical computation. Linear codes. Shor code, Theory of Quantum Error–Correction, Constructing Quantum Codes, Stabilizer codes, Fault – Tolerant Quantum Computation, Entropy and information – Shannon Entropy, Basic properties of Entropy, Von Neumann, Strong Sub Additivity, Data Compression, Entanglement as a physical resource.	6	Understand quantum communication and design error correcting codes
6	**Quantum Cryptography:** Cryptography, classical cryptography, introduction to quantum cryptography. BB84, B92 protocols. Introduction to security proofs for these protocols.	4	Understand quantum cryptography schemes
7	**Implementations:** Different implementations of quantum computers. NMR and ensemble quantum computing, Ion trap implementations. Optical implementations.	4	Appreciate some current implementations

Course Outcome	ability to understand the basics of quantum information ability to explain and utilize the phenomena of quantum physics ability to design algorithms based on quantum circuits ability to design quantum communication and cryptography systems ability to appreciate different quantum computer implementation technologies

Learning Resources	**Reference Books:** 1. Micheal A. Nielsen &Issac L. Chuang, “Quantum Computation and Quantum Information”, Cambridge University Press, 10th Anniversary Edition 2. Phillip Kaye Raymond Laflamme Michele Mosca, “An Introduction to Quantum Computing”, Oxford University Press

Course Code	CS5226N	**Course Name**	Graph Algorithms	**Course Category**	PSE2	L	T	P
						3	0	0

Pre-requisite Courses		**Co-requisite Courses**		**Progressive Courses**	
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	1. Understanding of fundamental graph concepts, including graph types, traversals, and representations. 2. Familiarity with various graph algorithms. 3. Knowledge of very large graph properties and their applications.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Introduction and review of graphs:** graphs and digraphs, subgraphs, representation of graphs, Graph traversal, complete graphs, regular graphs, Petersen graph, Bipartite graphs, Isomorphism of graphs. Walks, trails, paths, connected graphs, distance, cut-vertices, cut-edges, blocks, weighted graphs, connectivity, Minimum spanning tree, Single source & all pair shortest path.	8	Understand overview of graph theory, covering fundamental concepts equips learners with knowledge of graph theory for efficient problem solving.
2	**Network Flow:** The flow problem, flow and cut, Max flow min cut theorem, Ford Fulkerson, Shortest Path Augmentation algorithms	5	Learn the optimization of resource movement through networks
3	**Planarity of graphs:** Significance of planar graphs, Kuratowski's graphs, characterizing planar graphs – Euler's theorem, Planarity testing algorithms	6	Understand, characterize, and test planar graphs.
4	**Matching and covering in graphs:** maximum matching; perfect matching; matching in bipartite graphs - Konig's theorem, Hall's marriage theorem; matching in general graph-Tutte's theorem; Path covering - Gallai-Milgram theorem, Dilworth theorem.	5	Analyze matching, covering, and related theorems.
5	**Graph coloring:** Proper coloring of graphs, Chromatic number, algorithms for coloring, coloring of planar graphs, applications.	4	Apply graph coloring concepts and algorithms.
6	**Large graphs:** Structural properties of large graphs – degree distribution, clustering coefficient, node centrality, Dense Subgraphs, Applications	6	Analyze structural properties of large graphs.

7	**Design and Analysis of Algorithms for large graphs:** Ranking algorithms – PageRank, Community detection algorithms and their time complexity analysis, Applications	6	Design, analyze algorithms for large graphs.

Course Outcome	CO1: Knowledge of the basics of graphs, directed graphs, weighted graphs, and ability to apply algorithmic techniques to study basic properties of graphs CO2: Knowledge of algorithms for solving various problems in graphs such as flow optimization, matching, planarity testing, coloring CO3: Ability to map real-life problems in various domains to problems on graphs, and to apply suitable algorithms to solve them CO4: Knowledge of research problems on large graphs (such as the Web and social networks) and algorithms for such problems

Learning Resources	**References and Books:** 1. D.B. West, Introduction to Graph Theory, 2 nd Edition, PHI. 2. G. Chatrand and O.R. Oellermann, Applied and algorithmic Graph Theory, McGraw–Hill, Inc 3. Ton Kloks, Mingyu Xiao, “A Guide to Graph Algorithms”, Springer Singapore, https://doi.org/10.1007/978-981-16-6350-5

Course Code	CS5227N	**Course Name**	Cyber Security and Forensics	**Course Category**	PSE2	L	T	P
						3	0	0

Pre-requisite Courses	*Computer Networking*	**Co-requisite Courses**		**Progressive Courses**	
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	To provide the knowledge and skills to protect computer systems, detect and analyze cyber threats, and investigate cybercrimes while understanding legal and ethical aspects of digital security.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Cyber Security Concepts** Essential Terminologies: CIA, Risks, Breaches, Threats, Attacks, Exploits. Information Gathering (Social Engineering, Foot Printing & Scanning). Open Source/ Free/ Trial Tools: nmap, Zenmap, Port Scanners, Network scanners.	4	Students will understand core cyber security concepts, terminologies, and perform basic information gathering using common open-source tools.
2	**Infrastructure and Network Security** Introduction to System Security, Server Security, OS Security, Physical Security, Introduction to Networks, Network packet Sniffing, Network Design Simulation. DOS/ DDOS attacks. Asset Management and Audits, Vulnerabilities and Attacks. Intrusion detection and Prevention Techniques, Host based Intrusion prevention Systems, Security Information Management, Network Session Analysis, System Integrity Validation. Open Source/ Free/ Trial Tools: DOS Attacks, DDOS attacks, Wireshark, Cain & abel, iptables/Windows Firewall, snort, Suricata, fail2ban	8	Students will learn to secure systems and networks, detect and prevent intrusions, and analyze threats using simulation and open-source security tools.

3	**Cyber Security Vulnerabilities & Safe Guards** Internet Security, Cloud Computing & Security, Social Network sites security, Cyber Security Vulnerabilities-Overview, vulnerabilities in software, System administration, Complex Network Architectures, Open Access to Organizational Data, Weak Authentication, Authorization, Unprotected Broadband communications, Poor Cyber Security Awareness. Cyber Security Safeguards-Overview, Access control, IT Audit, Authentication. Open Web Application Security Project (OWASP), Web Site Audit and Vulnerabilities assessment.	10	Students will understand common cyber security vulnerabilities across platforms and apply safeguards like access control, authentication, and web auditing using OWASP guidelines.

4	**Security in Evolving Technologies** Biometrics, Mobile Computing and Hardening on android and ios, IOT Security, Web server configuration and Security. Introduction, Basic security for HTTP Applications and Services, Basic Security for Web Services like SOAP, REST etc., Identity Management and Web Services, Authorization Patterns, Security Considerations, Challenges. Open Source/ Free/ Trial Tools: adb for android, xcode for ios, Implementation of REST/ SOAP web services and Security implementations.	10	Students will explore security challenges in evolving technologies like biometrics, mobile devices, IoT, and web services, and apply basic security measures using practical tools.

5	**Cyber Law & Forensics** Introduction, Cyber Security Regulations, Roles of International Law, the state and Private Sector in Cyberspace, Cyber Security Standards. The INDIAN Cyberspace, National Cyber Security Policy 2013. Introduction to Cyber Forensics, Need of Cyber Forensics, Cyber Evidence, Documentation and Management of Crime Sense, Image Capturing and its importance, Partial Volume Image, Web Attack Investigations, Denial of Service Investigations, Internet Crime Investigations, Internet Forensics, Steps for Investigating Internet Crime, Email Crime Investigations.	10	Students will understand cyber laws, national policies, and forensic investigation techniques for cybercrimes including evidence handling, web, email, and DoS attack investigations.

Course Outcome	CO1: Identify and analyze cyber threats and vulnerabilities in systems and networks. CO2: Apply cybersecurity measures like firewalls, encryption, and secure protocols to protect data. CO3: Conduct digital forensic investigations to collect, analyze, and preserve electronic evidence. CO4: Use forensic tools and techniques for examining computers, networks, and mobile devices. CO5: Understand cyber laws and ethical practices related to information security and digital forensics. CO6: Respond effectively to cyber incidents through proper incident handling and recovery procedures.

Learning Resources	**Reference s /Books:** 1. William Stallings, “Cryptography and Network Security”, Pearson Education/PHI. 2. Matt Bishop, "Computer Security - Art and Science", Addison-Wesley. 3. Sarika Gupta, “Information and Cyber Security”, Khanna Publishing House, Delhi. 4. Nina Godbole, “Information System Security”, Wiley 5. Bothra Harsh, “Hacking”, Khanna Publishing House, Delhi.

Course Code	CS5228N	Course Name	Deep Learning	Course Category	PSE2	L	T	P

Pre-requisite Courses	Linear algebra, calculus, probabilities & statistics, Introductory course on ML, Programming skill	Co-requisite Courses	Applied Machine Learning	Progressive Courses	Generative AI
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	
	1. Understanding major differences between deep learning and other types of machine learning algorithms. 2. Introducing the principles of deep learning and its applications. 3. Developing practical skills to design, implement, and train deep learning systems. 4. Providing a structured approach covering Fundamentals of Machine Learning, Neural Networks, Advance Deep Learning, Its Applications .

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to Deep Learning: Overview of machine learning, Deep learning vs. Machine learning, Difference between Shallow and Deep Neural Networks, Feature Engineering vs Feature Learning, Bias variance tradeoff, Overfitting, Under fitting problem, Regularization	4	(i) Understanding flow of development shallow to deep learning models (ii) understanding different issues related to learning algorithms
2	Neural Network Basics: Linear classifier, Gradient Descent, Cost function, Activation functions, Backpropagation Algorithm, Vanishing Gradient problem, Adam Optimizer, Hyperparameter tuning, dropout. AutoEncoder, Variational Autoencoders (VAEs)	6	Introducing techniques used to train ANN, its parameters, hyperparameters
3	Convolutional Neural network: Architecture, different layers, Kernel filter, Multiple Filters, Channels, Convolution operation, Feature Map, Batch Normalization, Fully Convolutional NN	10	Equip with the Characteristics of CNN , Concepts of convolution operation, layers, classification
4	Sequential Modelling: Recurrent Neural network, Seq2Seq RNNs, LSTM, GRU, Encoder Decoder architectures, Attention Models, Encoder-Decoder Model.	10	Equip with the Characteristics of RNN, LSTM, training algorithm
5	Advance Topics: Generative Adversarial Network (GAN), Variants of GAN and applications, Transfer learning and domain adaptation, VGG, Inception (GoogLeNet), ResNet architectures Transformer Models, Graph Neural Network	8	(i) Distinguish Generative vs Adversarial network, (ii) Different variants of CNN
6	Applications of Deep Learning Models in Computer vision, Natural language Processing, text generation, and time series prediction, Multimodal Data handling	4	Real life applications using deep learning models.

Course Outcome	CO1: Demonstrate the ability to apply concepts of Mathematical, Stochastic Models, optimization, and machine learning to solve problems in the context of deep learning. CO2: Understanding the difference of applying deep learning neural network architectures and other approaches in the context of solving complex problems . CO3: Identifying key architecture parameters to learn the input-output function approximation CO4: Design, implement, and train deep learning models using convolutional, recurrent Neural Network. CO4: Solving real-world computer vision problems, NLP tasks using multimodal data

Learning Resources	1. Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT Press. 2. Pattern Recognition and Machine Learning by Bishop, C., M., Springer 3. Deep Learning for Computer Vision by Rajalingappaa Shanmugamani, Packt Publishing. 4. Generative Deep Learning by David Foster, O'Reilly Media.

Course Code	CS5229N	**Course Name**	In-Memory Computation	**Course Category**	PSE2	L	T	P
						3	0	0

Pre-requisite Courses	Basic Electronics, Digital Logic, Computer Architecture	**Co-requisite Courses**		**Progressive Courses**	
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	To explore the limitations of von Neumann architecture and to document the latest directions and trends in designing the platforms for compute- intensive applications. To investigate in-memory and near-memory computing architecture.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Introduction:** History of computing, the current need, technology road map, performance measure, Continuum computer for exaflops computation	3	
2	**Today's computers:** Introduction to von Neumann architecture, Pipelining datapath and instructions - speed up, memory hierarchy, content addressable memory (CAM), data intensive applications, Computing walls - power wall, memory wall, ILP wall, technology walls - leakage, reliability, cost, CPU-Centric/ Data-Centric computing – data-centric approaches, accelerators, specialised logic, requirements, challenges	5	
3	**Computation-In Memory:** Background, computation-in-memory (CIM) model - near-memory processing (NMP), processing-In-memory (PIM), computing in-memory computing (IMC)	3	
4	**SRAM-Based In-Memory Computing:** SRAM - cell design, memory array, read, write, SRAM-IMC based logic operations, arithmetic operations, sorting-searching and other operations, potential applications, challenges	6	
5	**NMP and IMC in DRAM:** DRAM organization - cell, memory array, charge sharing, read, write, Computing in DRAM - row copy, bulk initialization, bitwise operations, potential applications, challenges	6	
6	**Memristor based IMC:** Non-Volatile memories, memristor basics, memristor crossbar, memristor classification - VVM, VVH, RVM, etc, stateful logic design - IMPLY, MAGIC, IMC - sorting, searching, challenges	6	
7	**Akers for IMC:** Akers cell, array, stripping to calculate output of Akers array, 1/0-path, realization of logic functions, sorting, searching	4	
8	**Cellular Automata for IMC:** Cellular Automata (CA) basics - cell, neighborhood, dimension, boundary condition, state transitions, reversibility - irreversibility, IMC - realizations of logic functions, shortest path and traveling salesman problems, sorting etc.	5	

Course Outcome	To understand the limitations of traditional von Neumann architectures and the motivation for in-memory computing paradigms. To analyze different in-memory computing models including PIM, NMP, and IMC, and evaluate their operational principles and potential benefits. To design and implement in-memory computation techniques using various memory technologies such as SRAM, DRAM, and memristors. To explore specialized hardware architectures like Akers arrays and Cellular Automata for solving computational problems through in-memory approaches.

Learning Resources	1. In-Memory Computing Hardware Accelerators for Data-Intensive Applications - Baker Mohammad and Yasmin Halawani, Springer, 2024 2. Shortest Path Solvers. From Software to Wetware - Andrew Adamatzky, Springer, 2018

Course Code	CS5230N	**Course Name**	Reconfigurable Computing	**Course Category**	PSE2	L	T	P
						3	0	0

Pre-requisite Courses	*Digital Logic Design, Computer Organization and Computer Architecture*	**Co-requisite Courses**		**Progressive Courses**	*Embedded Systems*
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	1. To provide foundational knowledge of reconfigurable hardware. 2. To explore the differences and trade-offs between reconfigurable systems and traditional computing platforms (CPU, GPU, ASIC). 3. To introduce hardware description languages (HDLs) such as Verilog for modeling and implementing digital systems. 4. To develop competence in FPGA-based system design, including logic synthesis, placement, and routing using industry-standard tools. 5. To highlight applications and case studies where reconfigurable computing offers performance, flexibility, or energy efficiency advantages.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Device Architecture:** General Purpose Computing Vs Reconfigurable Computing, Simple Programmable Logic Devices (SPLD), Complex Programmable Logic Devices (CPLD), FPGAs Device Architecture, Fine-grained architectures, Coarse-grained architectures	10	Will learn the differences between general-purpose and reconfigurable computing, and gain knowledge of SPLDs, CPLDs, FPGA architectures, and the design principles of fine-grained and coarse-grained reconfigurable systems.
2	**Reconfigurable Platforms Design Cycle:** The Design Flow, Technology Mapping, FPGA Placement and Routing, Configuration Bitstream Generation, Case Studies with Appropriate Tools.	8	Able to understand the reconfigurable platform design cycle, including design flow, technology mapping, FPGA placement and routing, bitstream generation, and will gain practical experience through case studies using appropriate design tools.
3	**Design methodology and techniques:** Top-down, modular design, Controller/controlled-component architecture, Implementation methods - traditional, MUX, ROM, Design and Implementation techniques using FPGA's	8	Will learn design methodologies such as top-down and modular design, controller/controlled-component architecture, and implementation methods
4	**Register Transfer (RT)/Logic Synthesis:** Controller/Datapath synthesis, Logic minimization	6	Will understand register transfer and logic synthesis concepts
5	**Advanced Topics:** Partial Reconfiguration, Dynamic Reconfiguration, Fault Tolerance	6	Will learn some advanced concepts of reconfigurable computing

Course Outcome	CO1: To be able to understand the architecture and internal components of FPGAs. CO2: To be able to compare the performance, power, and flexibility trade-offs between reconfigurable systems and traditional processors. CO3: To be able to apply HDLs (such as Verilog) to design and simulate digital circuits for FPGA implementation. CO4: To be able to design, synthesize, and deploy digital systems on FPGAs using EDA tools. CO5: To be able to analyze case studies where reconfigurable computing improves performance or efficiency in real-world applications.

Learning Resources	1. Christophe Bobda, "Introduction to Reconfigurable Computing – Architectures, Algorithms and Applications", Springer. 2. C. Maxfield, "The Design Warrior's Guide to FPGAs", Newnes. 3. Scott Hauck and Andre Dehon (Eds.), "Reconfigurable Computing – The Theory and Practice of FPGA Based Computation", Elsevier / Morgan Kaufmann

Course Code	CS5261N	**Course Name**	Data mining and Knowledge Discovery	**Course Category**	OE2	L	T	P
						3	0	0

Pre-requisite Courses	*Foundation in Mathematics and Knowledge in Programming*	**Co-requisite Courses**		**Progressive Courses**	
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	*The Students should able to understand the concept of data mining, various algorithms used for data mining and their applications*

Module	Syllabus	Duration (class-hour)	Module Outcome
1.	Introduction, Motivation and Functionalities of Data mining	2	Students should get the motivation, applications, and overall knowledge discovery process involved in data mining
2.	**Data Preprocessing:** Cleaning, Integration, Transformation, Reduction, Discretization, Concept hierarchy generation	6	Students should be familiar with the various data pre-processing techniques.
3.	**Association Rule mining:** Market Basket Analysis, Frequent Items, Apriori Algorithm	4	Students should be able to understand how the association rules mining identifies patterns and relationships within large datasets
4.	**Cluster Analysis:** Partition based clustering, Hierarchical Clustering, Density based clustering, Cluster Evaluation	8	Familiarization of data mining tasks (such as

5.	**Classification and Prediction:** Classification algorithms, Regression Models, Classifier Evaluation	6	classification, regression and clustering) and the corresponding algorithms and their appropriate application
6.	**Ensemble Classifier:** Bagging, Boosting, Random Forest, etc	6	How to improve prediction accuracy and robustness by combining multiple models.
7.	**Mining Text Databases:** Document preprocessing, clustering, Classification and Summarization	8	Application of a real-world problem

Course Outcome	After completion of this course, the students should be able to implement pre-processing techniques to improve the data quality for further processing. They also implement different algorithms for solving real-world problems.

Learning Resources	**References:** 1. Data Mining Concepts and Techniques by Jiawei Han, Micheline Kamber, Jian Pei 2. Introduction to Data Mining by Pang-Ning Tan. Michael Steinbach. Vipin Kumar.

Course Code	CS5262N	**Course Name**	Information Security and Cryptography	**Course Category**	OE2	L	T	P
						3	0	0

Pre-requisite Courses	*Linear Algebra,*	**Co-requisite Courses**		**Progressive Courses**	
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	To provide the knowledge and skills to design, analyze, and implement secure systems, cryptographic protocols, and security best practices, enabling them to evaluate security risks and develop secure solutions.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Overview of Information Security** Basic components, Motivation, Principles of Security, Types of attack and related threats, Policy and mechanism, Assumptions and trust, Assurance, Operational issues, Human issues, Security life-cycle.	3	Students will understand the fundamental principles, components, threats, and lifecycle of information security, including human and operational aspects.
2	**Information Security Models** Protection state, Access control matrix model, Protection state transitions, Copying/Owning/Attenuation of privilege, Take-Grant protection model.	3	Students will learn formal models of information security, including access control mechanisms and privilege management using protection models.

3	**Security Policies** Types of security policies, Role of trust, Types of access control, Confidentiality and integrity policies and models, Access control policies	3	Students will understand various security policies and access control models focusing on confidentiality, integrity, and trust management.
4	**Mathematical Background:** Shannon's Theory, Computational Complexity, Finite Fields, Number Theory, Concept of Pseudo-Random Functions, Modes of Operation, Modular Arithmetic, Handling large prime numbers, Primality Testing Techniques, Factorization Methods, Fast Exponentiation.	4	Students will gain the mathematical foundation for cryptography, including number theory, finite fields, and complexity concepts essential for secure algorithm design.

5	**Cryptography - Basic Concepts & Techniques:** Introduction, Plaintext & Cipher text, Substitution Techniques, Transposition Techniques, Encryption & Decryption, Overview of Symmetric & Asymmetric key Cryptography, Key Range & Key Size.	3	Students will understand the basic principles of cryptography, including classical techniques, encryption/decryption, and key-based cryptographic methods.
6	**Symmetric Key Algorithm:** Introduction, Algorithm types & Modes, Fiestel Cipher, Simplified-Data Encryption Standard (S- DES) algorithm, DES algorithm, Variety of DES, Advanced Encryption Standard (AES), RC5 (Rivest Cipher 5) algorithm. Key exchange problem in symmetric key cryptography, Diffie-Hellman Key exchange algorithm and its analysis.	8	Students will learn symmetric key encryption algorithms, their modes of operation, and key exchange techniques including Diffie-Hellman.

7	**Asymmetric Key Algorithm:** Introduction, Knapsack Cryptosystem, RSA and ECC algorithms, Symmetric & Asymmetric key Cryptography together.	5	Students will understand the principles of asymmetric cryptography and analyze key algorithms like RSA, ECC, and the integration of symmetric and asymmetric techniques.
8	**Message Digests and Hash Functions:** Basic concepts of Message Digest, Properties of message digest, Digest Algorithms like MD 5 algorithm, SHA-1 algorithm.	3	Students will learn the fundamentals of message digests and implement algorithms like MD5 and SHA-1.
9	**Message Authentication and Signatures:** Idea of Message Authentication Codes, Hash-based Message Authentication Codes and Digital Signature Techniques.	3	Students will understand message authentication methods and digital signature techniques for ensuring data integrity and authenticity.

10	**Cryptanalysis** Basic cryptanalysis techniques, Linear and Differential Cryptanalysis, Intractable (Hard) problems, Recent trends in Cryptanalysis.	3	Students will explore fundamental and a few advanced cryptanalysis techniques, including linear, differential methods, and current challenges in breaking cryptographic systems.
11	**Key Management and Certification Authority:** Digital Certificate, Certification Authority, Registration Authority, Steps involved in digital certificate creation, Certificate Revocation, etc.	4	Students will understand key management processes and the role of registration and certification authorities in issuing, managing, and revoking digital certificates.

Course Outcome	After studying this course the students will be able to meet the following COs. CO1: Understand the principles of information security, cryptography, and security threats. CO2: Design and develop secure systems, applying cryptographic techniques and security principles. CO3: Analyze and evaluate the security of various cryptographic protocols and systems. CO4: Implement security measures to protect against threats and vulnerabilities. CO5:Identify and assess security risks, and develop strategies for mitigation.

Learning Resources	**References / Books:** 1. "Computer Security: Art & Science", Matt Bishop, Addison Wesley. 2. "Cryptography and Network Security", William Stallings, 2nd Edition, *Pearson Education Asia.* 3. "Network Security private communication in a public world", C. Kaufman, R. Perlman and M. Speciner,*Pearson.* 4. "Cryptography & Network Security", Atul Kahate, *TMH.* 5. "Network Security Essentials: Applications and Standards", William Stallings, *Pearson.* 6. "Designing Network Security", Merike Kaeo, 2nd Edition, *Pearson Books*. 7. "Practical Unix & Internet Security", Simson Garfinkel, Gene Spafford, Alan Schwartz, 3rd Edition, *Oreilly*.

Course Code	CS5263N	**Course Name**	Database Management Systems	**Course Category**	OE2	L	T	P
						3	0	0

Pre-requisite Courses	Data Structure and Algorithms	**Co-requisite Courses**	–	**Progressive Courses**	Advanced DBMS
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	To understand the fundamental concepts of database systems, including architecture, data models, and relational databases. To develop skills in data modeling, SQL querying, normalization, and transaction management for efficient database design and operation. To apply the acquired knowledge in designing and developing database solution for basic and conventional use cases.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Database (DB), Database Management Systems, Database systems versus File systems, DB users, DB Administrators, Basic Schema of 3-schema, Architecture, data independence, integrity, consistency.	6	Ability to understand the overall domain of database management systems (DBMS) with the emphasis on the difference between database and DBMS.
2	Basic concepts of E-R model, Constraints, Cardinality constraints, Weak-entity, Subclasses and inheritance, Specialization and Generalization, case study on E-R model	6	Ability to apply informal guidelines for designing relational schemas towards improving data integrity, reducing redundancy and anomalies.
3	Data Model: Relational Model, ER to relational mapping, Relational Algebra, Query language SQL, Views, Integrity constraints, Specifying indexes, Embedded SQL.	10	Ability to present queries using relational algebra, write SQL queries for data retrieval and manipulation; create and utilize views; enforce integrity constraints; specify indexes for performance
4	Overview of Normalization and its role in Database Design using 1NF to BCNF	8	understanding the concept and benefits of normalization in relational databases

5	Issues in DBMS implementation: Security, Recovery and concurrency control, transaction management	6	· Understanding the need for concurrency control · Ability to grasp the knowledge of transaction management principles to achieve ACID properties · Confidence to apply recovery mechanisms to maintain data consistency in the face of failures.
6	Case study on Design and Implementation of real-life applications (e.g. student admission system, Banking system)	6	Ability to learn and analyze real-world applications to identify requirements, apply design principles, implement solutions and evaluate their effectiveness

Course Outcome	CO1: Ability to design entity-relationship diagrams to represent simple database application scenarios. CO2: Skill to convert entity-relationship diagrams into relational tables (database design), populate a relational database and formulate SQL queries on the data. CO3: Ability to cross check and improve the design by normalization. CO4: Knowledge of the basics of query evaluation techniques and query optimization. CO5: Ability to implement conventional applications with the acquired knowledge.

Learning Resources	**References:** 1. A Silberschatz, H Korth, S Sudarshan, Database System and Concepts, 5th Ed., McGraw-Hill 2. C. J. Date, An introduction to database systems, Addision Wesley

Course Code	CS5264N	**Course Name**	Computer Control of Industrial Processes	**Course Category**	OE2	L	T	P
						3	0	0

Pre-requisite Courses		**Co-requisite Courses**		**Progressive Courses**	
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	To understand and model industrial processes using modern control theory and state-space techniques. To design and analyze discrete control systems including stability, controllability, and observability. To implement real-time control strategies with fault tolerance, estimation, and optimal control approaches.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to modern control theory and system modelling methods	4	Getting overview
2	Discrete dynamical system representation, Z-transform and approximations	4	Foundation mathematics
3	Concept of stability, Controllability and Observability criterion	4	Appreciate system behaviours
4	Controller and observer design in the discrete domain, Concepts of estimation, prediction and smoothing	6	Ability to design algorithms
5	Processor architecture and interfacing - microcontrollers and DSP case studies	6	Ability to write programs
6	Ideas of optimal, adaptive and stochastic control	6	Appreciate state-of-the-art
7	Real time integration and synchronization	6	Understand OS concepts
8	Fault tolerance techniques and algorithms	6	Ability to do fault tolerant design

Course Outcome	CO1: Ability to define the digital inputs and outputs of any dynamical system CO2: Ability to describe and model the process mathematically CO3: Ability to interface the computer as control element CO4: Ability to study stability, controllability and observability of given system CO5: Ability to design algorithms for control and estimation of states CO6: Ability to implement computer based control system in real time

Learning Resources	**References/Books:** 1. Lamont and Houpis, Digital control system - theory, hardware, software 2. K Ogata, Discrete time control system 3. Arthur Gelb, Applied optimal estimation

Course Code	CS5265N	**Course Name**	Mobile and Pervasive Computing	**Course Category**	OE2	L	T	P
						3	0	0

Pre-requisite Courses	*(i) Computer Networks (ii) Basic Programming Courses*	**Co-requisite Courses**	*(i) Advanced Networking (ii) Internet of Things*	**Progressive Courses**	*(i) Future Generation Network Systems*
Course Offering Department		***CST***		**Data Book / Codes/Standards**	

Course Objective	-Understand the fundamentals of mobile and pervasive computing systems. -Explore the role of mobile networks and sensor technologies in pervasive environments. -Analyze context-aware systems and their applications in real-world scenarios. -Develop skills to design and implement applications for mobile and pervasive computing platforms. -Examine security, privacy, and ethical considerations in pervasive computing.

Module	Syllabus	Duration (class-hour)	Module Outcome
1	**Introduction to Mobile and Pervasive Computing:** Evolution and vision of pervasive computing; Characteristics and challenges of mobile computing; Standards and protocols in mobile computing; Applications and services in pervasive environments	4	-Understand the evolution, vision, and fundamental concepts of pervasive computing. -Explain the key characteristics, opportunities, and challenges of mobile computing systems. -Identify and describe relevant standards and protocols used in mobile computing environments. -Explore and analyze real-world applications and services enabled by pervasive computing technologies.

2	**Mobile Network Technologies:** Overview of earlier cellular networks: 2G GSM, 3G, 4G-LTE; 5G/6G wireless network: System architecture, NG-RAN, NG-core, Deployment options, Network virtualization and Slicing, Software Defined Networks; Wireless network technologies: Wi-Fi, Bluetooth, Zigbee; Mobile IP and mobility management;	16	-Understand the evolution and architecture of mobile networks, including 2G (GSM), 3G, 4G-LTE, and emerging 5G/6G technologies. -Explain the components and architecture of next-generation networks including NG-RAN, NG-core, and their deployment options. -Analyze advanced concepts such as network virtualization, network slicing, and Software Defined Networking (SDN) in mobile networks. -Compare wireless technologies such as Wi-Fi, Bluetooth, and Zigbee, and describe their applications. -Describe the principles of Mobile IP and techniques for mobility

			management in mobile networks.
3	**Sensor Networks and Context-Aware Computing:** Architecture of wireless sensor networks; Data dissemination and aggregation techniques; Context-aware computing models and middleware; Service discovery; Location-aware services and positioning techniques; Wearable devices: architectures and applications; Energy harvesting	12	-Understand the architecture and components WSNs. -Explain techniques for data dissemination and aggregation in sensor networks. -Describe models and middleware for context-aware computing, including service discovery mechanisms. -Analyze location-aware services and positioning techniques in pervasive environments. -Explore the architecture and applications of wearable devices, along with energy harvesting methods for sustainable operation.

4	**Application Development and Emerging Trends:** Overview on mobile application development frameworks (e.g., Android, iOS); Security and privacy in mobile applications; Emerging trends: IoT integration, edge computing, and AI in pervasive systems; Case study on Smart homes/Healthcare/Intelligent transportation systems.	10	-Understand the fundamentals of mobile application development frameworks, such as Android and iOS. -Analyze key considerations in security and privacy for mobile and pervasive applications. -Explore emerging trends including IoT integration, edge computing, and the use of AI in pervasive environments. -Evaluate and present real-world case studies in domains like smart homes, healthcare, and intelligent transportation systems.

Course Outcome	CO1: Differentiate between mobile computing and pervasive computing paradigms. CO2: Design architectures for context-aware and location-aware applications. CO3: Implement mobile applications using modern development frameworks. CO4: Evaluate security and privacy challenges in pervasive environments. CO5: Integrate sensor networks and wearable devices into computing solutions.

Learning Resources	Jochen Burkhardt et al., Pervasive Computing: Technology and Architecture of Mobile Internet Applications, Addison-Wesley, 2002. Frank Adelstein et al., Fundamentals of Mobile and Pervasive Computing, McGraw-Hill, 2005. William C. Y. Lee, Mobile Cellular Telecommunications, Tata McGrawHill William Stallings, Wireless Communications and Networks, Pearson S. DasBit and B.K. Sikdar, Mobile Computing, PHI