

**ANNA UNIVERSITY, CHENNAI**  
**NON- AUTONOMOUS COLLEGES**  
**AFFILIATED TO ANNA UNIVERSITY**  
**M.E. COMPUTER SCIENCE AND ENGINEERING (AI & ML)**  
**REGULATIONS 2025**

**PROGRAMME OUTCOMES (POs):**

<b>PO</b>	<b>Programme Outcomes</b>
<b>PO1</b>	An ability to independently carry out research /investigation and development work to solve practical problems
<b>PO2</b>	An ability to write and present a substantial technical report/document.
<b>PO3</b>	Students should be able to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

**PROGRAMME SPECIFIC OUTCOMES:**

**PSO1:** Intelligent Systems Design: Design, develop, and deploy intelligent computing systems by leveraging artificial intelligence and machine learning techniques to solve real-world problems in domains such as healthcare, finance, and smart systems.

**PSO2:** Research, Innovation, and Ethical AI Practice: Conduct cutting-edge research in AI and ML, develop innovative solutions, and ensure responsible and ethical use of AI technologies considering fairness, transparency, and societal impact.



**ANNA UNIVERSITY, CHENNAI**

**POSTGRADUATE CURRICULUM (NON-AUTONOMOUS AFFILIATED INSTITUTIONS)**

**Programme:** M.E. Computer Science and Engineering (AI & ML)

Regulations: 2025

**Abbreviations:**

**BS** – Basic Science (Mathematics,)

**L** – Laboratory Course

**ES** – Engineering Science (Programme Core (**PC**),  
Programme Elective (**PE**))

**T** – Theory

**SD** – Skill Development

**LIT** – Laboratory Integrated Theory

**SL** – Self Learning

**PW** – Project Work

**TCP** – Total Contact Period(s)

Semester I									
S. No.	Course Code	Course Title	Type	Periods per week			TCP	Credits	Category
				L	T	P			
1.	MA25C07	Advanced Mathematical Methods (CSIE)	T	3	1	0	4	4	BS
2.	CP25C01	Advanced Data Structures and Algorithms	LIT	3	0	4	7	5	ES (PC)
3.	ML25101	Image and Video Processing	T	3	0	0	3	3	ES (PC)
4.	CP25C03	Advanced Operating Systems	T	3	0	0	3	3	ES (PC)
5.	CP25C04	Advanced Compiler Design	T	3	0	0	3	3	ES (PC)
6.	ML25102	Technical Seminar	-	0	0	2	2	1	SD
<b>Total Credits</b>							<b>22</b>	<b>19</b>	

Semester II									
S. No.	Course Code	Course Title	Type	Periods per week			TCP	Credits	Category
				L	T	P			
1.	ML25201	Soft Computing and Evolutionary AI	LIT	3	0	2	5	4	ES (PC)
2.	IF25C01	Advanced Software Testing Techniques	LIT	3	0	2	5	4	ES (PC)
3.	CP25C06	Cloud and Big Data Analytics	T	3	0	0	3	3	ES (PC)
4.	CP25C07	Quantum Computing	T	2	0	0	2	2	ES (PC)
5.		Programme Elective I	T	3	0	0	3	3	ES (PE)
6.		Industry Oriented Course I	-	1	0	0	1	1	SD
7.	ML25202	Industrial Training	-	-	-	-	-	2	SD
8.		Self-Learning Course	-	-	-	-	-	1	-
<b>Total Credits</b>							<b>19</b>	<b>20</b>	

Semester III									
S. No.	Course Code	Course Title	Type	Periods per week			TCP	Credits	Category
				L	T	P			
1.		Programme Elective II	T	3	0	0	3	3	ES (PE)
2.		Programme Elective III	T	3	0	0	3	3	ES (PE)
3.		Programme Elective IV	T	3	0	0	3	3	ES (PE)
4.		Programme Elective V	T	3	0	0	3	3	ES (PE)
5.		Industry Oriented Course II	-	1	0	0	1	1	SD
6.	ML25301	Project Work I	-	0	0	12	12	6	SD
<b>Total Credits</b>							<b>25</b>	<b>19</b>	

Semester IV									
S. No.	Course Code	Course Title	Type	Periods per week			TCP	Credits	Category
				L	T	P			
1.	ML25401	Project Work II	-	0	0	24	24	12	SD
<b>Total Credits</b>							<b>24</b>	<b>12</b>	

PROGRAMME ELECTIVE COURSES (PE)								
S. No.	Course Code	Course Title	Type	Periods per week			TCP	Credits
				L	T	P		
1.	CP25C08	Advanced Software Testing and Quality Assurance	T	3	0	0	3	3
2.	CP25C09	Agile Methodologies	T	3	0	0	3	3
3.	IF25C05	Generative AI	T	3	0	0	3	3
4.	ML25001	AutoML	T	3	0	0	3	3
5.	CP25C11	Advanced Deep Learning and Neural Networks	T	3	0	0	3	3
6.	CP25C13	Quantum Machine Learning	T	3	0	0	3	3
7.	CP25C14	AI in IoT	T	3	0	0	3	3
8.	ML25002	Equitable and Responsible AI	T	3	0	0	3	3
9.	CP25C16	Advanced Large Language Models	T	3	0	0	3	3
10.	CP25C17	Edge and Fog Computing	T	3	0	0	3	3
11.	CP25C18	Green Computing and Sustainability	T	3	0	0	3	3
12.	IF25C03	Devops and Micro Services	T	3	0	0	3	3
13.	CP25C19	Cognitive Computing	T	3	0	0	3	3
14.	CP25C20	Agentic AI	T	3	0	0	3	3
15.	CP25C21	Mixed Reality	T	3	0	0	3	3
16.	BD25C02	Exploratory Data Analysis	T	3	0	0	3	3
17.	BD25C01	Prompt Engineering	T	3	0	0	3	3
18.	CP25C23	Human-Centered AI	T	3	0	0	3	3
19.	CP25C24	Vibe Coding	T	3	0	0	3	3
20.	CP25C25	Federated Learning	T	3	0	0	3	3

MA25C07	Advanced Mathematical Methods (CSIE)	L	T	P	C
		3	1	0	4
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Develop an in-depth understanding of advanced concepts in linear algebra, multivariate analysis, and number theory for computer science applications.</li> <li>• Apply mathematical tools such as eigenvalue decomposition, SVD, and multivariate statistical methods to real-world computing and data-driven problems.</li> <li>• Analyze and implement number-theoretic techniques for cryptography, security, and algorithmic problem-solving in computer science.</li> </ul>					
<p><b>Linear Algebra:</b> Vector spaces, norms, Inner Products, Eigenvalues using QR transformations, QR factorization, generalized eigenvectors, Canonical forms, singular value decomposition and applications, pseudo inverse, least square approximations.</p>					
<p><b>Multivariate Analysis:</b> Random vectors and matrices, Mean vectors and covariance matrices, Multivariate normal density and its properties, Principal components, Population principal components, Principal components from standardized variables.</p>					
<p><b>Elementary Number Theory:</b> The division algorithm, Divisibility and the Euclidean algorithm, The fundamental theorem of arithmetic, Modular arithmetic and basic properties of congruences; Principles of mathematical induction and well ordering principle. Primality Testing algorithms, Chinese Remainder Theorem, Quadratic Congruence.</p>					
<p><b>Advanced Number Theory:</b> Advanced Number Theory, Primality Testing algorithms, Chinese Remainder Theorem, Quadratic Congruence, Discrete Logarithm, Factorization Methods, Side Channel Attacks, Shannon Theory, Perfect Secrecy, Semantic Security.</p>					
<p><b>Weightage:</b> Continuous Assessment: 40%, End Semester Examinations: 60%.</p>					
<p><b>Assessment Methodology:</b> Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20).</p>					
<p><b>References:</b></p> <ol style="list-style-type: none"> <li>1. Gilbert Strang, Linear Algebra and Its Applications, Cengage Learning.</li> <li>2. Richard A. Johnson &amp; Dean W. Wichern, Applied Multivariate Statistical Analysis, Pearson.</li> <li>3. Neal Koblitz, A Course in Number Theory and Cryptography, Springer.</li> <li>4. Victor Shoup, A Computational Introduction to Number Theory and Algebra, Cambridge University Press.</li> </ol>					
<p><b>E-resources:</b></p> <ol style="list-style-type: none"> <li>1. <a href="https://ocw.mit.edu/courses/18-06-linear-algebra">https://ocw.mit.edu/courses/18-06-linear-algebra</a></li> <li>2. <a href="https://nptel.ac.in/courses/111105041">https://nptel.ac.in/courses/111105041</a></li> <li>3. <a href="https://crypto.stanford.edu/pbc/notes/numbertheory">https://crypto.stanford.edu/pbc/notes/numbertheory</a></li> </ol>					

CP25C01	Advanced Data Structures and Algorithms	L	T	P	C
		3	0	4	5
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• To explore advanced linear, tree, and graph data structures and their applications.</li> <li>• To design efficient algorithms using appropriate algorithmic paradigms.</li> <li>• To evaluate computational complexity and identify tractable vs. intractable problems.</li> </ul>					
<p><b>Linear Data Structures and Memory Optimization:</b> Advanced arrays: Sparse arrays, dynamic arrays, cache-aware structures, Linked lists: Skip lists, unrolled linked lists, XOR linked lists, Stacks and Queues: Priority queues, double-ended queues, circular buffers, Hashing: Perfect hashing, cuckoo hashing, extendible hashing.</p> <p><b>Practical:</b></p> <ul style="list-style-type: none"> <li>• Implement skip lists and measure performance compared with balanced BST.</li> <li>• Experiment with cache-aware data structures and analyze memory utilization.</li> </ul>					
<p><b>Advanced Tree Data Structures:</b> Balanced Trees: AVL, Red-Black Trees, Splay Trees, Treaps, Multi-way Trees: B-Trees, B+ Trees, R-Trees, Segment Trees, Fenwick Trees, Suffix Trees and Tries for string processing, Applications in indexing, text retrieval, computational geometry.</p> <p><b>Practical:</b></p> <ul style="list-style-type: none"> <li>• Implement B+ tree for database indexing use-case.</li> <li>• Design a suffix tree-based algorithm for DNA sequence matching.</li> </ul>					
<p><b>Graph Data Structures and Algorithms:</b> Representation: Adjacency list/matrix, incidence matrix, compressed storage, Traversals: DFS, BFS with applications, Shortest Path Algorithms: Dijkstra, Bellman-Ford, Floyd-Warshall, Johnson's algorithm, Minimum Spanning Trees: Prim's, Kruskal's, Borůvka's algorithm, Network Flow Algorithms: Ford-Fulkerson, Edmonds-Karp, Push-Relabel.</p> <p><b>Practical:</b></p> <ul style="list-style-type: none"> <li>• Implement Johnson's algorithm for sparse graph shortest paths.</li> <li>• Demonstration of Maximum flow in traffic or network routing simulation.</li> </ul>					
<p><b>Algorithm Design and Paradigms:</b> Divide and Conquer: Karatsuba's multiplication, Strassen's algorithm, Greedy Methods: Huffman coding, interval scheduling, set cover approximation, Dynamic Programming: Matrix chain multiplication, Floyd-Warshall, knapsack variants, Backtracking and Branch-and-Bound, Randomized Algorithms and Probabilistic Analysis.</p> <p><b>Practical:</b></p> <ul style="list-style-type: none"> <li>• Implement Strassen's algorithm and compare with naive matrix multiplication.</li> <li>• Develop a randomized algorithm for primality testing (Miller–Rabin).</li> </ul>					

**Computational Complexity and Approximation Algorithms:** Complexity Classes: P, NP, NP-Complete, NP-Hard, Reductions: Polynomial-time reductions, Cook-Levin theorem (overview), Approximation Algorithms: Vertex cover, set cover, TSP, k-center problem, Heuristic Algorithms: Local search, simulated annealing, genetic algorithms.

**Practical:**

- Implement approximation algorithm for vertex cover.
- Complexity analysis of a chosen NP-hard problem and implement a heuristic.

**Advanced Topics and Emerging Trends:** Randomized Algorithms – Monte Carlo Algorithms, Parallel and Distributed Algorithms – PRAM Model, Divide and Conquer in Parallel, Load Balancing, Streaming Algorithms – Data Stream Models, Sketching and Sampling, Frequency Moments, Advanced String Matching – Suffix Trees, Suffix Arrays, Pattern Matching in Linear Time.

**Practical:**

- Implement randomized and streaming algorithms on real-world datasets.
- Design of parallel and distributed algorithms.

**Weightage:** Continuous Assessment: 50%, End Semester Examinations: 50%

**Assessment Methodology:** Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20)

**References:**

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms. MIT Press.
2. La Rocca, M. (2021). Advanced algorithms and data structures. Manning Publications.
3. Goodrich, M. T., Tamassia, R., & Mount, D. M. (2011). Data structures and algorithms in C++. John Wiley & Sons, Inc.
4. Weiss, M. A. (2014). Data structures and algorithm analysis in C++. Pearson Education.
5. Drozdek, A. (2013). Data structures and algorithms in C++. Cengage Publications.

**E-resources:**

1. <https://www.theiotacademy.co/blog/data-structures-and-algorithms-in-c/>
2. [https://github.com/afrid18/Data\\_structures\\_and\\_algorithms\\_in\\_cpp](https://github.com/afrid18/Data_structures_and_algorithms_in_cpp)
3. <https://www.udemy.com/course/introduction-to-algorithms-and-data-structures-in-c/?srsltid=AfmBOorEZlkgV7QzaEh6lqzAaKLjC-lpFU1NGgWFOHMLhOos-uDVKjCK>

	Description of CO	PO	PSO
CO1	Describe data structures and implement algorithmic solutions for complex computational problems.	--	--
CO2	Analyze the time complexity and efficiency of algorithms for various computing problems.	PO1(3)	PSO1(3)

	<b>Description of CO</b>	<b>PO</b>	<b>PSO</b>
CO3	Evaluate algorithmic techniques and data structures to determine their suitability for different applications.	PO3(2)	PSO2(2)
CO4	Design optimized solutions for real-world problems using appropriate algorithms and data structures.	PO2(1)	PSO1(3)

ML25101	Image and Video Processing	L	T	P	C
		3	0	0	3
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>To provide knowledge about digital image and video processing techniques.</li> <li>To enable students to apply computational methods extracting meaningful information from images and video sequences.</li> <li>To develop skills in designing and implementing image and video processing applications for real-world domains.</li> </ul>					
<p><b>Image Processing and Transform Techniques:</b> Image formation and Image Processing, Radiometric and geometric models, Advanced transforms, DCT- KLT- Wavelet Transform - Multispectral and hyperspectral imaging basics</p> <p><b>Activity:</b> Decompose image and visualize approximation coefficients.</p>					
<p><b>Image Enhancement and Filtering Techniques:</b> Adaptive contrast enhancement and histogram specification, Edge enhancement and noise suppression using anisotropic diffusion, Frequency filtering with emphasis on high-frequency detail preservation.</p> <p><b>Activity:</b> Demonstration of frequency domain enhancement techniques.</p>					
<p><b>Image Restoration and Morphological Analysis:</b> Image Analysis, Blind deconvolution and constrained least squares filtering, Mathematical morphology for pattern recognition, Morphological reconstruction and geodesic transforms</p> <p><b>Activity:</b> Restore degraded images using constrained optimization.</p>					
<p><b>Advanced Image Compression:</b> Context-based and predictive coding techniques - SPIHT, EZW and wavelet-based compression schemes, JPEG2000 standard and ROI-based encoding</p> <p><b>Activity:</b> Visualization of Compression techniques for JPEG image.</p>					
<p><b>Video Processing:</b> Motion Estimation, D motion models and affine transformations - Optical flow and gradient-based motion estimation -Sub-pixel motion estimation and compensation</p> <p><b>Activity:</b> Analyze Surveillance Video Dataset (provided offline) and annotate observed motion manually</p>					
<p><b>Video Compression and Analytics:</b> Scalable video coding and HEVC/H.265 standard - Video content analysis and summarization techniques - Real-time streaming protocols and latency handling</p> <p><b>Activity:</b> Evaluate compression and quality tradeoffs.</p>					
<p><b>Weightage:</b> Continuous Assessment: 40%, End Semester Examinations: 60%</p>					
<p><b>Assessment Methodology:</b> Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20).</p>					

**References:**

1. Gonzalez, R. C., & Woods, R. E. (2018). Digital image processing. Pearson Education.
2. Bovik, A. C. (2009). The essential guide to video processing. Academic Press.
3. Tekalp, M. A. (2015). Digital video processing. Prentice Hall.
4. Jain, A. K. (1989). Fundamentals of digital image processing. Prentice Hall.
5. Dey, S. (2022). Hands-on image processing with Python. Packt Publishing.

**E-Resources:**

1. Prof. P.K. Biswas, "Digital Image Processing", IIT Kharagpur, NPTEL - <https://nptel.ac.in/courses/117105079>
2. MIT OCW - Advanced Image Processing: <https://ocw.mit.edu>
3. MathWorks Documentation: <https://www.mathworks.com/help/images/>
4. Coursera - Advanced Image and Video Processing: <https://www.coursera.org>  
FFmpeg Official Documentation: <https://ffmpeg.org/documentation.html>

	<b>Description of CO</b>	<b>PO</b>	<b>PSO</b>
CO1	Explain image processing techniques to improve image quality in diverse applications.	--	--
CO2	Analyze image and video compression algorithms, to evaluate their efficiency and quality trade-offs.	PO1(3)	PSO1(3)
CO3	Design and implement algorithms for motion estimation, object tracking and video stabilization.	PO3(2)	PSO2(2)
CO4	Evaluate multispectral/hyperspectral imaging and scalable video coding.	PO2(1)	PSO1(3)

CP25C03	Advanced Operating Systems	L	T	P	C
		3	0	0	3
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>To analyze the architectures and design issues of advanced operating systems.</li> <li>To develop the model for process synchronization and recovery in complex environments.</li> <li>To evaluate algorithms for distributed coordination, resource management, fault tolerance, and security.</li> </ul>					
<p><b>Advanced Process and Thread Management:</b> Multithreading models, thread pools, context switching, Synchronization issues and solutions: semaphores, monitors, lock-free data structures, CPU scheduling in multi-core systems</p> <p><b>Activity:</b> CPU scheduler simulation for multicore systems.</p>					
<p><b>Memory and Resource Management in Modern OS:</b> Virtual memory, demand paging, page replacement policies-Huge pages, NUMA-aware memory management-Resource allocation in cloud-native environments</p> <p><b>Activity:</b> Simulate demand paging and page replacement algorithms.</p>					
<p><b>Virtualization and Containerization:</b> Hypervisors (Type I &amp; II), KVM, QEMU, Xen-Containers: Docker, LXC, systemd-nspawn-OS-level virtualization and namespaces</p> <p><b>Activity:</b> Deploy and configure Docker containers with various images.</p>					
<p><b>Distributed Operating Systems and File Systems:</b> Distributed scheduling, communication, and synchronization-Distributed file systems: NFS, GFS, HDFS-Transparency issues and fault tolerance</p> <p><b>Activity:</b> Simulate distributed process synchronization.</p>					
<p><b>Security and Trust in Operating Systems:</b> Access control models: DAC, MAC, RBAC-OS hardening techniques, sandboxing, SELinux, AppArmor-Secure boot, rootkit detection, trusted execution environments</p> <p><b>Activity:</b> Implement Role-Based Access Control (RBAC) using Linux user and group permissions.</p>					
<p><b>Real-Time and Embedded Operating Systems:</b> Real-time scheduling algorithms (EDF, RM)-POSIX RT extensions, RTOS architecture-TinyOS, FreeRTOS case studies</p> <p><b>Activity:</b> Analyze FreeRTOS task scheduling behavior.</p>					
<p><b>Edge and Cloud OS: Future Paradigms:</b> Serverless OS, unikernels, lightweight OS for edge computing-Mobile OS internals (Android, iOS)-OS for quantum and neuromorphic computing (intro)</p> <p><b>Activity:</b> Analyze Android's system architecture using emulator tools.</p>					
<p><b>Weightage:</b> Continuous Assessment: 40%, End Semester Examinations: 60%</p>					

**Assessment Methodology:** Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20).

**References:**

1. Tanenbaum, A. S., & Bos, H. (2023). Modern operating systems. Pearson.
2. Buyya, R., et al. (2022). Content delivery networks and emerging operating systems. Springer.
3. Silberschatz, A., Galvin, P. B., & Gagne, G. (2022). Operating system concepts. Wiley.
4. Anderson, T., & Dahlin, M. (2021). Operating systems: Principles and practice. Recursive Books.
5. Arpaci-Dusseau, R. H., & Arpaci-Dusseau, A. C. (2020). Operating systems: Three easy pieces.

**E-Resources:**

1. Prof. Smruti Ranjan Sarangi, "Advanced Distributed Systems", IIT Delhi, NPTEL, [https://onlinecourses.nptel.ac.in/noc22\\_cs80/preview](https://onlinecourses.nptel.ac.in/noc22_cs80/preview)
2. Prof. Rajiv Misra, "Cloud Computing and Distributed Systems", IIT Patna, NPTEL, <https://nptel.ac.in/courses/106104182>

	Description of CO	PO	PSO
CO1	Describe operating system concepts for memory and resource management.	--	--
CO2	Analyse virtualization and distributed OS mechanisms for scalability and performance.	PO1(3)	PSO1(3)
CO3	Evaluate OS security and resource handling strategies in diverse environments.	PO3(2)	PSO2(2)
CO4	Design innovative OS solutions using modern tools and techniques.	PO2(1)	PSO1(3)

CP25C04	Advanced Compiler Design	L	T	P	C
		3	0	0	3
<p><b>Course Objective:</b></p> <ul style="list-style-type: none"> <li>• To analyze the theory and principles of modern compiler design and advanced optimization techniques.</li> <li>• To design and implement efficient front-end and back-end compiler components for programming languages.</li> <li>• To evaluate code optimization strategies and runtime environment management in contemporary architectures.</li> </ul>					
<p><b>Intermediate Representations and Control Flow Analysis:</b> Static single assignment (SSA) form- Context-Free Grammar (CFG) construction-dominance relations-Intermediate Representation (IR) design for functional and imperative languages-Static single assignment and def-use chains</p> <p><b>Activities:</b></p> <ol style="list-style-type: none"> <li>1. Convert source code to SSA form using LLVM IR.</li> <li>2. Visualize control flow graphs from SSA using LLVM tools.</li> </ol>					
<p><b>Program Analysis and Transformations:</b> Data flow analysis- live variable analysis-reaching definitions-Alias analysis and dependence analysis-Loop optimizations and transformations</p> <p><b>Activities:</b></p> <ol style="list-style-type: none"> <li>1. Perform loop unrolling and strength reduction.</li> <li>2. Conduct live variable analysis and visualize data flow graphs.</li> </ol>					
<p><b>Advanced Optimizations and Polyhedral Compilation:</b> Polyhedral model for loop nests-Tiling, skewing, fusion, and vectorization-Profile-guided and feedback-directed optimizations</p> <p><b>Activities:</b></p> <ol style="list-style-type: none"> <li>1. Implement loop tiling and loop skewing on a matrix multiplication program.</li> <li>2. Analyze the effect on loop-intensive code with LLVM optimization flags.</li> </ol>					
<p><b>Just-in-Time (JIT) and Runtime Compilation:</b> JIT compilation models: tracing, method-based-GraalVM architecture, Java HotSpot internals-LLVM JIT and dynamic language support</p> <p><b>Activities:</b></p> <ol style="list-style-type: none"> <li>1. Develop a basic JIT-enabled interpreter with LLVM or GraalVM.</li> <li>2. Implement dynamic dispatch using LLVM JIT API.</li> </ol>					

**Machine Learning in Compiler Design:** ML for phase ordering, auto-tuning, and IR prediction-Reinforcement learning for optimization passes-Dataset creation and benchmarking for compiler ML

**Activities:**

1. Train an ML model to predict optimization passes.
2. Use reinforcement learning for pass selection in toy compiler.

**Domain-Specific Languages (DSLs) and Compiler Extensions:** Designing DSLs for AI/ML, DSP, graphics-Code generation for custom accelerators-Integration with TensorFlow XLA and Halide

**Activities:**

1. Design and test a simple DSL grammar using ANTLR.
2. Integrate a DSL with TensorFlow XLA or Halide.

**Security, Verification, and Future Trends:** Secure compilation and type-safe intermediate representations-Compiler fuzzing and formal verification (e.g., CompCert)-Quantum compilers, multi-target compilers, and neuromorphic systems

**Activities:**

1. Use CompCert to verify compilation of simple programs.
2. Apply compiler fuzzing using tools like libFuzzer.

**Weightage:** Continuous Assessment: 40%, End Semester Examinations: 60%

**Assessment Methodology:** Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20).

**References:**

1. Cooper, K. D., & Torczon, L. (2023). Engineering a compiler. Morgan Kaufmann.
2. Grune, D., Bal, H. E., Jacobs, C. J. H., & Langendoen, K. G. (2012). Modern compiler design (2nd ed.). Springer.
3. Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2006). Compilers: Principles, techniques, and tools (2nd ed.). Pearson.
4. Völter, M. (2013). DSL engineering: Designing, implementing and using domain-specific languages. dslbook.org.
5. Sarda, S., & Pandey, M. (2015). LLVM essentials. Packt Publishing.

**E-Resources:**

1. Prof. AmeyKarkare, IIT Kanpur, "Advanced Compiler Optimizations"  
Link: <https://www.cse.iitk.ac.in/users/karkare/Courses/cs738/>
2. Prof. Santanu Chattopadhyay, "Compiler Design", IIT Kharagpur  
Link:" [https://onlinecourses.nptel.ac.in/noc21\\_cs07/preview](https://onlinecourses.nptel.ac.in/noc21_cs07/preview)"

	<b>Description of CO</b>	<b>PO</b>	<b>PSO</b>
CO1	Explain intermediate control flow techniques in compiler design.	--	--
CO2	Apply program analysis techniques and advanced optimizations for design of compilers.	PO1(3)	PSO1(3)
CO3	Develop compiler features and machine learning techniques for optimization.	PO3(2)	PSO2(2)
CO4	Evaluate secure compilation strategies for quantum and multi-target compilation.	PO2(1)	PSO1(3)

<b>ML25201</b>	<b>Soft Computing and Evolutionary AI</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		3	0	2	4
<p><b>Course Objective:</b></p> <ul style="list-style-type: none"> <li>• To provide fundamental knowledge of soft computing techniques and their applications.</li> <li>• To understand neural networks, fuzzy logic, and genetic algorithms in the context of intelligent systems.</li> <li>• To explore hybrid intelligent systems and evolutionary AI models.</li> </ul>					
<p><b>Basics of Soft Computing:</b> Definition – Hard computing vs. soft computing – Components of soft computing – Applications in engineering – Intelligent systems overview – Need for tolerance and approximation.</p>					
<p><b>Practical</b></p> <ol style="list-style-type: none"> <li>1. Design a model for implementing soft computing need in Education.</li> <li>2. Implement the need of tolerance in Decision making system</li> </ol>					

**Artificial Neural Networks:** Biological neuron model – Perceptron – Backpropagation algorithm – Multi-layer feedforward networks – Radial Basis Function networks – Applications in classification and prediction.

**Practical**

1. Implement Perceptron and Multi-layer Feedforward Neural Networks  
– Apply on binary/multi-class classification tasks (e.g., Iris dataset)
2. Train a Neural Network Using Backpropagation Algorithm  
– Solve XOR problem or perform handwritten digit recognition (MNIST)

**Fuzzy Logic Systems:** Fuzzy sets – Membership functions – Fuzzy rules and inference – Fuzzy reasoning – Defuzzification – Fuzzy logic controllers – Applications in control and decision systems.

**Practical**

1. Design a Fuzzy Logic Controller for a Real-time System  
– Example: Fan speed or temperature control using fuzzy inference
2. Develop Membership Functions and Rule Base for Fuzzy Reasoning  
– Application: Decision system for travel or risk assessment

**Genetic Algorithms and Evolutionary Computing:** Introduction to evolutionary algorithms – Selection, crossover, mutation – GA for optimization – Real-coded and binary-coded GAs – Constraint handling – Applications in engineering design and image analysis.

**Practical**

1. Apply Genetic Algorithm for Function Optimization  
– Minimize benchmark functions like Sphere or Rastrigin
2. Solve Scheduling or Knapsack Problems Using Genetic Algorithms  
– Use real-coded or binary-coded GA representations

**Hybrid Intelligent Systems:** Neuro-fuzzy systems – Genetic-fuzzy systems – Adaptive neuro-fuzzy inference system (ANFIS) – Co-evolutionary systems – Hybrid optimization strategies.

**Practical**

1. Develop and Train an Adaptive Neuro-Fuzzy Inference System (ANFIS)  
– Predict outcomes based on given datasets
2. Compare Swarm Algorithms (PSO, ACO, ABC) on Benchmark Functions  
– Analyze convergence speed and accuracy

**Swarm Intelligence and Bio-Inspired Algorithms:** Particle Swarm Optimization (PSO) – Ant Colony Optimization (ACO) – Artificial Bee Colony (ABC) – Differential Evolution (DE) – Applications in medical image processing and wireless networks.

**Practical**

1. Implement Ant Colony optimization for Optimum path prediction  
– Example: Best path prediction for Travelling Salesman Problem
2. Implement Differential Evolution (DE) for Nonlinear Optimization  
– Example: DE for engineering design or curve fitting

**Evolutionary Deep Learning and Current Trends:** Deep neural networks with evolutionary optimization – AutoML and Neural Architecture Search (NAS) – Applications in generative AI – Recent research trends in evolutionary AI – Explainable AI (XAI) in soft computing

**Practical**

1. Build an Evolutionary Deep Neural Network Model – Use Genetic Algorithm to evolve architecture or weights (AutoML-like task)
2. Build an application for Explainable AI

**Weightage:** Continuous Assessment: 40%, End Semester Examinations: 60%

**Assessment Methodology:** Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20).

**References:**

1. Q. Yang, Y. Liu, T. Chen, and Y. Tong, *Federated Learning*, PHI Learning Pvt. Ltd., 2021.
2. M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*, 2nd ed., Pearson Education, 2020.
3. K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, Pearson Education, 2017.
4. C. M. Bishop, *Pattern Recognition and Machine Learning*, Oxford University Press, 2016.
5. T. Mitchell, *Machine Learning*, Tata McGraw-Hill Education, 2017.

**E-Resources**

- Introduction To Soft Computing, IIT Kharagpur Prof. Debasis Samanta <https://nptel.ac.in/courses/106105173>
- Artificial Intelligence: Concepts and Techniques By Prof. V. Susheela Devi IISc Bangalore [https://onlinecourses.nptel.ac.in/noc25\\_cs159/preview](https://onlinecourses.nptel.ac.in/noc25_cs159/preview)

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, components, and problem-solving approaches of soft computing and evolutionary artificial intelligence techniques.	--	--
CO2	<b>Analyze</b> soft computing models such as fuzzy systems, neural networks, and evolutionary algorithms to understand their behavior and performance in uncertain environments.	PO1 (3)	PSO1 (3)
CO3	<b>Evaluate</b> soft computing and evolutionary AI techniques to assess their effectiveness in optimization, learning, and decision-making problems.	PO3 (2)	PSO2 (2)
CO4	<b>Design</b> intelligent systems using appropriate soft computing and evolutionary AI methods to solve complex real-world problems.	PO2 (1)	PSO1 (3)

IF25C01	Advanced Software Testing Techniques	L	T	P	C
		3	0	2	4

**Course Objective:**

To provide students with a deep understanding of modern and advanced software testing techniques, including exploratory testing, test heuristics, risk analysis, and real-world defect detection strategies. This course emphasizes hands-on problem-solving and designing robust testing strategies.

**Foundations and Testing Thinking**

Role of the Software Tester-Common Testing Fallacies-Challenges in Software Testing-Characteristics of Good Tests-Developing a Tester’s Mindset-Heuristics in Software Testing-Risk Based Testing Approaches-Importance of Negative Testing-Software Modeling for Testing

**Activities:** Understand the Role of a Tester and Identify Risks, Design Good Tests using Test Heuristics

**Advanced Test Design Techniques**

Exploratory and Scripted Testing-Equivalence Partitioning and Boundary Value Analysis-State Transition Testing Techniques-Decision Tables and Pairwise Testing-Stress and Load Testing Methods-Approaches to Security Testing-Regression Test Planning-Designing Test Automation-Effective Bug Reporting Practices

**Activities:** Perform Boundary Value, Equivalence, and Pairwise Testing

**Exploratory Testing in Practice**

Overview of Exploratory Testing-Charter-Based Testing Sessions-Test Tour Techniques-Investigative Testing with Models-Observation and Note-Taking Techniques-Use of Oracles and Heuristics-Session-Based Test Management (SBTM)-Debriefing and Reporting in Exploratory Testing-Designing Focused Test Missions

**Activities:** Conduct Exploratory Testing with Session Notes, Practice Bug Reporting and Use Oracles, Perform Load and Stress Testing

**Automation and Strategy Integration**

Guidelines for Test Automation-Automation in Exploratory Testing-Test Framework Selection and Design-Continuous Testing in DevOps Environments-Testing in Agile and CI/CD-Workflows-Behavior-Driven Development (BDD) Techniques-Analyzing Test Coverage and Identifying Gaps-Test Metrics and Dashboarding Tools-Managing Technical Debt in Testing

**Activities:** Design and Automate Regression Tests, Use BDD for Agile Testing

**Domain-Specific and Specialized Testing**

Mobile Application Testing-Testing Web and Cloud-Based Applications-Usability and Accessibility Testing-Testing for Embedded and IoT Systems-Game Testing Methodologies-Testing AI/ML-Based Software-Ethical Considerations in Software

Testing-Conducting Test Strategy Reviews and Audits-Emerging Trends in Software Testing

**Activities:** Perform Mobile and Web App Testing, Review Test Strategy and Metrics Dashboard

**Course Outcomes:**

1. Analyze and apply various advanced testing strategies to real-world problems.
2. Design and execute effective exploratory testing sessions.
3. Apply test heuristics and risk-based strategies in complex systems.
4. Integrate testing techniques into CI/CD and Agile environments.
5. Automate testing using appropriate tools and frameworks effectively.

**Weightage:** Continuous Assessment: 60%, End Semester Theory Examination: 40%

**(i) Activity: 15% (ii) Internal Theory Examination: 35% (iii) Internal Laboratory Examinations: 15%**

**Assessment Methodology:** Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

**References:**

1. Cem Kaner, James Bach, Bret Pettichord, Lessons Learned in Software Testing, Wiley, 2002.
2. Elisabeth Hendrickson, Explore It! Reduce Risk and Increase Confidence with Exploratory Testing, Pragmatic Bookshelf, 2013.
3. Rex Black, Advanced Software Testing – Vol 2, Rocky Nook, 2009.
4. Alan Richardson, Dear Evil Tester, Self-published, 2016.
5. Lisa Crispin & Janet Gregory, Agile Testing Condensed, Agile Testing Fellowship, 2019.

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, methodologies, and advanced practices of software testing used to ensure high-quality software systems.	--	--
CO2	Analyze advanced software testing techniques and test strategies to identify defects, assess risks, and improve software reliability.	PO1 (3)	PSO1 (3)
CO3	Evaluate software testing tools, automation frameworks, and testing models to determine their effectiveness in complex software environments.	PO3 (2)	PSO2 (2)
CO4	Design comprehensive and efficient software testing solutions by selecting appropriate techniques, tools, and automation strategies for real-world applications.	PO2 (1)	PSO1 (3)

CP25C06	Cloud and Big Data Analytics	L	T	P	C
		3	0	0	3
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>To understand cloud computing paradigms and service models.</li> <li>To study big data characteristics, technologies and storage.</li> <li>To explore Hadoop ecosystem and processing tools.</li> <li>To analyze, store, and visualize large-scale data.</li> <li>To develop and deploy cloud-based big data applications.</li> </ul>					
<p><b>Cloud Computing:</b> Introduction to Cloud Computing - Service Models: IaaS, PaaS, SaaS - Deployment Models: Public, Private, Hybrid, Community - Virtualization: Hypervisors, VM provisioning - Cloud providers: AWS, Azure, Google Cloud overview</p> <p><b>Activities:</b> Create a free-tier AWS account and explore EC2, S3.</p>					
<p><b>Big Data Ecosystem &amp; Hadoop:</b> Big Data characteristics (Volume, Velocity, Variety, Veracity, Value) - Challenges in Big Data - Hadoop Architecture and HDFS - MapReduce Programming Model –YARN</p> <p><b>Activities:</b> Watch Hadoop tutorial and submit a summary report.</p>					
<p><b>Data Storage &amp; Processing Techniques:</b> NoSQL Databases: MongoDB, Cassandra, HBase - Data Ingestion: Sqoop, Flume - Hive, Pig – Architecture and Querying - Data Cleaning and Preparation.</p> <p><b>Activities:</b> HiveQL assignments on datasets from Kaggle.</p>					
<p><b>Advanced Big Data Analytics:</b> Introduction to Apache Spark - Spark vs. MapReduce - Spark SQL, Data Frames, RDDs- Machine Learning using Spark MLlib - Real-time processing: Kafka and Spark Streaming</p> <p><b>Activities:</b> Mini-project using PySpark on COVID or Traffic dataset.</p>					
<p><b>Cloud Integration and Security:</b> Data on Cloud – Storage and Compute Services - Data Security and Privacy in Cloud - Cloud-based Data Analytics Use Cases - Cloud-Native vs Traditional Applications - Case Study: Smart Cities / Healthcare Analytics</p> <p><b>Activities:</b> Create a cloud deployment report using Azure/AWS.</p>					
<p><b>Weightage:</b> Continuous Assessment:40%, End Semester Theory Examination: 60%</p>					
<p><b>Assessment Methodology:</b> Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE &amp; IES questions (25).</p>					
<p><b>References:</b></p> <ol style="list-style-type: none"> <li>Rajkumar Buyya, James Broberg, Andrzej Goscinski, Cloud Computing: Principles and Paradigms, Wiley.</li> <li>Tom White, Hadoop: The Definitive Guide, O'Reilly Media.</li> </ol>					

3. Vignesh Prajapati, Big Data Analytics with R and Hadoop, Packt Publishing.
4. Kleppmann, Martin, Designing Data-Intensive Applications, O'Reilly.
5. Alan Gates, Programming Pig, O'Reilly.
6. Jure Leskovec et al., Mining of Massive Datasets, Cambridge University Press.

**E-resources:**

<https://www.nasa.gov/smallsat-institute/sst-soa/thermal-control/>

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and service models of cloud computing and big data analytics used in modern data-driven environments.	--	--
CO2	Analyze cloud-based big data processing frameworks and analytics techniques to understand their performance, scalability, and resource utilization.	PO1 (3)	PSO1 (3)
CO3	Evaluate big data analytics tools, cloud platforms, and data processing models for their effectiveness in handling large-scale and complex datasets.	PO3 (2)	PSO2 (2)
CO4	Design scalable cloud-based big data analytics solutions by selecting appropriate architectures, platforms, and analytical methods for real-world applications.	PO2 (1)	PSO1 (3)

CP25C07	Quantum Computing	L	T	P	C
		2	0	0	2
<b>Course Objective:</b>					
<ul style="list-style-type: none"> <li>• To provide a mathematical foundation for Quantum Computing and provide the basics of working</li> <li>• To interpret the various aspects and applications of quantum computing.</li> <li>• To examine the factors that affect Quantum computation</li> </ul>					
<b>Physical properties and mathematical foundations</b>					
Double Slit Experiment; Light: Particle Vs Wave; Heisenberg Uncertainty Principle. Vector spaces – basis; Inner product; Outer product; Tensor product; Linear operators					
<b>Activities:</b>					
<ul style="list-style-type: none"> <li>• Simulate the experiment using an interactive virtual lab</li> <li>• Construct simple operators and visualize action on vectors</li> </ul>					
<b>Quantum computing postulates and gates</b>					
Review of postulates, Bloch sphere, Single qubit states and gates, superposition; Two Qubit States and Gates - Bell States, Entanglement, CNOT gate, Phase oracles, Pauli Gates.					

<b>Activities:</b>	
<ul style="list-style-type: none"> <li>• Group quiz to match postulates to physical implications</li> <li>• Visualization with Bloch sphere simulators</li> </ul>	
<b>Quantum computing circuits</b>	
Dirac's notation for quantum computing, Computational Basis, Orthonormality, Hadamard and Phase Gates- building quantum circuits	
<b>Activities:</b>	
<ul style="list-style-type: none"> <li>• Use IBM Q Composer to build and simulate custom circuits</li> <li>• Hands-on: Apply X, H, Z gates and observe results on simulators</li> </ul>	
<b>Fundamental Quantum Algorithms</b>	
Deutsch–Josza Algorithm, Grover search algorithm: Problem definition, Amplitude amplification, Grover oracle, diffuser, multiple solutions in the search space	
<b>Activities:</b>	
<ul style="list-style-type: none"> <li>• Construct DJ circuit for a 3-bit input function</li> <li>• Simulation of Grover's algorithm with multiple marked elements</li> </ul>	
<b>Programming on a real quantum computer</b>	
Coding a real time quantum computer via IBMQ to carry out basic quantum measurement and state analysis.	
<b>Activities:</b>	
<ul style="list-style-type: none"> <li>• Connect Qiskit with IBMQ using personal API token</li> <li>• Hands-on: Create 1- and 2-qubit circuits using Hadamard, X, Z, and measurement gates</li> <li>• Compare real and simulated results and Observe impact of quantum noise</li> </ul>	
<b>Text Books:</b>	
<ol style="list-style-type: none"> <li>1. Chuck Easttom, "Quantum Computing Fundamentals", 1st edition, Published by Addison-Wesley Professional (June 1st 2021)</li> <li>2. Qiskit TextBook - <a href="https://qiskit.org/textbook/preface.html">https://qiskit.org/textbook/preface.html</a> (2022)</li> </ol>	
<b>References:</b>	
<ol style="list-style-type: none"> <li>1. Kasirajan, Venkateswaran. Fundamentals of quantum computing. Springer International Publishing, 2021.</li> <li>2. Chris Bernhardt, Quantum Computing for Everyone, The MIT Press, Cambridge, 2020</li> <li>3. Nielsen, Michael A., and Isaac L. Chuang, "Quantum Computation and Quantum Information" Cambridge University Press (5 April 2013)</li> </ol>	
<b>Weightage:</b> Continuous Assessment:40%, End Semester Theory Examination: 60%	
<b>Assessment Methodology:</b> Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).	

CO	Description of CO	PO	PSO
CO1	Describe the fundamental principles, postulates, and computational models of quantum computing and their significance in next-generation computing systems.	--	--
CO2	Analyze quantum algorithms and quantum circuit models to understand their computational advantages, limitations, and performance characteristics.	PO1 (3)	PSO1 (3)
CO3	Evaluate quantum computing paradigms, error correction techniques, and hardware technologies for their suitability in solving complex computational problems.	PO3 (2)	PSO2 (2)
CO4	Design quantum circuits and algorithmic solutions by selecting appropriate quantum gates, qubit architectures, and computational models for real-world problem scenarios.	PO2 (1)	PSO1 (3)

CP25C08	Advanced Software Testing and Quality Assurance	L	T	P	C
		3	0	0	3
<p><b>Course Objectives:</b></p> <ol style="list-style-type: none"> <li>1. Provide an in-depth understanding of advanced software testing techniques and their role in quality assurance.</li> <li>2. Explore automated testing frameworks, tools, and methods for test design and test process improvement.</li> <li>3. Enable students to design test plans, develop test cases, and assess test results critically.</li> </ol>					
<p><b>Foundations and Test Lifecycle:</b> Evolution of software testing in agile and DevOps ecosystems -Software Test Life Cycle (STLC) phases: Requirements analysis to test Closure - Quality attributes: reliability, maintainability, testability - Introduction to testing types and classifications.</p>					
<p><b>Activity:</b> Mapping STLC to Agile/DevOps pipelines.</p>					
<p><b>Types of Testing and Risk-Based Approaches:</b> Types of testing: Unit, Integration, System, Acceptance, Regression, Smoke, and Sanity - Risk-based testing and test Prioritization - Aligning testing strategies with product risk and delivery goals</p>					
<p><b>Activity:</b> Test strategy design for risk-based scenarios</p>					

**Advanced Test Design Techniques:** Black-box and white-box test strategies: BVA, ECP, decision tables, control/data flow - State transition and use-case testing - Model-based testing, combinatorial testing, and cause-effect graphing - Mutation testing, fault injection, fuzz testing - Test case minimization and prioritization techniques

**Activity:** Create test cases for a real-world problem using model-based design

**Test Automation and Continuous Testing:** Design of reusable and maintainable automation frameworks - Open-source tools: Selenium, TestNG, JUnit, Robot Framework, Cypress - Performance testing: JMeter and basic LoadRunner Concepts - Integration with Jenkins and GitHub for continuous testing - BDD and TDD using Cucumber

**Activity:** Implement a hybrid automation framework for a web-based application

**Software Quality Assurance Frameworks and Metrics:** Quality planning, control, assurance, and improvement - SQA tools and activities: Reviews, audits, inspections - Quality models: McCall, Boehm, ISO 9126 / ISO 25010 -Metrics for testing: test coverage, defect density, MTBF, test effort -Cost of quality and defect leakage

**Activity:** Analyze QA strategy in enterprise software (e.g., ERP or banking)

**Test Process Improvement and Research Trends:** Test maturity models: TMMi, CMMI-Dev (QA perspective) - Test process audits, benchmarking, and reviews -Empirical methods in testing and defect prediction models -AI/ML-based testing approaches: self-healing tests, anomaly detection -Testing trends: shift-left/right testing, visual and mobile testing.

**Activity:** Design a test improvement roadmap based on TMMi for a sample company

**Weightage:** Continuous Assessment: 40%, End Semester Theory Examinations: 60%

**Assessment Methodology:** Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

**References:**

1. Glenford J. Myers, The Art of Software Testing, 3<sup>rd</sup> edition, Wiley
2. Paul C. Jorgensen, Software Testing: A Craftsman's Approach, 4<sup>th</sup> edition, CRC Press
3. Ron Patton, Software Testing, Second Edition, Sams Publishing, Pearson Education, 2007
4. Aditya Mathur, Foundations of Software Testing, 2<sup>nd</sup> edition, Pearson, 2013
5. Pressman & Maxim, Software Engineering: A Practitioner's Approach, McGraw-Hill.

**E-Resources:**

1. MIT OpenCourseWare (Software Testing)
2. NPTEL (Software Testing by Prof. Nandini Prasad, IIT Kharagpur)

<b>CO</b>	<b>Description of CO</b>	<b>PO</b>	<b>PSO</b>
<b>CO1</b>	Describe the fundamental concepts, principles, and processes involved in advanced software testing and quality assurance for reliable software development.	--	--
<b>CO2</b>	<b>Analyze</b> software systems and testing strategies to identify defects, assess risks, and ensure compliance with quality standards.	PO1 (3)	PSO1 (3)
<b>CO3</b>	<b>Evaluate</b> software testing techniques, quality models, and assurance frameworks to determine their effectiveness in diverse development environments.	PO3 (2)	PSO2 (2)
<b>CO4</b>	<b>Design</b> comprehensive testing and quality assurance strategies by selecting appropriate testing tools, automation frameworks, and quality metrics for real-world software projects.	PO2 (1)	PSO1 (3)

CP25C09	Agile Methodologies	L	T	P	C
		3	0	0	3
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Introduce students to the principles, values, and practices of Agile software development.</li> <li>• Provide hands-on experience with Agile tools and techniques used in industry.</li> <li>• Develop the ability to plan, manage, and deliver software in Agile environments.</li> </ul>					
<p><b>Evolution of Software Development and Agile Foundations:</b> Evolution of Software Development Methodologies - Challenges in traditional models that led to Agile - The Agile Manifesto - Agile Values and Principles</p> <p><b>Activity:</b> Agile vs Waterfall in large-scale government or startup projects (e.g., FBI Sentinel, Spotify)</p>					
<p><b>Agile Mindset and Cultural Transformation:</b> Agile Mindset in Individuals and Teams - Characteristics of Agile teams and leaders - <b>Lean Thinking in Software Development</b> - Cultural Shift from Traditional to Agile Thinking - Applying Lean in Agile - Agile vs Traditional Methodologies</p> <p><b>Activity:</b> Reflective write-up on Agile mind set adoption challenges in traditional teams</p>					
<p><b>Agile Process Models and Frameworks:</b> Scrum: Roles, artifacts, events - Extreme Programming (XP): Pair programming, test-first, refactoring - Kanban: Flow, work-in-progress (WIP), pull systems - SAFe (Scaled Agile Framework) and Disciplined Agile Delivery (DAD) - Feature-Driven Development (FDD), Crystal, LeSS, Nexus</p> <p><b>Activity:</b> Team simulation using Scrum ceremonies (sprint planning, stand-ups, retrospectives)</p>					
<p><b>Agile Project Management and Planning:</b> User stories and story mapping - Estimation techniques: Planning poker, T-shirt sizing - Product backlog grooming and sprint planning - Agile metrics: Velocity, burndown chart, cumulative flow diagram (CFD) - Risk management in Agile</p> <p><b>Activity:</b> Tool Demo: Jira, Trello, Azure DevOps</p>					
<p><b>Agile Engineering Practices:</b> Test-Driven Development (TDD) and Behaviour Driven Development (BDD) - Continuous Integration and Continuous Delivery (CI/CD) - DevOps principles and its relationship with Agile - Refactoring, code smells, clean code - Design patterns in Agile context</p> <p><b>Activity:</b> Lab Component: Automate testing and CI/CD for a sample app</p>					
<p><b>Agile at Scale and Industry Case Studies:</b> Agile scaling frameworks and challenges - Distributed Agile teams and remote collaboration - Agile in non-software contexts (e.g., education, HR, manufacturing) - Agile contracting and governance - Agile case studies from Google, Spotify, IBM, and startups</p> <p><b>Activity:</b> Project based experiential learning: End-to-end Agile project using Scrum/Kanban and CI/CD pipeline</p>					

**References:**

1. Ken Schwaber & Jeff Sutherland, *Software in 30 Days: How Agile Managers Beat the Odds*, Wiley
2. Mike Cohn, *User Stories Applied: For Agile Software Development*, Addison-Wesley
3. Craig Larman & Bas Vodde, *Scaling Lean & Agile Development*, Addison-Wesley
4. Robert C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship*, Pearson

**E-Resources:**

1. NPTEL: Agile Software Development – Prof. Rajib Mall (IIT Kharagpur)
2. MIT Open Course Ware Agile *content*

**Weightage:** Continuous Assessment: 40%, End Semester Theory Examinations: 60%

**Assessment Methodology:** Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

CO	Description of CO	PO	PSO
CO1	Describe the fundamental principles, values, and frameworks of agile methodologies and their role in modern software development practices.	--	--
CO2	<b>Analyze</b> agile processes and practices to understand their effectiveness in managing requirements, collaboration, and iterative software development.	PO1 (3)	PSO1 (3)
CO3	<b>Evaluate</b> agile models, project management techniques, and team practices to assess their suitability for different organizational and project contexts.	PO3 (2)	PSO2 (2)
CO4	<b>Design</b> agile-based project workflows by selecting appropriate methodologies, roles, and practices to deliver high-quality software solutions.	PO2 (1)	PSO1 (3)

IF25C05	Generative AI	L	T	P	C
		3	0	0	3

**Course Objectives:**

- Introduce the fundamental concepts and evolution of generative modeling in AI.
- Study deep generative models including VAEs and GANs and their applications.
- Explore transformer-based models for multimodal generation.
- Provide hands-on experience in building generative AI systems using open-source tools.
- Analyze ethical, legal, and deployment aspects of generative technologies.

**Generative AI:** Overview of Generative AI – Differences between Discriminative and Generative Models – Applications in Art, Code, and Text – Overview of Deep Learning – Introduction to Latent Variable Models – Statistical Foundations – Random Sampling – Maximum Likelihood Estimation – Brief Introduction to AI Ethics.

**Activities:**

- **Group Discussion:** Compare and contrast discriminative vs. generative models with real-world examples.
- **Mini Research Presentation:** Each student presents one real-world application of Generative AI (art, music, code, etc.).
- **Concept Map:** Create a visual map showing the relationship between deep learning, latent variables, and generative models.

**Autoencoders and Variational Methods:** Autoencoders – Undercomplete vs. Overcomplete AEs – Denoising Autoencoders – Variational Autoencoders (VAE) – Latent Space Sampling – Kullback-Leibler Divergence – Reparameterization Trick – ELBO Objective – Applications: Face Generation, Image Compression.

**Activities:**

- **Build a Basic Autoencoder:** Use PyTorch or TensorFlow to implement an undercomplete autoencoder on the MNIST dataset.
- **Denoising Experiment:** Corrupt input images and train a **denoising autoencoder**
- **Conceptual Quiz:** Short quiz on latent space, VAE architecture, and reparameterization trick.

**Generative Adversarial Networks (Gans):** Architecture – Generator and Discriminator – Minimax Objective – Deep Convolutional GAN (DCGAN) – Conditional GANs – CycleGAN, StyleGAN – Challenges: Mode Collapse, Non-convergence – Evaluation Metrics: Inception Score, FID.

**Activities :**

- **Architecture Sketching:** Students draw and annotate the GAN architecture showing interactions between the Generator and Discriminator.
- **Experiment with DCGAN:** Modify a pre-trained DCGAN and generate new images.
- **Case Study Discussion:** Compare CycleGAN vs. StyleGAN for different use cases (e.g., image translation vs. high-quality image generation).

**FOUNDATION MODELS AND MULTIMODAL GENERATION**

Sequence Modeling – Attention Mechanism – Transformer Architecture – GPT Models – Text-to-Text and Text-to-Image Generation – CLIP and DALL·E – Multimodal Datasets – Prompt Engineering – Role of RLHF in Alignment.

**Activities:**

- **Transformer Demo:** Use Hugging Face to explore transformer-based models (e.g., GPT-2, T5).
- **Prompt Engineering Challenge:** Students experiment with different prompts using ChatGPT or similar to generate desired outputs.

- **Text-to-Image Generation:** Generate images using **DALL-E mini** or **Craiyon**, and critique results.

**Ethics, Safety and Deployment:** Generative AI Risks – Deepfakes and Detection – Bias and Hallucinations – Model Interpretability – Copyright and Intellectual Property – Safety Layers and Human Feedback – Open Source vs. Commercial APIs – Tools for Deployment – Hugging Face, Replicate, ONNX, Gradio.

**Activities :**

- **Workshop on Detection Tools:** Use available tools to detect AI-generated content (e.g., Deepfake Detection).
- **Group Presentation:** Propose safety mechanisms for deploying a generative AI model in a sensitive domain (healthcare, law, etc.).

**Capstone Project and Industry Applications:** Case Studies in Healthcare, Gaming, EdTech, Robotics – Model Compression and Optimization – Transfer Learning in Generative Context – Custom Fine-tuning – Deployment on Cloud Platforms – API Integration – Real-time Generative Systems.

**Activities :**

Understand real-world uses of generative AI across industries.

- Assign each team/individual one industry (e.g., healthcare, gaming).
- Research a real-world application using generative AI (e.g., AI-assisted medical report generation, generative design in games).
- Prepare a **5–7-minute presentation** covering:
  - Problem solved
  - Type of generative model used
  - Results and challenges
  - Ethical implications

<b>Weightage:</b>	Continuous Assessment: 40%	End Semester Theory Examination: 60%
	(i). Activities: 10% (ii). Internal Theory Examinations: 30%	

**Mandated Activities with marks:**

Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

**References:**

1. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press, 2016.
2. David Foster, Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play, O'Reilly Media, 2nd Edition, 2022.
3. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, 3rd Edition, 2022.
4. Sebastian Raschka and Vahid Mirjalili, Python Machine Learning, Packt Publishing, 3rd Edition, 2020.
5. Max Tegmark, Life 3.0: Being Human in the Age of Artificial Intelligence, Penguin Books, 2017.

**E-resources:**

1. <https://huggingface.co/learn>
2. <https://www.coursera.org/learn/build-basic-generative-adversarial-networks-gans>
3. <https://openai.com/research>
4. <https://developers.google.com/machine-learning/guides/text-classification>
5. <https://nptel.ac.in/courses/106105216> – Deep Learning by Prof. Mitesh Khapra (IITM)

<b>CO</b>	<b>Description of CO</b>	<b>PO</b>	<b>PSO</b>
<b>CO1</b>	Describe the fundamental concepts, models, and applications of generative artificial intelligence in creating data-driven content and solutions.	--	--
<b>CO2</b>	Analyze generative AI models and architectures to understand data generation processes, learning behavior, and performance characteristics.	PO1 (3)	PSO1 (3)
<b>CO3</b>	Evaluate generative AI techniques, models, and ethical considerations to assess their effectiveness and impact in real-world applications.	PO3 (2)	PSO2 (2)
<b>CO4</b>	Design generative AI solutions by selecting appropriate models, architectures, and training strategies for practical content generation tasks.	PO2 (1)	PSO1 (3)

ML25001	Auto ML	L	T	P	C
		3	0	0	3
<p><b>Course Objective:</b></p> <ul style="list-style-type: none"> <li>The objective of the AutoML course is to provide students with a comprehensive understanding of automated machine learning and its applications in solving complex real-world problems.</li> <li>This course aims to equip students with both the theoretical foundation and practical skills needed to use AutoML tools and techniques</li> <li>Students will explore various AutoML platforms such as TPOT, Auto-sklearn, and H2O.ai, and learn how these tools simplify and optimize the process of building machine learning models.</li> </ul>					
<p><b>Basics of AUTOML:</b> Overview of AutoML- Basics of Machine Learning - Types of Machine Learning: Supervised, Unsupervised, and Reinforcement Learning - Importance of AutoML in Modern Data Science - Overview of AutoML Tools (TPOT, Auto-sklearn, H2O.ai).</p> <p><b>Activities:</b></p> <ul style="list-style-type: none"> <li>Train a classification model on the Iris dataset and evaluate its accuracy.</li> <li>Use TPOT to automatically generate the best classification pipeline for the Titanic dataset and evaluate its performance.</li> </ul>					
<p><b>Foundational Techniques in Data Preprocessing:</b> Data Cleaning – Handling Missing Values – Categorical Data Encoding (One-Hot, Label) – Feature Scaling and Normalization.</p> <p><b>Activities:</b></p> <ul style="list-style-type: none"> <li>Preprocess the Boston Housing dataset by handling missing values and scaling features, then train a model and evaluate performance.</li> </ul>					
<p><b>Advanced Feature Engineering and Imbalanced Data Handling:</b> Feature Selection Techniques – Data Imbalance Handling (SMOTE, Oversampling, Undersampling).</p> <p><b>Activities:</b></p> <ul style="list-style-type: none"> <li>Apply SMOTE to handle class imbalance in the Breast Cancer dataset and train a classifier; evaluate its performance.</li> </ul>					
<p><b>Model Selection and Hyperparameter Tuning:</b> Model Selection – Grid Search – Random Search – Bayesian Optimization – AutoML Tuning – Cross-Validation – Ensemble Methods.</p> <p><b>Activities:</b></p> <ul style="list-style-type: none"> <li>Perform hyperparameter tuning for a Decision Tree model on the Wine Quality dataset using GridSearchCV and evaluate its performance.</li> <li>Use Auto-sklearn to automatically tune a model for the Diabetes dataset and compare its performance with a manually tuned model.</li> </ul>					

**Model Evaluation & Metrics:** Classification Metrics: Accuracy, Precision, Recall, F1 – Regression Metrics: MSE, RMSE,  $R^2$  – ROC & AUC – SHAP and LIME.

**Activities:**

- Evaluate a Random Forest classifier on the Iris dataset using accuracy, precision, recall, F1-score, and AUC.
- Perform K-fold cross-validation on a linear regression model using the Boston Housing dataset and calculate RMSE for each fold.

**Deployment & Monitoring:** Model Deployment with Flask/FastAPI – Monitoring – Drift Detection – Continuous Training – AutoML on Cloud and Edge.

**Activities:**

- Deploy a trained model (e.g., from the Titanic dataset) using Flask and serve predictions through an API.
- Set up a monitoring system to track model performance over time and retrain the model when accuracy drops below a threshold.

**Industry Exposure & Future of AUTOML:** Site Visit Observation – Use of AutoML in Finance, Healthcare, Marketing – Self-Learning AI – Scalability – Interpretability.

**Activities:**

- Report and presentation on industry visit
- Mini-project on a real-world problem using AutoML

**Weightage:** Continuous Assessment: 40%, End Semester Theory Examinations: 60%

**Assessment Methodology:** Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

**References:**

1. AlokShukla, Hands-On AutoML: A Practical Guide to Automated Machine Learning, Packt Publishing, 1st Edition, 2020.
2. Sebastian Raschka, VahidMirjalili, Python Machine Learning, Packt Publishing, 3rd Edition, 2019.
3. Frank Hutter, Lars Kotthoff, Joaquin Vanschoren, Automated Machine Learning: Methods, Systems, Challenges, Springer, 1st Edition, 2018.
4. AurélienGéron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, 2nd Edition, 2019.
5. Jason Brownlee, Automated Machine Learning: A Hands-On Approach, Machine Learning Mastery, 1st Edition, 2020.
6. Hugo Bowne-Anderson, Data Science at the Command Line, O'Reilly Media, 1st Edition, 2019.

**E-resources:**

1. **Auto-sklearn Documentation**  
<https://automl.github.io/auto-sklearn/>
2. **TPOT Documentation**  
<https://epistasislab.github.io/tpot/>
3. **Google Cloud AutoML**  
<https://cloud.google.com/automl>
4. **H2O.ai AutoML**  
<https://www.h2o.ai/products/h2o-automl/>
5. **Microsoft Azure AutoML**  
<https://azure.microsoft.com/en-us/services/machine-learning/automated-ml/>
6. **Kaggle Learn - AutoML**  
<https://www.kaggle.com/learn/overview>

<b>CO</b>	<b>Description of CO</b>	<b>PO</b>	<b>PSO</b>
<b>CO1</b>	Describe the fundamental concepts, frameworks, and workflows of Automated Machine Learning (AutoML) and its role in building efficient machine learning systems.	--	--
<b>CO2</b>	<b>Analyze</b> AutoML pipelines and optimization techniques to understand model selection, hyperparameter tuning, and performance trade-offs.	PO1 (3)	PSO1 (3)
<b>CO3</b>	<b>Evaluate</b> AutoML tools, frameworks, and generated models to assess their effectiveness, scalability, and applicability across different problem domains.	PO3 (2)	PSO2 (2)
<b>CO4</b>	<b>Design</b> machine learning solutions using AutoML approaches by selecting appropriate pipelines, search strategies, and evaluation methods for real-world applications.	PO2 (1)	PSO1 (3)

CP25C11	Advanced Deep Learning and Neural Networks	L	T	P	C
		3	0	0	3
<p><b>Course Objective:</b></p> <p>This course aims to provide a comprehensive understanding of neural network architectures and their training methodologies. The course explores optimization techniques, regularization methods, and state-of-the-art architectures including encoder-decoder models, attention mechanisms, variational auto encoders, GANs, and transformer-based models.</p>					
<p><b>Neural Networks:</b> Introduction – Perceptron – Multilayer Feed forward networks – Back propagation – Activation functions – Loss Function – Regularization: Data Augmentation - Noise Robustness – Early Stopping – Bagging – Dropout – batch normalization</p> <p><b>Activity:</b> Understanding the working principle of Neural Networks</p>					
<p><b>CNN and RNN:</b> Convolutional Neural Networks – Convolution Operation – Architecture Overview – Input layers – Convolutional layers – pooling layers – fully connected layers Recurrent Neural networks – LSTM – Bidirectional RNNs – RNN Language model – Word Level RNN - Deep Recurrent Networks – Recursive Neural Networks.</p> <p><b>Activity:</b> Reproduction of research paper related to CNN and RNN</p>					
<p><b>Optimization and Deep Networks:</b> Optimization in deep learning – Non-convex optimization for deep networks- Stochastic Optimization- Generalization in neural networks - Spatial Transformer Networks Tuning Deep Networks – Basic Concepts – Matching input data and network architecture – relating model goal and output layers – Working layer count, parameter count and Memory – weight initialization strategies – using activation functions – applying loss function – Dealing with overfitting.</p> <p><b>Activity:</b> Simple application using deep neural network with three layers</p>					
<p><b>Encoder / decoder Models and Regressive Models:</b> Encoder Decoder Models - Attention Mechanism - Attention over images – Hierarchical Attention – Variational auto encoders – Autoregressive models – NADE – MADE - PixelRNN – Generative Adversarial Networks (GANs) – Graph Convolution Network – Deep Belief Network.</p> <p><b>Activity:</b> Design of any Regressive, Generative and Graph Models</p>					
<p><b>Advanced Applications of Neural Networks:</b> Object Detection: RCNN, Faster RCNN, and Yolo – MobileNet – DarkNet – Object Tracking - Audio WaveNet - Natural Language Processing Word2Vec Joint Detection - Face Recognition - Scene captioning Language Transformer Models.</p> <p><b>Activity:</b> Design of face recognition model.</p>					
<p><b>Advanced applications of Deep learning:</b> Deep learning for Computer Vision – Deep learning for Time Series – Deep learning for text – Generative Deep learning</p> <p><b>Suggested Activity:</b> medical image segmentation and time series data analysis</p>					

**Weightage:** Continuous Assessment: 40%, End Semester Theory Examinations: 60%

**Assessment Methodology:** Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

### References

1. François Chollet, "Deep Learning with Python ", 2<sup>nd</sup> Edition, Manning Pubns Co, 2021.
2. Josh Patterson, Adam Gibson," Deep Learning: A Practitioner"s Approach", 1st Edition, O"Reilly Media, Inc, 2017.
3. Bengio, Yoshua, Ian J. Goodfellow, and Aaron Courville. "Deep learning." An MIT Press book in preparation, 2016.
4. Dr.Adrian Rosebrock, Deep Learning for Computer Vision with Python: Starter Bundell, Pylmage Search, 1st Edition, 2017.
5. Michael Nielsen, Neural Networks and Deep Learning, Determination Press, 2015.

### Online Courses

1. CS7015: Deep Learning (iitm.ac.in)
2. Deep Learning - Course (nptel.ac.in)
3. Deep Learning Part 1 (IITM) - Course (nptel.ac.in)
4. Deep Learning - IIT Ropar - Course (nptel.ac.in)

### CO-PO-PSO Mapping

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and learning paradigms of deep learning and neural networks used in advanced intelligent systems.	--	--
CO2	Analyze deep learning models and neural network architectures to understand their learning behavior, performance, and optimization characteristics.	PO1 (3)	PSO1 (3)
CO3	Evaluate deep learning techniques, neural network models, and training strategies to assess their effectiveness for complex data-driven applications.	PO3 (2)	PSO2 (2)
CO4	Design advanced deep learning solutions by selecting and integrating suitable neural network architectures, training algorithms, and optimization methods for real-world problems.	PO2 (1)	PSO1 (3)

CP25C13	Quantum Machine Learning	L	T	P	C
		3	0	0	3
<p><b>Course Objective:</b></p> <p>The objective of the Quantum Machine Learning (QML) course is to provide students with a comprehensive understanding of the intersection between quantum computing and machine learning. This course aims to equip students with the theoretical foundation and practical skills necessary to design, develop, and implement quantum algorithms for solving complex machine learning problems.</p>					
<p><b>Introduction to Quantum Computing and Quantum Machine Learning:</b> Basics of Quantum Computing: Qubits, Superposition, and Entanglement - Quantum Gates, Circuits, and Measurement - Quantum Algorithms: Grover's and Shor's Algorithm - Introduction to Quantum Machine Learning (QML) - Classical vs Quantum ML - Applications of Quantum ML</p> <p><b>Activities:</b></p> <ul style="list-style-type: none"> <li>• Implement a quantum circuit using Qiskit to demonstrate superposition and entanglement (Create a simple Bell state).</li> <li>• Use Qiskit to implement and visualize Grover's search algorithm for an unsorted database problem.</li> </ul>					
<p><b>Quantum Information Theory and Quantum Algorithms:</b> Quantum Entropy and Quantum Information - Quantum Teleportation and Superdense Coding - No-Cloning Theorem and Quantum Cryptography - Quantum Fourier Transform (QFT) - Quantum Phase Estimation (QPE)</p> <p><b>Activities:</b></p> <ul style="list-style-type: none"> <li>• Implement a Quantum Fourier Transform (QFT) using Qiskit and visualize the output on a simple example (e.g., the 3-qubit QFT).</li> <li>• Simulate quantum teleportation using Qiskit to transfer quantum information between two qubits.</li> </ul>					
<p><b>Quantum Algorithms for Machine Learning:</b> Quantum Support Vector Machines (QSVM) - Quantum Principal Component Analysis (QPCA) - Quantum k-Nearest Neighbors (QkNN) - Quantum Neural Networks (QNN)</p> <p><b>Activities:</b></p> <ul style="list-style-type: none"> <li>• Implement Quantum Support Vector Machine (QSVM) using Qiskit to classify a small dataset (e.g., Iris dataset).</li> <li>• Use Qiskit to perform Quantum Principal Component Analysis (QPCA) on a synthetic dataset for dimensionality reduction.</li> </ul>					
<p><b>Quantum Optimization and Hybrid Quantum-Classical Models:</b> Variational Quantum Eigensolver (VQE) - Quantum Approximate Optimization Algorithm (QAOA) - Hybrid Quantum-Classical Algorithms - Applications of Quantum Optimization in Machine Learning</p>					

**Activities:**

- Implement the Variational Quantum Eigensolver (VQE) to find the ground state energy of a Hamiltonian using Qiskit.
- Create a Quantum Approximate Optimization Algorithm (QAOA) to solve a Max-Cut problem on a simple graph using Qiskit.

**Applications, Tools, and Future of Quantum Machine Learning:** Applications of Quantum Machine Learning in Finance, Healthcare, NLP, and Drug Discovery - Quantum Programming Tools: Qiskit, TensorFlow Quantum - The Future of Quantum ML: Challenges and Opportunities

**Activities:**

- Use TensorFlow Quantum to build a simple quantum classifier and run it on a small dataset.
- Explore Qiskit and create a quantum-based recommendation system using hybrid quantum-classical methods.

**Advanced Quantum Machine Learning – I:** Quantum Mechanics and Data-Driven Physics - Kernelizing Quantum Mechanics - Qubit Maps - One-Qubit Transverse-Field Ising Model and Variational Quantum Algorithms

**Activities**

- Write a case study about combinatorial optimization with the Ising model and its quantum counterpart.

**Advanced Quantum Machine Learning – II:** Two-Qubit Transverse-Field Ising Model and Entanglement - Variational Algorithms, Quantum Approximate Optimization Algorithm, and Neural Network Quantum States with Two Qubits - Quantum Reservoir Computing

**Activities:** Write a case study about Quantum Approximate Optimization Algorithm.

**Observations from an Industry Exposure Visit:** For the Quantum Machine Learning (QML) course, a hands-on learning experience can be organized where students visit a quantum computing research lab, a quantum software company, or a machine learning-focused organization that integrates quantum computing into real-world applications. During this visit, students will have the opportunity to observe cutting-edge workflows involving quantum algorithms, quantum optimization techniques, and hybrid classical-quantum models applied to machine learning tasks. Students will gain insights into how these technologies are deployed to solve complex, high-dimensional problems in industries like finance, healthcare, and optimization. Following the visit, students will write a reflection report summarizing their observations, discussing the tools and technologies encountered, and reflecting on the practical challenges faced in quantum machine learning applications.

**Weightage:** Continuous Assessment: 40%, End Semester Theory Examinations: 60%

**Assessment Methodology:** Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

**References :**

1. Claudio Conti, Quantum Machine Learning Thinking and Exploration in Neural Network Models for Quantum Science and Quantum Computing, Springer Nature Link Publications 2024
2. Michael A. Nielsen, Isaac L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2nd Edition, 2010.
3. Peter Wittek, Quantum Machine Learning: What Quantum Computing Means to Data Mining, Springer, 2014.
4. Chris Bernhardt, Quantum Computing for Everyone, MIT Press, 1st Edition, 2019.
5. Eleanor G. Rieffel, Wolfgang H. Polak, Quantum Computing: A Gentle Introduction, MIT Press, 1st Edition, 2014.
6. Nathan Wiebe, Ashish Kapoor, Krysta Svore, Quantum Machine Learning, Springer, 1st Edition, 2021.
7. Brendon S. H. Tan, Quantum Computing and Quantum Information Science, CRC Press, 1st Edition, 2021.

**E-resources/E-materials:**

1. **Qiskit Documentation**  
<https://qiskit.org/documentation/>  
**IBM Quantum Experience**  
<https://quantum-computing.ibm.com/>
2. **Quantum Computing for the Very Curious**  
<https://quantum.country/qcvc>
3. **Google Quantum AI**  
<https://quantumai.google/>
4. **TensorFlow Quantum**  
<https://www.tensorflow.org/quantum>

## NPTEL/MOOC/SWAYAM Courses:

1. **NPTEL – Introduction to Quantum Computing**  
<https://nptel.ac.in/courses/106/105/106105226/>
2. **NPTEL – Quantum Computing and Quantum Information**  
<https://nptel.ac.in/courses/106/106/106106220/>
3. **Swayam – Quantum Machine Learning: Algorithms and Applications**  
[https://swayam.gov.in/nd1\\_noc21\\_cs56/](https://swayam.gov.in/nd1_noc21_cs56/)
4. **Coursera – Quantum Computing and Machine Learning**  
<https://www.coursera.org/learn/quantum-machine-learning>
5. **Udemy – Quantum Machine Learning**  
<https://www.udemy.com/course/quantum-machine-learning/>

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, models, and principles of quantum machine learning and their role in next-generation intelligent systems.	--	--
CO2	Analyze quantum machine learning algorithms and hybrid quantum–classical models to understand their computational advantages and limitations.	PO1 (3)	PSO1 (3)
CO3	Evaluate quantum machine learning techniques and frameworks to assess their effectiveness for complex data-driven and optimization problems.	PO3 (2)	PSO2 (2)
CO4	Design quantum machine learning solutions by selecting appropriate quantum circuits, learning models, and hybrid architectures for real-world applications.	PO2 (1)	PSO1 (3)

CP25C14	AI in IoT	L	T	P	C
		3	0	0	3
<p><b>Course Objective:</b></p> <ul style="list-style-type: none"> <li>To gain a deep understanding of AI concepts, IoT architectures, and their convergence for intelligent systems.</li> <li>To build and train machine learning and deep learning models suited for IoT data processing using TensorFlow / Keras.</li> <li>To apply data normalization, filtering, and feature engineering to enhance AI model accuracy</li> <li>To deploy AI models on edge devices for low-latency decision-making in IoT environments.</li> <li>To implement AI-based cyber security solutions for intrusion detection and anomaly detection in IoT networks.</li> </ul>					
<p><b>Introduction to AI in IoT:</b> Fundamentals of AI and IoT - IoT Architecture and AI Integration - AI Models for IoT (Machine Learning, Deep Learning) - AI-Driven IoT Applications (Smart Cities, Healthcare, Industrial IoT) - Ethical Considerations in AI-Enabled IoT – Case Study: Smart Agriculture with AI-Enabled IoT Sensors.</p> <p><b>Activities:</b></p> <ul style="list-style-type: none"> <li>Analytical study of AI integration within IoT ecosystems, focusing on architectural layers, data flow, and intelligence distribution across device, edge, and cloud.</li> <li>Simulation-based generation and statistical analysis of IoT sensor data using computational tools (e.g., Jupyter Notebook) to interpret trends, patterns, and variability.</li> </ul>					

**Data Acquisition & Pre-processing in AI-Enabled IoT:** Sensor Data Collection & Processing - AI-Based Data Normalization, Filtering & Feature Engineering - Challenges in Data Handling & Storage for AI Models - Ethical Considerations in AI-Driven Data Processing – Case Study: Predictive Maintenance in Smart Factories

**Activities:**

- Develop a conceptual framework for real-time sensor data preprocessing, detailing data collection, normalization, filtering, and feature engineering stages.
- Analyze AI-enhanced IoT data filtering techniques, focusing on methodologies, performance metrics, and practical challenges.

**AI Models for IoT Data Processing & Decision-Making:** Supervised & Unsupervised AI Models in IoT - Classification, Clustering & Anomaly Detection - Deep Learning Models for IoT: CNNs, RNNs - Performance Metrics & Model Optimization – Case Study: AI-Driven Traffic Monitoring System

**Activities**

- Analyze and justify the selection of appropriate AI models for real-time IoT data classification, considering problem context, data characteristics, and performance constraints.
- Conduct structured discussions or guided demonstrations on deep learning models (CNNs, RNNs) for IoT applications, focusing on architecture, working principles, and suitability for time-series data.

**Edge AI & Security for IoT Systems:** Edge AI Deployment Strategies - Edge AI for Real-Time Decision-Making - Deploying AI Models in Resource-Constrained IoT Devices - AI-Powered Security & Intrusion Detection in IoT Networks - Ethical Implications & Privacy Concerns – Case Study: AI- Based Intrusion Detection in Healthcare IoT Networks

**Activities:**

- Conceptually analyze AI-powered anomaly detection frameworks for IoT systems, focusing on model architecture, detection strategies, and performance evaluation criteria.
- Critically review and present recent research articles on security and privacy mechanisms in IoT systems, with emphasis on AI-based intrusion detection approaches.

**AI in IoT – Industry Trends:** Industry Applications of AI in IoT - Smart Cities, Healthcare, & Industrial IoT - Scaling AI for Large-Scale IoT Deployments - Sustainability & Scalability Challenges in AI-Enabled IoT

**Activities:**

- Undertake an analytical case study of AI-driven Smart Cities, examining sustainability, scalability, governance frameworks, and ethical considerations.

**AI in IoT – Ethics and Future Trends:** Monetization Strategies & AI Business Models in IoT - Green AI and IoT Regulations and Standardization - Edge AI advancements for real-time IoT decision-making - Quantum AI & IoT interoperability for future applications - Case Study: AI-Driven Smart Cities for Sustainable Urban Development

**Activities:**

- Prepare a conceptual design proposal for a real-time IoT-based decision-making system, detailing system architecture, AI model rationale, deployment strategies, and regulatory compliance aspects.

**Weightage:** Continuous Assessment: 40%, End Semester Theory Examinations: 60%

**Assessment Methodology:** Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

**References**

1. Fadi Al-Turjman, "Artificial Intelligence in IoT", Springer, 2019
2. Ambika Nagaraj, "The Role of AI in Enhancing IoT-Cloud Applications", Bentham Books, 2023
3. Rajkumar Buyya and Amir Vahid Dastjerdi, "Internet of Things: Principles and Paradigms", Elsevier, 2016
4. Jason Ioannou, "Data Analytics for IoT and Smart Systems", Springer, 2020
5. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, "Deep Learning", MIT

Press, 2016

6. Weisong Shi, "Edge Computing: Vision and Challenges", Springer, 2020
7. Mauro Conti, "Security and Privacy in the Age of IoT", Springer, 2020
8. François Chollet, "Deep Learning with Python" (2nd Edition), Manning Publications, 2021
9. Stuart Russell & Peter Norvig, "Artificial Intelligence: A Modern Approach (4th Edition), Pearson, 2020
10. Arshdeep Bahga & Vijay Madisetti, "Internet of Things: A Hands-On Approach" (2nd Edition), Universities Press, 2016

### Web References

<https://www.springer.com/gp/campaign/artificial-intelligence/artificial-intelligence-in-iot>

<https://iotworldmagazine.com/2025/01/16/2675/list-of-top-50-best-ai-automation-tools-in-2025>

<https://www.opensourceforu.com/2025/02/open-source-ai-frameworks-integrating-ai-with-iot/>

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and applications of integrating artificial intelligence techniques with Internet of Things (IoT) systems.	--	--
CO2	Analyze AI-enabled IoT systems to understand data processing, learning models, and decision-making mechanisms in distributed environments.	PO1 (3)	PSO1 (3)
CO3	Evaluate AI techniques and IoT platforms to assess their effectiveness, scalability, and reliability for intelligent connected applications.	PO3 (2)	PSO2 (2)
CO4	Design intelligent IoT solutions by selecting appropriate AI models, sensing strategies, and system architectures for real-world applications.	PO2 (1)	PSO1 (3)

ML25002	Equitable and Responsible AI	L	T	P	C
		3	0	0	3
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Understand the principles and importance of responsible AI in real-world systems.</li> <li>• Identify and address bias and fairness in AI datasets and models.</li> <li>• Apply interpretability and transparency frameworks for trustworthy AI.</li> </ul>					
<p><b>Foundations of Responsible AI:</b> Ethics in Artificial Intelligence – Human-centered and Inclusive AI – Historical failures of AI systems – Ethical frameworks (IEEE, OECD, ACM) – Principles of Responsible AI (transparency, accountability, fairness, privacy, safety).</p> <p><b>Activity</b></p> <ul style="list-style-type: none"> <li>• Case discussion on AI failures (e.g., COMPAS or Amazon Resume Tool) and mapping to ethical lapses.</li> </ul>					
<p><b>BIAS and Fairness in Machine Learning:</b> Understanding bias: dataset, label, measurement, historical bias – Fairness definitions: demographic parity, equalized odds, individual fairness – Mitigation strategies: pre-processing, in-processing, and post-processing – Measuring fairness in classifiers.</p> <p><b>Activity</b></p> <ul style="list-style-type: none"> <li>• Train and assess a classifier for fairness using group fairness metrics.</li> </ul>					
<p><b>Explainability, Transparency, and Accountability:</b> Need for explainability – Techniques: SHAP, LIME, counterfactuals – Model transparency and auditability – Human-AI interaction – Accountability and responsible reporting – Interpretable AI design principles.</p> <p><b>Activity</b></p> <ul style="list-style-type: none"> <li>• Prepare a responsible AI audit report for a loan prediction system</li> </ul>					
<p><b>Data Privacy, Security and Policy Considerations:</b> Differential Privacy and Federated Learning – Data security risks in AI – Privacy laws (GDPR, Indian PDP Act) – Ethical considerations in surveillance, facial recognition – National and International AI policies (EU AI Act, NITI Aayog Guidelines).</p> <p><b>Activity</b></p> <ul style="list-style-type: none"> <li>• Group presentation comparing Indian and EU AI guidelines.</li> </ul>					
<p><b>Responsible AI in Practice &amp; Social Impact:</b> AI for Social Good: healthcare, education, sustainability – Risk assessment in AI deployments – Responsible AI lifecycle – Capstone project planning and tool integration (Fairlearn, What-If Tool).</p> <p><b>Activity</b></p> <ul style="list-style-type: none"> <li>• Build a small socially beneficial AI model with integrated fairness/explain ability check</li> </ul>					

<b>Weightage:</b> Continuous Assessment: 40%, End Semester Theory Examinations: 60%
<b>Assessment Methodology:</b> Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)
<b>References:</b> <ol style="list-style-type: none"> <li>1. Kearns, M. &amp; Roth, A., The Ethical Algorithm: The Science of Socially Aware Algorithm Design, Oxford University Press, 2020.</li> <li>2. O’Neil, C., Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy, Crown Publishing, 2016.</li> <li>3. Ruha Benjamin, Race After Technology, Polity, 2019.</li> <li>4. Virginia Eubanks, Automating Inequality, St. Martin’s Press, 2018.</li> <li>5. Ethically Aligned Design, IEEE Global Initiative, 2021.</li> <li>6. IBM AI Fairness 360 Toolkit and Documentation.</li> <li>7. Google What-If Tool: <a href="https://pair-code.github.io/what-if-tool/">https://pair-code.github.io/what-if-tool/</a></li> </ol>
<b>E-Resources:</b> <ul style="list-style-type: none"> <li>• Prof. Ponnurangam Kumaraguru, Prof. Balaraman Ravindran, Prof. Arun Rajkumar   IIIT Hyderabad, IIT Madras Link “<a href="https://onlinecourses.nptel.ac.in/noc24_cs132/preview">https://onlinecourses.nptel.ac.in/noc24_cs132/preview</a>”</li> </ul>

CO	Description of CO	PO	PSO
CO1	Describe the fundamental principles, ethics, and societal considerations of equitable and responsible artificial intelligence.	--	--
CO2	<b>Analyze</b> AI systems to identify potential biases, fairness issues, and ethical challenges in data, models, and applications.	PO1 (3)	PSO1 (3)
CO3	<b>Evaluate</b> AI algorithms, frameworks, and policies to assess their compliance with fairness, accountability, and transparency standards.	PO3 (2)	PSO2 (2)
CO4	<b>Design</b> AI solutions incorporating fairness, accountability, and ethical guidelines to ensure responsible and socially equitable AI systems.	PO2 (1)	PSO1 (3)

CP25C16	Advanced Large Language Models	L	T	P	C
		3	0	0	3
<p><b>Course Objective:</b>  This course provides the basic knowledge to understand and harness the capabilities of cutting-edge language models. Large Language Models (LLM) have revolutionized the way we interact with text, enabling us to communicate, analyse, and generate content with unprecedented sophistication. This course will help the students to learn LLM from the introduction to the uses and limitations, the inner workings of LLMs, equipping the students with the tools to explore their versatile applications in content creation, code generation, sentiment analysis, and more.</p>					
<p><b>Building Blocks of Modern LLM's:</b> Large language model basics, Large Language model architecture, Transformer Architecture, Attention all you need – Encoder – Decoder</p> <p><b>Suggested Activities</b></p> <ul style="list-style-type: none"> <li>• Introduces the large language model basics and building blocks of language models.</li> <li>• A detailed study about language architecture and transformer architecture.</li> <li>• Analyze a real-world application for the Modern LLM.</li> </ul>					
<p><b>Pre-Training Methods:</b> Pre-training and Language Modeling, Auto-Regressive Language Modelling and Auto-Encoder Language Modelling, Early experiments with decoder and auto-encoder pretraining -Transformer Decoding Pretraining by Denoising Task</p> <p><b>Suggested Activities</b></p> <ul style="list-style-type: none"> <li>• Study the algorithm for Auto-Regressive and Auto-Encoder language modelling and implement the code to analyse the model flow.</li> <li>• Implement the decoding strategies, scaling methods and experiment the same.</li> </ul>					
<p><b>LLM Pretraining Methods:</b> Masked Language Modeling, BERT Pretraining with Masked LM, Autoregressive LM and Masked LM, LLM pretraining data, preprocessing clean text, Scaling Up pretraining data using the web, Language Model (LM)-Decoding Strategies.</p> <p><b>Suggested Activities</b></p> <ul style="list-style-type: none"> <li>• Study the algorithm for LLM pretraining and implement the code to analyse the model flow.</li> <li>• Implement the LM decoding strategies and experiment the same</li> </ul>					
<p><b>Parameter Efficient Tuning Methods:</b> Categories of PETM – Prefix tuning, Prompt tuning- Adapters, Compacters- Layer freezing-Bias terms, Fine-tuning- diff pruning, Low Rank Adaptation (LoRA) – Advantages and Pitfalls, Explainability and LLMs, ethical considerations.</p> <p><b>Suggested Activities</b>  Implement the LoRA methods for the standard dataset using the python code and analyze the results.</p>					

**In-Context Learning, Prompt Engineering and Alignment:** In-Context Learning, Fine-tuning - Zero-shot learning- Few-shot learning, Basics of prompting –Instruction prompting, Chain of thought prompting –Prompt selection, Automatic prompt design, Case Study: Visual Questions and Answering Systems, Sentiment training with Multi language dataset –CLIP.

**Suggested Activities**

- Learning the zero- shot learning, few-shot learning and prompt selection
- Build the question and answering systems
- Sentiment analysis using Multi-Language dataset

**Generation Based Automatic Evaluation Methods:** What is Evaluation – Human Evaluation- Intrinsic Vs Extrinsic Evaluation, Why automatic evaluation- Ranking- Multiple metrics, GLUE-GEM- Beyond metrics, Human Evaluation metric – Quantitative and Qualitative evaluation, Human annotation- Reporting- Challenges in Evaluation, Evaluation metrics.

**Suggested Activities**

- Analyze the GLUE-GEM score for the training and testing dataset.

**Weightage:** Continuous Assessment: 40%, End Semester Theory Examinations: 60%

**Assessment Methodology:** Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

**References**

1. Sebastian Raschka, Build a Large Language Model (From Scratch), September 2024, Manning Publications.
2. Tanmoy Chakraborty, Introduction to Large Language Models, Wiley Publications
3. Jay Alamar, Maarten Grootendorst, Hands on Large Language Models, Language Understanding and Generation, Shroff Publishers.
4. Amaratunga, Thimira, Understanding Large Language Models: Learning Their Underlying Concepts and Technologies, Springer Verlag, 2023
5. Raj Arun R, Mastering Large Language Models with Python, AVA publications 2024

**E-Resources**

1. NPTEL Course: Introduction to Large Language Modelling
2. <https://github.com/HandsOnLLM/Hands-On-Large-Language-Models>

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and capabilities of advanced large language models and their applications in intelligent systems.	--	--
CO2	<b>Analyze</b> large language model architectures and training methodologies to understand their performance, scalability, and limitations.	PO1 (3)	PSO1 (3)
CO3	<b>Evaluate</b> large language models and deployment strategies to assess their effectiveness, ethical considerations, and applicability in real-world scenarios.	PO3 (2)	PSO2 (2)
CO4	<b>Design</b> applications and systems leveraging advanced large language models by selecting appropriate architectures, fine-tuning strategies, and deployment frameworks.	PO2 (1)	PSO1 (3)

CP25C17	Edge and Fog Computing	L	T	P	C
		3	0	0	3

**Course Objective:**

This course aims to provide students with a comprehensive understanding of designing computing systems that process data closer to where it's created, rather than sending everything to distant cloud servers. Students will learn to build smart applications that work across different computing layers - from powerful cloud servers to small edge devices like sensors and smartphones. By the end, students can create practical solutions for smart cities, healthcare, and industry that respond quickly and work efficiently.

**Cloud to Edge Computing Paradigms:** Evolution from Cloud to Fog and Edge Computing - IoT Architecture Requirements and Latency Limitations - Fog vs Edge vs Cloud: Differences in Computation and Storage - Multi-access Edge Computing (MEC) Concepts - Multi-tier Architectural Models for Distributed Systems

**Suggested Activities:**

- Case Study: IoT solution using fog or edge
- Group Discussion: Business opportunities in edge markets

**Edge Offloading and System Optimization:** Edge Offloading Strategies and Placement Issues - Challenges in Latency, Energy Efficiency, QoS, and Privacy - Queuing Theory and System Performance Metrics - Optimization Techniques using Machine Learning for Slice Orchestration

**Suggested Activities:**

- Performance Comparison: Cloud vs Edge
- Mathematical Modeling: Latency or offloading simulation

**Resource Federation and Network Slicing:** Cloud-to-Fog-to-Things (C2F2T) Integration - Network Slicing for 5G/6G: Slice Types and Lifecycle - Resource Virtualization: SDN, NFV, and Container-Based Federation - Modeling and Simulation for Resource Management

**Suggested Activities:**

- Implementation Case Study: 5G network slicing for verticals
- Quiz: Virtualization and resource slicing concepts

**Middleware for Edge Systems:** Middleware Design Goals: Heterogeneity, Discovery, Scalability – Service Orchestration at the Edge - Lightweight Containers: Docker and Kubernetes on Constrained Devices - Edge Cluster Formation and Node Coordination - Communication Protocols: MQTT, CoAP, and Microservices - Distributed Storage, Synchronization, and Fault Recovery

**Suggested Activities:**

- Prototype: Middleware architecture on edge devices
- Containerization Project: Docker/Kubernetes deployment
- Security Analysis: Vulnerabilities in edge middleware

**Intelligence and Analytics at the Edge:** Data Management at the Fog Layer: Caching, Placement, Lifecycle - Federated Learning and Distributed ML for Edge Analytics - Stream Processing and Real-Time Analytics Frameworks - Use Cases: Smart Cities, Healthcare, Smart Homes - Security Challenges: ML-Based Threat Detection, Behavioral Analysis - Privacy Techniques: Differential Privacy, Federated Anonymization

**Suggested Activities:**

- Implement Predictive Analytics: Apache Kafka + Flink
- Develop ML Model: IoT threat detection
- Case Study: Privacy-preserving fog computing

**Advanced Technologies for Secure and Efficient Edge Computing:** Blockchain for Fog Resource Security and Transparency - Software-defined Edge Networks and Intent-Based Networking - Green and Sustainable Edge Computing-Security and Privacy Challenges in Edge Environments

**Suggested Activities:**

- Case Study: Edge security or sustainability
- Technical Report: Designing a secure, green edge architecture

**Future Directions and Emerging Trends:** Edge AI and TinyML: Neural Compression, Quantization, Model Deployment - 6G Vision: Ultra-Low Latency, Terahertz Communications - Advanced Paradigms: Digital Twins, Serverless Edge, Cognitive Networks

**Suggested Activities:**

- Research Project: Emerging tech integration with edge computing
- Presentation: Future research directions and innovation trends
- Technical presentation on future research directions in edge computing

**Weightage:** Continuous Assessment: 40%, End Semester Theory Examinations: 60%

**Assessment Methodology:** Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

**References:**

1. Rajiv Pandey, Sunil Kumar Khatri, Neeraj Kumar Singh, Parul Verma, Artificial Intelligence and Machine Learning for EDGE Computing, 1st Edition, 2022, Academic Press
2. S. Goundar, Ed., 'Edge Computing - Technology, Management and Integration'. IntechOpen, 2023.
3. Ahmed Banafa, Secure And Smart Internet of Things (IoT) using Blockchain and Artificial Intelligence (AI). 2018, River publications
4. Javid Taheri, Schahram Dustdar, Albert Zomaya, Shuiguang Deng, Edge Intelligence From Theory to Practice, 2023, Springer
5. Fog, Edge, and Pervasive Computing in Intelligent IoT Driven Applications Deepak Gupta, Aditya Khamparia, 2021, Wiley-IEEE Press

**E-Resources:**

1. Kubernetes on Edge: K3s Documentation and Tutorials - <https://k3s.io/>
2. Foundation of Cloud IoT Edge ML, NPTEL, Prof. Rajiv Misra, IIT Patna - <https://archive.nptel.ac.in/courses/106/104/106104242/>
3. Open5GS Documentation - <https://open5gs.org/>
4. Eclipse IoT Working Group - <https://iot.eclipse.org/>
5. TensorFlow Lite for Microcontrollers - <https://www.tensorflow.org/lite/microcontrollers>
6. Apache Flink Training - <https://flink.apache.org/>
7. Hyperledger Fabric Tutorials - <https://hyperledger-fabric.readthedocs.io/>
8. Subedi, P., Alsadoon, A., Prasad, P.W.C. et al. Network slicing: a next generation 5G perspective. J Wireless Com Network 2021, 102 (2021). <https://doi.org/10.1186/s13638-021-01983-7>

CO	CO Description	PO	PSO
CO1	Explain the evolution of cloud-to-edge paradigms and describe the role of edge and fog computing in real-time IoT applications.	PO2, PO3	PSO1
CO2	Design and optimize network slicing strategies for integrated cloud-fog-edge systems using mathematical modeling and optimization techniques.	PO1, PO3	PSO1, PSO2

<b>CO3</b>	Develop middleware architectures and container-based solutions for distributed edge computing environments.	PO1, PO3	PSO1
<b>CO4</b>	Implement intelligent data management systems and machine learning solutions at the fog layer for real-time analytics.	PO1, PO3	PSO1, PSO2
<b>CO5</b>	Evaluate emerging technologies and research directions in edge and fog computing for future system design.	PO1, PO2, PO3	PSO2

CP25C18	Green Computing and Sustainability	L	T	P	C
		3	0	0	3
<p><b>Course Objective:</b></p> <ul style="list-style-type: none"> <li>To understand the principles of green computing.</li> <li>To assess environmental impacts of computing technologies and need for green computing.</li> <li>To explore sustainable IT practices and energy-efficient hardware system for sustainability growth.</li> </ul>					
<p><b>Introduction to Green Computing:</b> Scope - Environmental Impact of Computing - Need for Sustainable Computing – Power-Aware Resource Management - Green IT - OCED Green IT Framework – Greening IT - Benefits and Challenges - Quantifying IT Energy Efficiency.</p> <p><b>Activities:</b> Review: E-waste management.</p>					
<p><b>Green Hardware:</b> Life cycle of hardware - Reuse, Recycle and Dispose Green Design and Manufacturing of Computers- Green Software - Energy-Saving Software Techniques</p> <p><b>Activities:</b> Demo: DAQ tools</p>					
<p><b>Sustainable Software Development:</b> attributes – metrics – Methodology - Energy-aware Software Engineering - Green Algorithms</p> <p><b>Activities:</b> Case study: Project's Carbon Footprint</p>					
<p><b>Green Data Center:</b> Infrastructure for Energy Efficiency – server power management</p> <p><b>Activities:</b> Poster presentation: Energy issues in Data Centers</p>					

**Green Data Storage:** Practices for Improving the Energy Efficiency of Data Storage - Optical Interconnects for Green Computers and Data Centers

**Activities:** Case study: Energy-efficient cloud computing

**Managing Green IT:** Strategizing Green Initiatives – Implementation - Green IT Laws, Standards and Protocols- Regulatory Environment and IT Manufacturers

**Activities:** Demo: Green IT Laws in India.  
Debate: Role of IT professional in Green Computing.

**Nonregulatory Government Initiatives:** Green Building Standards - Sustainability Reporting and Green Audits

**Activities:** Perform Energy audit in computer lab using software tools.

**Weightage:** Continuous Assessment: 40%, End Semester Theory Examinations: 60%

**Assessment Methodology:** Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

### References

1. San Murugesan, *Harnessing Green IT: Principles and Practices*, Wiley, 2012.
2. Suyel Namasudra(Editor), *Green and Sustainable Computing*, 1st Edition, Volume 87, Elsevier, 2012.
3. Toby Velte, Anthony Velte, Robert Elsenpeter, "Green IT: Reduce Your Information System's Environmental Impact", McGraw Hill, 2008.
4. Bhuvan Unhelkar, "Green IT Strategies and Applications: Using Environmental Intelligence", CRC Press, 2011.
5. Balamurugan Balusamy, Naveen Chilamkurti, Seifedine Kadry, *Green Computing in Smart Cities: Simulation and Techniques*, Springer Singapore, 2020.
6. Mike Halsey, *The Green IT Guide - How to Make Your IT Systems and Business Sustainable and Carbon Neutral*, Apress, 2025

### E-Resources

1. Prof. Deepu Philip, Prof. Amandeep Singh | IIT Kanpur  
Link [http://onlinecourses.nptel.ac.in/noc21\\_mg85/preview](http://onlinecourses.nptel.ac.in/noc21_mg85/preview)
2. Prof. Sayak Banerjee | IIT Hyderabad  
Link "https://onlinecourses.nptel.ac.in/noc23\_me138/preview"

<b>CO</b>	<b>Description of CO</b>	<b>PO</b>	<b>PSO</b>
<b>CO1</b>	Describe the fundamental concepts, principles, and practices of green computing and sustainability in modern information technology systems.	--	--
<b>CO2</b>	Analyze computing systems and technologies to assess their energy efficiency, environmental impact, and sustainability metrics.	PO1 (3)	PSO1 (3)
<b>CO3</b>	Evaluate green computing strategies, sustainable IT practices, and regulatory frameworks for their effectiveness in reducing environmental impact.	PO3 (2)	PSO2 (2)
<b>CO4</b>	Design sustainable computing solutions by selecting energy-efficient architectures, technologies, and practices for environmentally responsible systems.	PO2 (1)	PSO1 (3)

<b>IF25C03</b>	<b>DevOps and Micro Services</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>
<b>Course Objectives:</b>					
<ul style="list-style-type: none"> <li>• To understand the DevOps culture and its significance in modern software development.</li> <li>• Explore automation tools and practices used in continuous integration, delivery, and deployment, gain hands-on experience with version control, build automation, testing, and infrastructure as code and develop the ability to monitor, manage, and secure DevOps pipelines effectively.</li> </ul>					

## DevOps Foundations and Process Automation

Evolution of Software Development Life Cycle (SDLC), Agile, Lean, and DevOps  
What is DevOps? Principles and Practices, DevOps vs Traditional IT, Benefits and Challenges of DevOps, DevOps Lifecycle and Key Concepts (CI/CD, Infrastructure as Code, Monitoring, Feedback Loops), DevOps Toolchain Overview. DevOps maturity and automation

### Activities :

- **Quiz:** Short quiz on DevOps principles, benefits, and challenges.
- **Case Study:** Analyze a company's software delivery before and after adopting DevOps. Identify key improvements.

**Version Control and Continuous Integration:** Basics: Repositories, Branching, Merging, Rebase, GitHub/GitLab for Collaboration, Continuous Integration Concepts, Build Automation Tools: Maven, Gradle, CI Tools: Jenkins, GitHub Actions, GitLab CI, Creating and Managing CI Pipelines.

### Activities:

- **Hands-on Lab:** Initialize a Git repository, create branches, and perform merges and rebases.
- **Collaboration Exercise:** Use GitHub/GitLab to fork a repository, submit pull requests, and review code.

**Configuration Management and Containerization:** Configuration Management Tools: Ansible, Puppet, Chef, Infrastructure as Code (IaC) Concepts, Introduction to Docker: Images, Containers, Dockerfiles, Docker Compose, Container Orchestration Overview (Kubernetes basics), Comparing VMs vs Containers

### Activities:

- **Tool Demo:** Write a basic Ansible playbook or Puppet manifest to configure a web server.
- **Discussion:** Compare VM and container architectures; list pros and cons.

**Continuous Deployment and Monitoring:** Continuous Deployment vs Continuous Delivery, Tools for Deployment: Jenkins Pipelines, Spinnaker, Monitoring Tools: Prometheus, Grafana, ELK Stack, Logging and Alerting Mechanisms, DevOps Metrics and KPIs.

### Activities:

- **Monitoring Setup:** Install and configure Prometheus and Grafana dashboards to monitor application metrics.
- **Log Analysis:** Set up ELK stack to collect and analyze logs from a demo app.

**Cloud, Security, and Best Practices:** Introduction to Cloud Platforms: AWS, Azure, GCP, DevOps on Cloud: CloudFormation, Terraform, SecDevOps: Integrating Security into DevOps, Blue-Green and Canary Deployments, Best Practices in DevOps: Culture, Collaboration, Feedback Loops.

**Activities:**

- **Security Review:** Conduct a security checklist for CI/CD pipeline (identify vulnerabilities and propose fixes).
- **Deployment Strategy Roleplay:** Simulate blue-green and canary deployments and discuss rollback strategies.
- **Culture Brainstorm:** Develop a plan to improve collaboration and feedback loops in a team environment.

<b>Weightage:</b>	Continuous Assessment: 40%	End Semester Theory Examination: 60%
	(i). Activities: 10% (ii). Internal Theory Examinations: 30%	

**Mandated Activities with marks:**

Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

**References:**

1. Gene Kim, Jez Humble, Patrick Debois, and John Willis, The DevOps Handbook, IT Revolution Press, 2016.
2. Jez Humble and David Farley, Continuous Delivery, Addison-Wesley, 2010.
3. Kelsey Hightower et al., Kubernetes: Up and Running, O’Reilly Media.
4. Official documentation of tools like Git, Docker, Kubernetes, Jenkins, Ansible, Terraform.

**E-resources:**

**NPTEL Video**

1. <https://youtu.be/2N-59wUIPVI?si=4SrHAJvOsOKmxi21>
2. <https://youtu.be/87ZqwoFeO88?si=084GDdu-pDfsVKMq>

CO	Description of CO	PO	PSO
<b>CO1</b>	Describe the fundamental concepts, principles, and practices of DevOps and microservices architecture for efficient software delivery.	--	--
<b>CO2</b>	Analyze DevOps pipelines and microservices-based systems to understand deployment strategies, integration, and scalability.	PO1 (3)	PSO1 (3)
<b>CO3</b>	Evaluate DevOps tools, microservices frameworks, and operational practices to assess their effectiveness in continuous delivery and system reliability.	PO3 (2)	PSO2 (2)

<b>CO4</b>	Design DevOps-enabled microservices solutions by selecting appropriate architectures, tools, and workflows for real-world software applications.	PO2 (1)	PSO1 (3)
------------	--	------------	-------------

<b>CP25C19</b>	<b>Cognitive Computing</b>	L	T	P	C
		3	0	0	3

**Course Objectives:**

- To understand the core principles of cognitive computing and how they emulate human thought processes.
- To apply cognitive computing to solve complex, unstructured, and real-world problems.
- Develop cognitive computing applications.

Human cognition - Introduction to Cognitive Computing – Elements of a Cognitive System - Design Principles for Cognitive Systems - Artificial Intelligence, Natural Language Processing and Machine Learning in Cognitive Computing.

**Activities:**

1. Concept Quizzes: Human cognition and Cognitive Computing
2. Research paper review on cognitive systems

Cognitive architectures: ACT-R, Soar, OpenCog - Knowledge Representation and Reasoning - Machine Learning models in cognitive systems - Human-like Learning: Supervised, Unsupervised, Reinforcement

**Activities:** Demo on IBM Watson APIs (Language Translator)

Role of NLP in a Cognitive System - NLP pipeline: POS tagging, NER, parsing - Sentiment Analysis and Question Answering

Debate: Sentiment analysis and its challenges in Regional Languages.

Semantic Web - Semantic Understanding: Ontologies and Knowledge Graphs

**Activities:**

Ideathon: Innovative solution to health care problems.

Cognitive Agents - Rule-based Agents and Expert Systems - Decision Support Systems -Emotion-aware computing and human-computer interaction- Cognitive bias and ethics in decision making

**Activities:** Mini project: Develop health care chatbot. Prototype Demonstrations.

Process of Building a Cognitive Application - IBM's Watson as a Cognitive System- Business Implications of Cognitive Computing

**Activities:** Research Paper Review: Cognitive Computing

Cognitive Computing Applications: Healthcare, Finance, Education, Legal, Robotics- Building a Cognitive Healthcare Application - Emerging Cognitive Computing Areas - Cognitive cloud- Cognitive IoT - Cognitive Chatbots - Neuromorphic computing and brain-inspired models.

**Activities:**

1. Prototype Demonstrations.
2. Project demo and peer review.

**Weightage:** Continuous Assessment: 40%, End Semester Theory Examinations: 60%

**Assessment Methodology:** Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

### References

1. Velankar, Mahalle & Shinde – Cognitive Computing for Machine Thinking, CRC Press, 2024.
2. Judith S. Hurwitz, Marcia Kaufman, Adrian Bowles – Cognitive Computing and Big Data Analytics, Wiley, 2015.
3. Peter Fingar , Cognitive Computing: A Brief Guide for Game Changers, Meghan Kiffer, 2015.
4. Building Cognitive Applications with IBM Watson Services, Redbooks, 2017
5. Steven Bird, Ewan Klein, Edward Loper, Natural Language Processing with Python, O'Reilly, 2009.
6. Fuchun Sun, Qinghu Meng, Zhumu Fu, Bin Fang (Editors), Communications in Computer and Information Science (CCIS, volume 1918) – Springer Series, 2024.

### E-Resources

- Prof. Sharba Bandyopadhyay, Prof. Rajlakshmi Guha | IIT Kharagpur, IIT Kharagpur  
Link [https://onlinecourses.nptel.ac.in/noc22\\_ee122/preview](https://onlinecourses.nptel.ac.in/noc22_ee122/preview)
- Cognition and its Computation, IIT Kharagpur  
Prof. Sharba Bandyopadhyay, Prof. Rajlakshmi Guha  
Link “<https://nptel.ac.in/courses/108105185>”

<b>CO</b>	<b>Description of CO</b>	<b>PO</b>	<b>PSO</b>
<b>CO1</b>	Describe the fundamental concepts, architectures, and paradigms of cognitive computing and their role in intelligent decision-making systems.	--	--
<b>CO2</b>	Analyze cognitive computing models and learning mechanisms to understand perception, reasoning, and knowledge representation in intelligent systems.	PO1 (3)	PSO1 (3)
<b>CO3</b>	Evaluate cognitive computing techniques, frameworks, and applications to assess their effectiveness in solving complex real-world problems.	PO3 (2)	PSO2 (2)
<b>CO4</b>	Design cognitive computing solutions by selecting appropriate models, algorithms, and system architectures for human-centric intelligent applications.	PO2 (1)	PSO1 (3)

<b>CP25C20</b>	<b>Agentic AI</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>
<p><b>Course Objectives:</b></p> <ol style="list-style-type: none"> <li>1. Understand the foundations and architecture of intelligent agents.</li> <li>2. Design agents capable of autonomous and rational decision-making.</li> <li>3. Apply reinforcement learning for adaptive agent behavior.</li> <li>4. Explore coordination in multi-agent and human-AI systems.</li> <li>5. Analyse the ethical implications of agentic systems.</li> <li>6. Develop real-world applications using modern agent-based AI frameworks.</li> </ol>					
<p><b>Foundations of Intelligent Agents</b></p> <p>Introduction to Agentic AI: History, Motivation, Applications - Agents and Environments: Sensors, Actuators, Environment Types - Types of Agents: Simple Reflex, Goal-Based, Utility-Based, Learning Agents - Architectures: Reactive, Deliberative, Hybrid, Subsumption</p> <p><b>Activity:</b></p> <ul style="list-style-type: none"> <li>• Report preparation about types of Agents, role and architecture of each agent</li> </ul>					

## **Agent Decision-Making and Planning**

Rationality and Utility Theory - Task Environment Analysis - AI Planning Techniques: STRIPS, Classical Planning - Decision-Making under Uncertainty: MDPs, Bayesian Models, Game Theory

### **Activity:**

- Design a simple application for making decision using AI in python

## **Learning in Agentic Systems**

Reinforcement Learning: Q-Learning, SARSA - Value and Policy Iteration - Deep RL: DQNs, Policy Gradient Methods, Actor-Critic - Tools: OpenAI Gym, PettingZoo, TensorFlow RL

### **Activity:**

- Write a case study for Tools about OpenAI and TensorFlow with proper application

## **Multi-Agent Systems and Collaboration**

Multi-Agent Coordination: Cooperation, Competition - Agent Communication and Negotiation - Human-AI Collaboration: Interactive and Mixed-Initiative Systems - Platforms: NetLogo, RoboCup Simulation

### **Activity:**

- Create Autonomous agents using NetLogo platform

## **LLM Agents and Embodied Cognition**

Language-based Agents: Tool Use, Planning, and Reasoning - Memory-Augmented and Situated Agents - Agents in Robotics and Smart Environments - Unity ML-Agents, Hugging Face Tools

### **Activity:**

- Design a simple application based on LLM-based tool-using agents

## **Ethics and Applications of Agentic AI**

AI Alignment, Fairness, Explainability - Social and Environmental Implications - Case Studies: Autonomous Vehicles, Conversational Agents - Course Project Evaluation and Presentations

### **Activity:**

- Agent-based simulations for real-world problems

**References:**

1. Artificial Intelligence: A Modern Approach, Russell & Norvig, 4th Ed., Pearson, 2021
2. Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations, Shoham & Leyton-Brown, CUP, 2009
3. Reinforcement Learning: An Introduction, Sutton & Barto, 2nd Ed., MIT Press, 2018

**Tools & Technologies:****Languages:** Python**Libraries:** PyTorch, TensorFlow, OpenAI Gym, LangChain**Frameworks:** PettingZoo, Unity ML-Agents, NetLogo, Hugging Face**Platforms:** Jupyter, VS Code, RoboCup Simulators

<b>Weightage:</b>	<b>Continuous Assessment: 40%</b>	<b>End Semester Theory Examinations: 60%</b>
	<b>(i) Activity: 10%</b> <b>(ii) Internal Theory Examination: 30%</b>	

**Mandated Activities with Marks:**

Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

**Internal Examinations: Two Tests**

<b>CO</b>	<b>Description of CO</b>	<b>PO</b>	<b>PSO</b>
<b>CO1</b>	Describe the fundamental concepts, architectures, and principles of agentic artificial intelligence and autonomous agent systems.	--	--
<b>CO2</b>	Analyze agentic AI models and decision-making mechanisms to understand autonomy, goal-directed behavior, and interaction in intelligent agents.	PO1 (3)	PSO1 (3)
<b>CO3</b>	Evaluate agentic AI frameworks, learning strategies, and coordination techniques to assess their effectiveness in complex and dynamic environments.	PO3 (2)	PSO2 (2)
<b>CO4</b>	Design agentic AI solutions by selecting appropriate agent architectures, learning methods, and coordination mechanisms for real-world applications.	PO2 (1)	PSO1 (3)

CP25C21	Mixed Reality	L	T	P	C
		3	0	0	3
<p><b>Course Objective:</b>  This course to define mixed reality and differentiate between virtual reality and augmented reality. It is used to describe the core ideas behind holograms and start designing and developing 3D applications. It also used to understand what mixed reality offers in real-world apps.</p>					
<p><b>Defining Virtual and Augmented Reality</b>  Four Key Elements of Virtual Reality Experience Looking at Some Other Types of Virtual and Augmented Reality.</p> <p><b>Activity:</b></p> <ul style="list-style-type: none"> <li>• Assignment about Virtual Reality Experience for real world applications</li> </ul>					
<p><b>Exploring the Current State of Virtual Reality</b>  Looking at the Available Form Factors, Focusing on Features. Room-scale versus stationary experience. Inside-out tracking. Tracking devices Haptic feedback, Audio, Considering Controllers.</p> <p><b>Activity:</b></p> <ul style="list-style-type: none"> <li>• Create a flipped class room for exploring the current state of Virtual Reality</li> </ul>					
<p><b>Exploring the Current State of Augmented Reality</b>  Mobile devices AR headsets. AR glasses. Hand tracking. Motion controllers Hand and Gesture Tracking, Whole Body Tracking</p> <p><b>Activity:</b></p> <ul style="list-style-type: none"> <li>• Create a flipped class room for exploring the current state of Augmented Reality</li> </ul>					
<p><b>Simple VR Environment Development</b>  Interacting with the Virtual World, Interactions: Manipulation, Navigation, and Communication, <b>Creating</b> Basic objects in Sketchup tool, Grouping and Components, Development of Virtual Environment using SketchUp tool, connecting the VR environment to Meta Quest 2 (HMD) using VR Sketch, Navigation in the Virtual Environment.</p> <p><b>Activity:</b></p> <ul style="list-style-type: none"> <li>• Create a Simple Virtual Environment for a simple real - world application</li> </ul>					
<p><b>Creating 2D/3DProjects</b></p>					

Getting started with Unity, Installing Unity Hub and Unity Editor, Editor interface, Creating Projects with templates, Asset store, creating a simple 2d game using game objects, Scripting in c#, Visual Scripting, creating a simple 3D game by importing game objects from asset store, Building the projects to desktop OS/ Mobile OS

**Activity:**

- Create a simple 3D game using Visual Scripting

**Setting Up for AR Development**

Installing XR plugins for AR devices, Installing the AR Foundation package, Building the Simple AR scene in project, Importing the Sample assets, Using AR Session and AR Session Origin, AR Camera, Adding AR Ray cast Manager, Building and running the scene. Placing an object on a plane, Writing the Place Object on Plane script, creating a prefab for placing on the scene.

**Activity:**

- Write a case study about setting up for AR development

**References:**

1. Jonathan Lenowes, Augmented Reality with Unity AR Foundation, 2021 Packt Publishing
2. William R. Sherman, Alan B. Craig, Understanding-Virtual-Reality-Interface-Application-and-Design-The-Morgan-Kaufmann-Series-in-Computer-Graphics
3. Virtual & Augmented Reality for Dummies by Paul Mealy, Wiley Publications 2013
4. Chen, Virtual, Augmented and Mixed Reality. Applications and Case Studies, Springer Publications 2019.
5. Rajiv Chopra, Virtual and Augmented Reality, Khanna Publishing House, 2021

<b>Weightage:</b>	<b>Continuous Assessment: 40%</b>	<b>End Semester Theory Examinations: 60%</b>
	<b>(i) Activity: 10%</b> <b>(ii) Internal Theory Examination: 30%</b>	
<b>Mandated Activities with Marks:</b> Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)		
<b>Internal Examinations: Two Tests</b>		

<b>CO</b>	<b>Description of CO</b>	<b>PO</b>	<b>PSO</b>
<b>CO1</b>	Describe the fundamental concepts, technologies, and components of mixed reality and their applications in immersive digital environments.	--	--

<b>CO2</b>	<b>Analyze</b> mixed reality systems and interaction techniques to understand performance, usability, and user experience in immersive applications.	PO1 (3)	PSO1 (3)
<b>CO3</b>	<b>Evaluate</b> mixed reality platforms, devices, and development frameworks to assess their suitability for various application domains.	PO3 (2)	PSO2 (2)
<b>CO4</b>	<b>Design</b> mixed reality applications by selecting appropriate hardware, software frameworks, and interaction models for real-world use cases.	PO2 (1)	PSO1 (3)

<b>BD25C02</b>	<b>Exploratory Data Analysis</b>	L	T	P	C
		3	0	0	3

**COURSE OBJECTIVES:**

This course aims to provide in-depth knowledge and practical skills in performing Exploratory Data Analysis (EDA), a foundational step in data science workflows. Students will learn to manipulate, analyze, and visualize data using statistical and graphical techniques.

**INTRODUCTION**

Purpose and goals of EDA - Mindset for effective data exploration - Significance of EDA in the data science lifecycle - Types of data and quality issues - EDA vs. Classical and Bayesian approaches - Tools and libraries for EDA - Visual techniques for initial insights.

**ACTIVITIES:**

1. Conduct a Think Pair Share activity on dataset exploration and question formulation.
2. Conduct a Jigsaw activity on Data type and quality issue identification.

**DATA WRANGLING**

Importing, loading, and cleaning datasets - Merging, reshaping, and pivoting data - Handling missing values and outliers - Renaming, deduplication, discretization, and binning - Permutation and random sampling - Challenges in real-world data preprocessing.

**ACTIVITIES:**

1. Conduct a Peer Debugging Session on handling missing values, duplicates, and type inconsistencies.
2. Assign a Mini Project on real-world dataset wrangling, focusing on merging and reshaping data.
3. Conceptual discussion on outlier identification.

**MULTIVARIATE ANALYSIS, RELATIONSHIP EXPLORATION AND CAUSAL INFERENCE**

Univariate, Bivariate, and Multivariate Analysis - Interpreting relationships and Simpson's Paradox - Multicollinearity and correlation pitfalls - Visualizing multivariate data using pair plots and heatmaps - Causal inference in data science: correlation vs. causation, confounding variables

**ACTIVITIES:**

1. Conduct a *Group Analysis Task* to explore correlation matrices and identify multicollinearity.
2. Use a *Role Play Simulation* to explain Simpson's Paradox and how it affects decision-making.

3. Conduct a Visual Inference Exercise where students use pair plots and heatmaps to explore relationships and patterns within a dataset.

### **TIME-ORIENTED DATA EXPLORATION**

Time-series data - Time-based indexing and date time conversion in pandas - Seasonal patterns, resampling (up/down), and aggregations - Line plots, rolling statistics, and anomaly spotting.

#### **ACTIVITIES:**

1. Facilitate an activity where students convert raw time-based data into a time series and analyze it to identify key trends and patterns.
2. Conduct a Pattern Spotting Contest using resampled time-series plots to identify anomalies.
3. Assign a Team Activity to evaluate the effect of resampling and rolling averages on data interpretation.

### **STATISTICAL TESTING FOR INSIGHTS**

Hypothesis Testing: Null vs. Alternative, Type I & II Errors - Statistical tests: t-tests, p-values using scipy and statsmodels - Interpreting test results in the context of EDA.

#### **ACTIVITIES:**

1. Conduct a Think-Pair-Share activity on hypothesis formulation from a business context.
2. Organize a Classification Activity where students identify examples of Type I and Type II errors.
3. Conduct a Hands-On Testing Activity where students perform t-tests on sample datasets and interpret the resulting p-values to draw meaningful conclusions.

### **FROM EDA TO MODEL DEPLOYMENT**

Supervised vs. unsupervised models - Simple and Multiple Linear Regression - Train/Test Split, Cross-validation - Evaluation metrics: MAE, RMSE,  $R^2$ , Accuracy, F1-score - Saving models using pickle or joblib - Basic deployment using Flask for local predictions.

#### **ACTIVITIES:**

1. Conduct a Case-Based Discussion on choosing between supervised and unsupervised models.
2. Conduct a Hands-On Evaluation Activity where students calculate and interpret key performance metrics such as RMSE, Accuracy, and F1-score to assess model effectiveness.
3. Build a simple Flask app for local prediction.

<b>Weightage:</b>	Continuous Assessment: 60%	End Semester Theory Examination: 40%
	(i). Activities: 10% (ii). Internal Theory Examinations: 30% (iii). Internal Laboratory Examination: 20%	
<b>Mandated Activities with marks:</b> Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).		
<b>Internal Examinations:</b> TWO tests		
<p><b>REFERENCES:</b></p> <ol style="list-style-type: none"> <li>1. Mukhiya, S. K., &amp; Ahmed, U. (2020). Hands-On Exploratory Data Analysis with Python. Packt Publishing.</li> <li>2. Wes McKinney (2022), <i>Python for Data Analysis</i>, O'Reilly Media (3rd Edition)</li> <li>3. Grus, J. (2019). Data science from scratch: First principles with Python. O'Reilly Media.</li> </ol> <p><b>E-RESOURCES:</b></p> <ol style="list-style-type: none"> <li>1. Deep Reinforcement Learning – Prof. Shivaram Kalyanakrishnan, IIT Madras – <a href="https://nptel.ac.in/courses/106106179">https://nptel.ac.in/courses/106106179</a></li> <li>2. <a href="https://www.youtube.com/user/joshstarmer">https://www.youtube.com/user/joshstarmer</a></li> <li>3. <a href="https://www.kaggle.com/learn">https://www.kaggle.com/learn</a></li> <li>4. <a href="https://www.coursera.org/specializations/jhu-data-science">https://www.coursera.org/specializations/jhu-data-science</a></li> <li>5. <a href="https://www.analyticsvidhya.com/blog/2021/06/exploratory-data-analysis-eda-step-by-step/">https://www.analyticsvidhya.com/blog/2021/06/exploratory-data-analysis-eda-step-by-step/</a></li> </ol>		
<p><b>Course Outcomes:</b></p> <p><b>CO1:</b> Apply Exploratory Data Analysis (EDA) techniques using Python libraries on real-world datasets.</p> <p><b>CO2:</b> Analyze data using statistical measures and derive meaningful patterns through appropriate visualizations.</p> <p><b>CO3:</b> Design and implement data wrangling workflows and perform time-series transformations.</p> <p><b>CO4:</b> Evaluate statistical hypotheses and build predictive models using regression techniques.</p> <p><b>CO5:</b> Build and deploy basic data science models and interpret the results to derive business insights.</p>		

<b>CO</b>	<b>CO Description</b>	<b>Mapped POs</b>	<b>Mapped PSOs</b>
<b>CO1</b>	Apply Exploratory Data Analysis (EDA) techniques using Python libraries on real-world datasets.	PO1, PO3	PSO1
<b>CO2</b>	Analyze data using statistical measures and derive meaningful patterns through appropriate visualizations.	PO1, PO3	PSO1, PSO2
<b>CO3</b>	Design and implement data wrangling workflows and perform time-series transformations.	PO1, PO3	PSO1
<b>CO4</b>	Evaluate statistical hypotheses and build predictive models using regression techniques.	PO1, PO3	PSO1, PSO2
<b>CO5</b>	Build and deploy basic data science models and interpret results to derive business insights.	PO1, PO2, PO3	PSO1, PSO2

<b>BD25C01</b>	<b>Prompt Engineering</b>	L	T	P	C
		3	0	0	3

**Course Objective:**

This course introduces the fundamental concepts and importance of Prompt Engineering in the domain of Large Language Models (LLMs). It enables learners to design and refine effective prompts for various AI applications using advanced techniques and tools to improve the quality and control of generated outputs. The course also emphasizes ethical considerations, addressing issues such as bias, hallucination, and misuse in prompt-driven AI systems.

**Introduction to Prompt Engineering:** Overview of Generative AI and LLMs – Fundamentals of Prompt Engineering – Types of Prompts – Basic Prompting Techniques – Prompt Patterns – Hallucination Problems – Importance and Applications – Challenges and Limitations.

**Activities**

1. Presentation on Generative AI and its industrial applications
2. Try basic prompts in ChatGPT to observe varied outputs

**Advanced Prompting Techniques:** System, Contextual, and Role Prompting – In-context Learning – Thought Generation – Problem Decomposition – Self-Criticism – Multimodal Prompting – Image and Audio Prompting – Prompt Optimization – Introduction to Prompt Hub and AI Agents.

**Activities**

1. Case study discussion on in-context learning and zero-shot prompting
2. Design prompt chains and analyze outputs using OpenAI Playground

**Advanced Tools, Frameworks, and Automation:** LangChain – PromptLayer – OpenAI APIs – Hugging Face Tools – Automating Prompt Generation and Evaluation – Integration with Jupyter and GitHub – Scripting Workflows for Prompt Deployment – Building Prompt Evaluation Pipelines.

**Activities**

1. Group demo on LangChain use cases
2. Discussion on scripting approaches for large-scale prompt workflows

**Evaluation and Ethical Considerations:** Metrics for Prompt Evaluation – Relevance, Coherence, Accuracy, and Factuality – Prompt Robustness and Sensitivity – Calibration of AI Outputs – Human-in-the-Loop Evaluation Methods – Explainability and Interpretability of AI Responses – Bias and Fairness in AI Outputs – Adversarial Prompting Techniques – Hallucination Detection – Mitigating Harmful or Misleading Outputs – Risk Assessment, Misuse Scenarios, and Ethical Deployment of Generative AI Systems.

**Activities**

1. Develop “Ethical AI Code of Conduct” in groups
2. Flipped classroom on adversarial prompting and safety risks

**Domain Applications & Case Studies:** Prompting for Healthcare, Education, Finance – Case Studies on Domain-Specific Prompting – Data Security and Privacy in Prompt Engineering – Threats in AI-Generated Content – Responsible Deployment – Future of Prompt Engineering.

**Activities**

1. Poster on future trends and secure applications in prompt engineering
2. Debate on ethical dilemmas and security policies in prompt-based systems
3. Seminar from industry guest on secure GenAI practices

**Real-Time Deployment, Promptops & Research Trends:** LLM App Deployment – PromptOps (Prompt Engineering in Production) – API Integration (OpenAI, Gemini, Claude) – Streamlit & Flask-based Deployment – Prompt Debugging & Logging – Token Usage Optimization – Security & Cost Control in Prompts – Retrieval-Augmented Generation (RAG) – Research Trends: Prompt Tuning vs Instruction Tuning – Open Source LLMs (LLaMA, Mixtral, Gemma) – Future of PromptOps & GenAI Monitoring.

**Activities**

1. Deploy a prompt-based AI assistant using Streamlit or Hugging Face Spaces
2. Analyze cost/token usage for real-time prompt workflows.

**Weightage:** Continuous Assessment: 40%, End Semester Theory Examinations: 60%

**Assessment Methodology:** Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25).

**References :**

1. Vanderbilt University, “Prompt Engineering for ChatGPT”, Coursera, 2024.
2. Indian Institute of Science (IISc) and TalentSprint, “Generative AI and Prompt Engineering”, 2024.
3. IBM, “Introduction to Prompt Engineering”, edX, 2024.
4. James Phoenix and Mike Taylor, Prompt Engineering for Generative AI, O’Reilly Media, 2024.

**E- Resources:**

1. OpenAI Learning, “**ChatGPT Prompting Techniques**”, 2024.  
<https://platform.openai.com/docs/guides/prompt-engineering>
2. LearnPrompting.org, “**Interactive Prompt Engineering Course**”, 2024.  
<https://learnprompting.org/>

3. Google Cloud, “Prompt Engineering for Generative AI”, 2024.  
<https://cloud.google.com/vertex-ai/generative-ai/docs/learn/prompts>
4. Hugging Face, “Prompt Engineering for NLP Models”, 2024.  
<https://huggingface.co/learn/nlp-course/chapter6/6>

<b>CO</b>	<b>Description of CO</b>	<b>PO</b>	<b>PSO</b>
<b>CO1</b>	Describe the principles and types of prompt engineering and their role in generative AI systems.	PO3 (2)	PSO1 (2)
<b>CO2</b>	Design and refine effective prompts using appropriate techniques and tools.	PO1 (2), PO3 (3)	PSO1 (3)
<b>CO3</b>	Apply prompt engineering in solving real-world problems across various domains.	PO1 (3), PO3 (2)	PSO1 (3), PSO2 (2)
<b>CO4</b>	Evaluate prompt performance and ethical implications in AI-generated outputs.	PO1 (2), PO3 (2)	PSO2 (3)
<b>CO5</b>	Demonstrate secure deployment and domain-specific case studies involving prompt engineering.	PO1 (3), PO2 (2), PO3 (2)	PSO1 (2), PSO2 (3)
<b>CO6</b>	Deploy prompt-engineered applications, optimize performance, and explore current trends in generative AI.	PO1 (3), PO2 (2), PO3 (3)	PSO1 (3), PSO2 (2)

CP25C23	Human - Centered AI	L	T	P	C
		3	0	0	3
<p><b>Course Objective:</b>  The course aims to introduce the fundamentals of Human-Centered Artificial Intelligence (HCAI), focusing on its principles, frameworks, and societal impact. This course includes fairness, privacy, reliability, and ethical considerations in the development of intelligent systems. Additionally, it explores governance structures, best practices, and regulatory measures to ensure safe and trustworthy AI deployment.</p>					
<p><b>Introduction To Human-Centered AI</b>  Overview of AI development - Defining Human-Centered AI (HCAI) - Difference between traditional AI and HCAI - Process of design thinking - Cognitive models</p> <p><b>Activity</b></p> <ul style="list-style-type: none"> <li>• Explore the key innovations and challenges involved in human-AI collaboration across different sectors.</li> </ul> <p><b>Human Factors and AI</b>  People and Computers: Are They in the Same Category? - HCAI principles - Impact of AI on society - Importance of human values</p> <p><b>Activity</b></p> <ul style="list-style-type: none"> <li>• Explore the impact of AI on ethics and human centric approaches.</li> </ul> <p><b>Fairness and Bias</b></p> <p>Sources of bias, Group fairness-Exploratory data analysis, limitation of a dataset - Group fairness and Individual fairness-Counterfactual fairness</p> <p><b>Activity</b></p> <ul style="list-style-type: none"> <li>• Use AI platforms to detect and analyze various types of bias</li> </ul> <p><b>Privacy Preservation:</b></p> <p>Attack models-Privacy: Preserving Learning-Differential privacy-Federated learning</p> <p><b>Activity</b></p> <ul style="list-style-type: none"> <li>• Explore why a coin flip is used in sensitive surveys to protect privacy</li> </ul> <p><b>HCAI Framework</b>  Levels of automation - Defining reliable, safe, and trustworthy systems - Two-dimensional HCAI framework: life-critical systems - Golden rules for design - Application examples</p> <p><b>Activity</b></p> <ul style="list-style-type: none"> <li>• Prepare a presentation on Human-Centered AI: Model Explainability with LIME and SHAP</li> </ul>					

### **Design Metaphor**

Science and innovation goals - Intelligent agents - Tele-bots - Assured autonomy: design, control centers - Social robots: active appliances - Enhancing human capabilities

### **Activity**

- Survey and analyze recent research papers emphasizing Human-Centered AI (HCAI)

### **HCAI Governance Structures**

Ethical implications of emerging AI technologies - Software engineering best practices - Establishing safety cultures in organizations - Independent oversight and certification - Role of government regulation

### **Activity**

- Explore national policies, ethical guidelines, and legal frameworks aimed at ensuring the safe and responsible use of Artificial Intelligence across the world

<b>Weightage:</b>	<b>Continuous Assessment: 40%</b>	<b>End Semester Theory Examinations: 60%</b>
	<b>(i) Activity: 10%</b> <b>(ii) Internal Theory Examination: 30%</b>	
<b>Mandated Activities with Marks:</b> Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)		
<b>Internal Examinations: Two Tests</b>		

### **References:**

1. **Human-Centered AI: A New Synthesis**, Ben Shneiderman, 1st Edition, Oxford University Press, 2022.
2. **Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way**, Virginia Dignum, Springer Nature, 2019.
3. **Interpretable Machine Learning**, Christoph Molnar, 1st Edition, Lulu, 2019. [eBook, Available Online]
4. **Designing the User Interface: Strategies for Effective Human-Computer Interaction**, Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs, Niklas Elmqvist, 6th Edition, Pearson Education, 2018.
5. **Interaction Design: Beyond Human-Computer Interaction**, Jenny Preece, Helen Sharp, Yvonne Rogers, 5th Edition, Wiley Student Edition, Wiley, 2019.
6. **Human-Computer Interaction**, Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, 3rd Edition, Pearson Education, 2004.
7. **AI Ethics**, Paula Boddington, Springer, 2023.
8. **The Ethics of Artificial Intelligence: Principles, Challenges, and Opportunities**, Luciano Floridi, Springer, 2023.

**E-Resources:**

1. NPTEL Video Lecture Notes on “Responsible & Safe AI Systems”
2. <https://aclanthology.org/2021.acl-long.330.pdf>
3. <https://arxiv.org/pdf/2004.09456>

**CO-PO-PSO Mapping:**

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, principles, and design philosophies of human-centered artificial intelligence for developing responsible and user-focused AI systems.	--	--
CO2	<b>Analyze</b> human-AI interaction models and ethical considerations to understand usability, transparency, and trust in intelligent systems.	PO1 (3)	PSO1 (3)
CO3	<b>Evaluate</b> human-centered AI approaches, frameworks, and applications to assess their effectiveness in enhancing user experience and societal impact.	PO3 (2)	PSO2 (2)
CO4	<b>Design</b> human-centered AI solutions by selecting appropriate interaction techniques, ethical guidelines, and evaluation strategies for real-world applications.	PO2 (1)	PSO1 (3)

CP25C24	Vibe Coding	L	T	P	C
		3	0	0	3
<p><b>Course Objective:</b></p> <ul style="list-style-type: none"> <li>• This course introduces the basics of AI-assisted coding and Vibe Coding</li> <li>• Participants will learn to use AI tools, build simple applications, and understand the roles and ethics involved.</li> <li>• Students able to create their own AI-powered projects.</li> </ul>					
<p><b>Introduction To Vibe Coding and Human-Ai Collaboration</b></p> <p>Introduction to Vibe Coding-Basic Prompting Techniques for Coding Tasks-Roles and -Responsibilities in Human-AI Collaboration-AI Assistant-Prompt Engineer-Developer-Designing Simple AI-Augmented Workflows-Interaction Between AI-Generated Code and Developer Feedback</p> <p><b>Activities</b></p> <ul style="list-style-type: none"> <li>• <b>Role Play:</b> Students take on roles such as AI, prompt engineer, and developer to simulate collaborative coding.</li> <li>• <b>Flipped Classroom &amp; Quiz:</b> Students learn prompting basics at home and participate in a class quiz and discussion.</li> </ul>					
<p><b>Tools And Platforms for Ai-Assisted Coding</b></p> <p>Overview of AI-Assisted Coding Tools-Cursor, v0, Bolt, etc.-Features and Use Cases of Each Tool-Setting Up and Navigating AI-Coding Platforms-Building Simple Web Applications with AI -Integration-Workflow Prototyping Using AI Tools</p> <p><b>Activities</b></p> <ul style="list-style-type: none"> <li>• <b>Flipped Classroom &amp; Hands-on Tutorial:</b> Students explore AI tools (e.g., Cursor or v0) at home and build a basic app in class.</li> <li>• <b>Team Activity:</b> Groups design a simple development workflow using AI tools and explain tool selection and purpose.</li> </ul>					
<p><b>AI Prompt Design, Oversight, and Ethical Coding Practices</b></p> <p>Vibe Coder's Toolkit: Prompts, Context, and Flow-Prompt Optimization Techniques-Human Oversight in AI Coding Systems-Ethical and Responsible AI Usage-Challenges and Best Practices in AI-Augmented Development</p> <p><b>Activities</b></p> <ul style="list-style-type: none"> <li>• <b>Team Presentation:</b> Each group presents a complete AI workflow, including prompt design, context handling, and oversight strategy.</li> <li>• <b>Ethical Case Discussion:</b> Class discussion on accountability, code quality, and risks of over-reliance on AI in software development.</li> </ul>					
<p><b>The Unbundled Programmer and the Future of Coding Roles</b></p>					

Evolving Developer Roles in the AI Era-The Shift from Coding to Orchestration and Strategy-AI Integration Across the Software Development Lifecycle-Human-AI Collaboration in Debug, Design, and Deployment-Strategic Thinking and Adaptability in AI-Augmented Teams

### Activities

- **Team Presentation:** Showcase how AI transforms developer roles across the SDLC.
- **Role Play & Quiz:** Simulate a modern dev team using AI tools and complete a quiz based on workflow observations.

### Code Quality, Reliability, and Ethical Development

Long-Term Maintainability of AI-Generated Code-Refactoring AI-Assisted Code for Clarity and Standards-Documentation Strategies for Mixed Human-AI Codebases-Common Security and Reliability Issues in Vibe Code-Ethical Coding Practices in AI-Augmented Environments-Navigating Cultural Shifts in 'Vibe Coding' and Accountability

### Activities

- **Tutorial & MCQ:** Refactor example AI-generated code and answer related questions on quality and security.
- **Flipped Classroom:** Study ethical case studies at home and present findings in class.

### Applied Projects and Real-World Ai Applications

AI-Powered Task Manager and To-Do Lists-Resume Builder with Smart AI Suggestions-Note-Taking App with Auto-Organization-Recipe Generator with Ingredient Substitution Logic-Visual Memory Game Using AI Tools-Non-diagnostic Health Symptom Checker

### Activities

- **Team Project:** Build a working prototype (e.g., to-do list, resume builder) using AI-assisted tools.
- **Presentation & Quiz:** Present the final project and complete a quiz on the design and development process.

### Advanced Ai-Driven Web Application Deployment

Planning and Scoping AI-Assisted Applications-Workflow Mapping from Idea to Deployment-Integrating Front-End, Back-End, and AI Logic-Testing and Deploying AI-Augmented Applications-Human Oversight and Continuous Improvement Loops-Project Ownership and Accountability in AI Codebases

### Activities

- **Capstone Project:** Design and deploy a full AI-assisted web application as a team.

- **Demo Day & Reflection:** Teams demo their solutions and reflect on human-AI collaboration challenges.

**Weightage:** Continuous Assessment: 40%, End Semester Theory Examinations: 60%

**Assessment Methodology:** Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

**References:**

1. Ted Winston, Mastering Vibe Coding: Build, Debug, and Ship Software with AI Assistants like Cursor, Replit, and GPT, Kindle Edition, April 25, 2025.
2. Ethan Voss, Vibe Coding: Build & Sell Apps Without Coding Experience: How to Use AI to Create Profitable Apps Even If You've Never Written a Line of Code, Kindle Edition, 26 March 2025.
3. Amit Iyer, Beginner's Guide to Vibe Coding: Exploring Features, Use Cases and Understanding Vibe Coding, Paperback, 19 March 2025.
4. Gene Kim, Steve Yegge, Vibe Coding: Building Production-grade Software With GenAI, Chat, Agents, and Beyond, Paperback, 21 October 2025.
5. Addy Osmani, Beyond Vibe Coding, O'Reilly Media, Inc., August 2025. ISBN: 9798341634756

**E-Resources**

- <https://elearn.nptel.ac.in/shop/iit-workshops/ongoing/computer-science/applied-vibe-coding-workshop/?v=13b5bfe96f3e>
- <https://www.deeplearning.ai/short-courses/vibe-coding-101-with-replit/>

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, philosophy, and practices of vibe coding and its role in modern, collaborative, and creative software development.	--	--
CO2	<b>Analyze</b> vibe coding workflows, tools, and collaborative practices to understand productivity, creativity, and developer experience.	PO1 (3)	PSO1 (3)
CO3	<b>Evaluate</b> vibe coding approaches, platforms, and community-driven practices to assess their effectiveness in real-world software development scenarios.	PO3 (2)	PSO2 (2)
CO4	<b>Design</b> software development solutions using vibe coding principles by selecting appropriate tools, collaboration strategies, and creative workflows.	PO2 (1)	PSO1 (3)

<b>CP25C25</b>	<b>Federated Learning</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		3	0	0	3

**Course Objective:**

1. Understand the fundamentals, architecture, and real-world relevance of Federated Learning.
2. Learn core FL algorithms, privacy techniques, and system challenges in decentralized settings.
3. Apply FL frameworks and tools to build practical applications through hands-on project work.

**Introduction to Federated Learning**

Definition and Scope – Motivation: Data Privacy, Decentralization, Compliance (GDPR, HIPAA) – Centralized vs. Federated vs. Distributed Learning – Types of FL: Horizontal, Vertical, and Transfer FL – Real-World Applications: Mobile Keyboard (GBoard), Healthcare, IoT – FL System Architecture – Federated Learning Workflow and Lifecycle – Communication Protocols.

**Activity**

Create a Quiz for understanding Federated Learning Basics

**Federated Learning Algorithms**

Federated Averaging (FedAvg) – Federated SGD (FedSGD) – FedProx for handling system heterogeneity – Personalized Federated Learning (pFedMe, FedPer, FedBN) – FL with Non-IID Data – Aggregation and Weight Update Strategies – Convergence Issues – Performance Metrics.

**Activity**

Create a flipped class room for Understanding federated learning algorithms

**Privacy and Security in FL**

Security Threats in FL – Differential Privacy (DP): Concept, Application in FL – Secure Multiparty Computation (SMC) – Homomorphic Encryption (Overview) – Secure Aggregation Protocols – Byzantine Robustness – Model Inversion Attacks – Data Poisoning – Mitigation Strategies.

**Activity**

Ask students to submit an assignment on Privacy and security in FL

**System & Communication Challenges**

Device Heterogeneity – Intermittent Availability – Resource Constraints (Memory, Battery, Network) – Client Selection Strategies – Communication Efficiency: Quantization, Compression, Sparsification – Dropout and Partial Client Participation – FL over Edge Devices – Energy-aware FL.

**Activity**

Create a flipped class room communication challenges

**Applications and Frameworks**

FL Applications in Healthcare, Finance, Smart City, Recommender Systems – Case Studies – Tools and Frameworks: TensorFlow Federated, PySyft, Flower – FL Simulators

**Activity**

Ask students to submit an Assignment on applications of FL

**Project Work and Future Directions**

Project Work: Building a Simple Federated Learning Model – Model Deployment and Evaluation – Future Directions: Split Learning, Hybrid FL, Federated Analytics, FL + AI/ML

**Activity**

Ask students to submit simple project on FL

**References:**

1. Q. Yang, Y. Liu, T. Chen, and Y. Tong, *Federated Learning*, PHI Learning Pvt. Ltd., 2021.
2. M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*, 2nd ed., Pearson Education, 2020.
3. K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, Pearson Education, 2017.
4. C. M. Bishop, *Pattern Recognition and Machine Learning*, Oxford University Press, 2016.
5. T. Mitchell, *Machine Learning*, Tata McGraw-Hill Education, 2017.

<b>Weightage:</b>	<b>Continuous Assessment: 40%</b>	<b>End Semester Theory Examinations: 60%</b>
	<b>(i) Activity: 10%</b> <b>(ii) Internal Theory Examination: 30%</b>	

**Mandated Activities with Marks:**

Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

**Internal Examinations: Two Tests****E-resources:**

1. <https://youtu.be/E3Kn5v-vdUc?si=UMnGqSNOB9NjheTS-> NPTEL YouTube– Lecture 42: Introduction to Federated Learning
2. <https://github.com/adap/flower-> Flower Framework GitHub
3. <https://github.com/OpenMined/PySyft-> PySyft (OpenMined)

<b>CO</b>	<b>Description of CO</b>	<b>PO</b>	<b>PSO</b>
<b>CO1</b>	Describe the fundamental concepts, architectures, and operational principles of federated learning and privacy-preserving machine learning systems.	--	--
<b>CO2</b>	<b>Analyze</b> federated learning algorithms and communication strategies to understand model convergence, privacy, and system performance.	PO1 (3)	PSO1 (3)
<b>CO3</b>	<b>Evaluate</b> federated learning frameworks, security mechanisms, and deployment approaches to assess their effectiveness in distributed and sensitive data environments.	PO3 (2)	PSO2 (2)
<b>CO4</b>	<b>Design</b> federated learning solutions by selecting appropriate aggregation methods, communication protocols, and privacy techniques for real-world applications.	PO2 (1)	PSO1 (3)