

**ANNA UNIVERSITY, CHENNAI**  
**NON- AUTONOMOUS COLLEGES**  
**AFFILIATED TO ANNA UNIVERSITY**  
**M.E. BIG DATA ANALYTICS**  
**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:** Data-Driven Decision Making and Visualization: Apply statistical, machine learning, and deep learning techniques to extract meaningful insights, build predictive models, and create interactive visualizations for informed decision-making in diverse domains.

**PSO2:** Research, Innovation, and Ethical Practice: Conduct research and develop innovative solutions in big data analytics addressing real-world challenges, while ensuring data privacy, security, and ethical use of data in alignment with global standards.



**ANNA UNIVERSITY, CHENNAI**

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

**Programme:** M.E. Big Data Analytics

**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.	BD25101	Applied Data Science with R Programming	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.	BD25102	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.	BD25201	Big Data Technologies	T	3	0	0	3	3	ES (PC)
2.	CP25C05	Artificial Intelligence and Machine Learning	T	3	0	0	3	3	ES (PC)
3.	BD25202	Python for e-Governance Practices	LIT	3	0	2	5	4	ES (PC)
4.		Programme Elective I	T	3	0	0	3	3	ES (PE)
5.	CP25C07	Quantum Computing	T	2	0	0	2	2	ES (PC)
6.		Industry-Oriented Course I	-	1	0	0	1	1	SD
7.	BD25203	Industrial Training	-	-	-	-	-	2	SD
8.		Self-Learning Course	-	-	-	-	-	1	-
<b>Total Credits</b>							<b>17</b>	<b>19</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.	BD25301	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.	BD25401	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	Periods per week			Total Contact Periods	Credits
			L	T	P		
1.	BD25001	Business Statistics and Data Modelling	3	0	0	3	3
2.	BD25C01	Prompt Engineering	3	0	0	3	3
3.	BD25002	Digital Marketing Analytics	3	0	0	3	3
4.	BD25003	Web Analytics and Development	3	0	0	3	3
5.	BD25004	Optimization Techniques for Analytics	3	0	0	3	3
6.	BD25005	Data Lake Architecture and Applications	3	0	0	3	3
7.	BD25006	Data Modelling and Integration	3	0	0	3	3
8.	BD25007	Data Acquisition and Production Management Practices	3	0	0	3	3
9.	CP25C13	Quantum Machine Learning	3	0	0	3	3
10.	CP25C17	Edge and Fog Computing	3	0	0	3	3
11.	BD25008	Data Intensive Computing	3	0	0	3	3
12.	BD25009	Deep Learning ecosystem	3	0	0	3	3
13.	CP25C19	Cognitive Computing	3	0	0	3	3
14.	CP25C21	Mixed Reality	3	0	0	3	3
15.	BD25010	Data Manipulation and Visualization	3	0	0	3	3
16.	BD25C02	Exploratory Data Analysis	3	0	0	3	3
17.	BD25011	Geospatial Analytics	3	0	0	3	3

S. No.	Course Code	Course Title	Periods per week			Total Contact Periods	Credits
			L	T	P		
18.	CP25C25	Federated Learning	3	0	0	3	3
19.	CP25C24	Vibe Coding	3	0	0	3	3
20.	CP25C20	Agentic AI	3	0	0	3	3

MA25C07	Advanced Mathematical Methods (CSIE)	L	T	P	C
		3	1	0	4
<b>Course Objectives:</b>					
<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>					
<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.					
<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.					
<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.					
<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.					
<b>Weightage:</b> Continuous Assessment: 40%, End Semester Examinations: 60%.					
<b>Assessment Methodology:</b> Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20).					
<b>References:</b>					
<ol style="list-style-type: none"> <li>1. Gilbert Strang, <i>Linear Algebra and Its Applications</i>, Cengage Learning.</li> <li>2. Richard A. Johnson &amp; Dean W. Wichern, <i>Applied Multivariate Statistical Analysis</i>, Pearson.</li> <li>3. Neal Koblitz, <i>A Course in Number Theory and Cryptography</i>, Springer.</li> <li>4. Victor Shoup, <i>A Computational Introduction to Number Theory and Algebra</i>, Cambridge University Press.</li> </ol>					
<b>E-resources:</b>					
<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> <ol style="list-style-type: none"> <li>1. To explore advanced linear, tree, and graph data structures and their applications.</li> <li>2. To design efficient algorithms using appropriate algorithmic paradigms.</li> <li>3. To evaluate computational complexity and identify tractable vs. intractable problems.</li> </ol>					
<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>					

**Algorithm Design and Paradigms:** 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.

**Practical:**

- Implement Strassen's algorithm and compare with naive matrix multiplication.
- Develop a randomized algorithm for primality testing (Miller–Rabin).

**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.
- 6.

**E-resources:**

1. <https://www.theiotacademy.co/blog/data-structures-and-algorithms-in-c/>
2. [https://github.com/afid18/Data\\_structures\\_and\\_algorithms\\_in\\_cpp](https://github.com/afid18/Data_structures_and_algorithms_in_cpp)
3. <https://www.udemy.com/course/introduction-to-algorithms-and-data-structures-in-c/?srsltid=AfmBOorEZlkgV7QzaEh6lqzAaKLjC-lpFU1NGgWFOHMLhOos-uDVKjCK>

	<b>Description of CO</b>	<b>PO</b>	<b>PSO</b>
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)
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)

BD25101	Applied Data Science with R Programming	L	T	P	C
		3	0	0	3
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>To equip students with the ability to apply R programming in real-world data science projects.</li> <li>To enable learners to implement statistical methods, machine learning models, and exploratory data analysis using R.</li> <li>To develop proficiency in handling datasets and effectively communicating data-driven findings.</li> </ul>					
<p><b>Introduction to R programming:</b> R programming Structures, vectors, matrices and arrays, lists, data frames, reading and writing data, connection Interfaces, subsetting R objects, dplyr package, control structures, Functions, scoping rules of R, Looping functions.</p> <p><b>Activity:</b> Demonstration of reading data, subsetting using dplyr live in RStudio Cloud.</p>					
<p><b>Extended Programming in R:</b> Data manipulation techniques - Math and simulations in R, Object-oriented programming, Input and Output, String Manipulation, Debugging, Performance Enhancement, Speed and Memory, Interfacing R to other languages, Parallel R.</p> <p><b>Activity:</b> Demonstration of R object types with their OOP class systems.</p>					
<p><b>Linear algebra for data science using R:</b> Linear Combinations - Vector Spaces, Fundamental Theorem of Linear Algebra, Least Squares, Change of Basis, Orthogonality, Eigen Problems.</p> <p><b>Activity:</b> Visualizations of vector spaces and Orthogonality.</p>					
<p><b>Statistics for Data Science:</b> Descriptive statistics, Notion of probability, Distributions, Mean, Variance, Covariance, Covariance matrix, understanding Univariate and Multivariate Normal distributions, hypothesis testing, Confidence interval for estimates, Optimization, Typology of data science problems and a solution framework.</p> <p><b>Activity:</b> Simulate sampling distributions with various sample sizes.</p>					
<p><b>Statistical Applications using R programming:</b> Time series data handling, power and computation of sample size, advanced data handling, Multiple regression, Survival Analysis, Rates and Poisson Regression, Nonlinear curve fitting, statistical modelling in R, Advanced Non-Linear Modelling and Evaluation: Quantile Regression Method.</p> <p><b>Activity:</b> Demonstration of Data handling and curve fitting.</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. Peng, R. D. (2016). R programming for data science.
2. Mitran, S. Linear algebra for data science. University of North Carolina at Chapel Hill.
3. Bennett, S. R. (2021). Linear algebra for data science with examples in R.
4. Montgomery, D. C., & Runger, G. C. Applied statistics and probability for engineers. John Wiley & Sons.
5. Wickham, H. R for data science. O'Reilly Media.

	<b>Description of CO</b>	<b>PO</b>	<b>PSO</b>
CO1	Apply R programming concepts to manipulate, transform, and visualize data effectively.	--	--
CO2	Analyze linear algebra concepts and statistical methods to solve data science problems using R.	PO1(3)	PSO1(3)
CO3	Evaluate the performance of statistical and machine learning techniques on real-world datasets.	PO3(2)	PSO2(2)
CO4	Design data-driven solutions integrating R programming for complex problems.	PO2(1)	PSO1(3)

CP25C03	Advanced Operating Systems	L	T	P	C
		3	0	0	3
<b>Course Objectives:</b> <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>					
<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 <b>Activity:</b> CPU scheduler simulation for multicore systems.					
<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 <b>Activity:</b> Simulate demand paging and page replacement algorithms.					
<b>Virtualization and Containerization:</b> Hypervisors (Type I & II), KVM, QEMU, Xen-Containers: Docker, LXC, systemd-nspawn-OS-level virtualization and namespaces <b>Activity:</b> Deploy and configure Docker containers with various images.					
<b>Distributed Operating Systems and File Systems:</b> Distributed scheduling, communication, and synchronization-Distributed file systems: NFS, GFS, HDFS-Transparency issues and fault tolerance <b>Activity:</b> Simulate distributed process synchronization.					
<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 <b>Activity:</b> Implement Role-Based Access Control (RBAC) using Linux user and group permissions.					
<b>Real-Time and Embedded Operating Systems:</b> Real-time scheduling algorithms (EDF, RM)-POSIX RT extensions, RTOS architecture-TinyOS, FreeRTOS case studies <b>Activity:</b> Analyze FreeRTOS task scheduling behavior.					
<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) <b>Activity:</b> Analyze Android's system architecture using emulator tools.					

**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. 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>

	<b>Description of CO</b>	<b>PO</b>	<b>PSO</b>
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>Convert source code to SSA form using LLVM IR.</li> <li>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>Perform loop unrolling and strength reduction.</li> <li>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>Implement loop tiling and loop skewing on a matrix multiplication program.</li> <li>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>Develop a basic JIT-enabled interpreter with LLVM or GraalVM.</li> <li>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)

BD25201	Big Data Technologies	L	T	P	C
		3	0	0	3
<p><b>Course Objective:</b></p> <p>This course will help to understand what big data is and how Big Data Technologies can help to achieve a competitive advantage in various applications with the help of different tools.</p>					
<p><b>Overview of Big Data</b></p> <p>Introduction – Types of Big Data – Big Data Characteristics – Big Data Integration – Analysis of Big Data – Applications of Big Data Analytics – Types of Big Data Analytics – Comparison of Data Analytics stages</p> <p><b>Activities:</b></p> <ol style="list-style-type: none"> <li>1. Identify and classify data sources and types.</li> <li>2. Map Big Data characteristics (5Vs).</li> <li>3. Propose a data integration and analytics workflow.</li> <li>4. Identify analytics approaches and tools (conceptual).</li> <li>5. Classify analytics types used.</li> <li>6. Compare traditional and Big Data analytics stages.</li> </ol>					
<p><b>Hadoop Ecosystems</b></p> <p>Components of Hadoop – HDFS Architecture – Map Reduce Architecture – Map Reduce Programming – Optimizing Map Reduce Tasks – Hadoop Ecosystems - Types of NoSQL Databases – Principles of NoSQL Data Models – Advantages and Disadvantages</p> <p><b>Activities:</b></p> <p><b>Project-based learning</b></p> <ol style="list-style-type: none"> <li>1. Hadoop-Based Big Data Processing System.</li> <li>2. NoSQL Database Design for Big Data Applications</li> </ol>					
<p><b>Data Lakes</b></p> <p>Architecture of Data Lakes – Usage of Data Lakes – Data Lake Challenges – Lake House – Best Practices</p> <p><b>Activity:</b></p> <p><b>Quiz/Puzzle</b> - Use online platforms like Socrative or Google Forms to quickly assess student understanding through polls and quizzes.</p>					
<p><b>Big Data Technology and Tools</b></p> <p>Overview of Spark – Spark Architecture – Job Execution in Spark – Sparklyr – Storm</p>					

Architecture – Working with RStorm

**Activity:**

**Seminar/Reproduction of Research Paper**

**Big Data Infrastructure**

Architecture of Data-intensive Computing – Clusters – Parallel Distributed Architecture – Sharding and Replication - Big Data Management in Clouds – Big Data Virtualization

**Activity:**

**Flipped Classroom, Poster Presentation** - students create visual representations of big data infrastructure concepts and their relationships, like a map of different data storage technologies and their uses. This helps them organize information and identify areas of confusion.

**Big Data Applications**

Big Data for Health Care – Big Data Analytics for Fraud Detection – Big Data Analytics in Social Media – Applications of IoT-based Big Data - Novel Applications and Research Directions

**Activity:**

**Case Studies** - Present students with real-world case studies of big data applications. Have them analyze the data, identify key insights, and discuss their findings. This could involve using specific big data tools (e.g., Hadoop, Spark) or focusing on the analytical techniques.

**Course Outcomes:**

1. Understand the fundamentals of Big Data Technologies.
2. Demonstrate the MapReduce programming model to process big data along with Hadoop tools and NoSQL data models.
3. Investigate the Data Lake architecture and its challenges.
4. Implement Big Data code in Spark and Storm.
5. Use of Big Data management in cloud and virtualization.
6. Perform big data analytics for various applications like healthcare, Fraud detection and Social Media.

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

**Mandated Activities with marks:**

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

**REFERENCES**

1. Big Data Concepts, Technologies and Applications by Mohammad Shahid Husain, Mohammad Zunnun Khan, Tamanna Siddigui, CRC Press, 2020
2. Big Data Analytics: Concepts, Techniques, Tools and Technologies by M. Thangaraj, S. Suguna, G. Sudha, Prentice Hall India (PHI), 2021
3. Big Data Analytics by G.Sudha Sadasivam, R. Thirumal, Oxford University Press, 2020
4. Raj Kama! and Preeti Saxena, "Big Data Analytics Introduction to Hadoop, Spark, and Machine-Learning", McGraw Hill Education, 2018
5. Arshdeep Bahga, Vijay Madiseti, "Big Data Analytics: A Hands-On Approach", 1st Edition, VPT Publications, 2018.

**E-Resources:**

1. [https://onlinecourses.nptel.ac.in/noc20\\_cs92](https://onlinecourses.nptel.ac.in/noc20_cs92)

<b>CO</b>	<b>Description of CO</b>	<b>PO</b>	<b>PSO</b>
<b>CO1</b>	<b>Understand</b> the fundamentals of big data, its characteristics, types, integration, and applications across various domains.	--	--
<b>CO2</b>	<b>Analyze</b> Hadoop ecosystems, MapReduce programming, and NoSQL databases to understand big data processing and storage solutions.	PO1 (3)	PSO1 (3)
<b>CO3</b>	<b>Evaluate</b> Data Lake architectures, big data tools like Spark and Storm, and cloud-based big data infrastructure for their effectiveness and challenges.	PO3 (2)	PSO2 (2)
<b>CO4</b>	<b>Design</b> end-to-end big data solutions for real-world applications by implementing Spark/Storm programs, data management strategies, and analytics workflows.	PO2 (1)	PSO1 (3)

CP25C05	Artificial Intelligence and Machine Learning	L	T	P	C
		3	0	0	3
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>To understand the fundamentals of Artificial Intelligence and its core problem-solving techniques using intelligent agents, search strategies, and logic-based reasoning.</li> <li>To comprehend the theoretical foundations and algorithmic frameworks of machine learning, with an emphasis on supervised and unsupervised learning methodologies.</li> </ul>					
<p><b>Problem Solving and Knowledge Representation:</b> Solving problems by searching – adversarial search – constraint satisfaction problems – Logical agents – Propositional logic – First order logic – Forward chaining – Backward chaining – Ontological Representations and Reasoning Systems.</p> <p><b>Activities:</b> Develop a simple Intelligent Agent that navigates a grid world using search algorithms.</p>					
<p><b>Uncertain Knowledge and Reasoning:</b> Overview of uncertainty and basic probability – Baye’s rule – Bayesian networks – Hidden Markov models – Kalman filters – Utility Functions – Decision Networks – Sequential decision problems – Game theory</p> <p><b>Activities:</b> Develop a simple Intelligent Agent that makes decisions in adversarial conditions</p>					
<p><b>Machine Learning Introduction:</b> Types of learning – Hypothesis space – Inductive bias – Evaluation – Cross validation – Bias variance – Bias Variance Trade off.</p> <p><b>Activities:</b> Develop a simple Intelligent Agent that uses logical reasoning to infer hidden facts or constraints</p>					
<p><b>Supervised Learning:</b> Linear Regression – Logistic Regression - Decision trees: Classification and regression trees – Neural networks, multilayer perceptron – Support vector machines: linear and non-linear kernel functions – K nearest neighbours – Ensemble learning: bagging and boosting.</p> <p><b>Activities:</b> Behaviour Modelling, Disease Diagnosis and any other similar ones suitable for supervised learning</p>					
<p><b>Unsupervised and Probabilistic Learning:</b> Clustering: partition, hierarchical and density based clustering – Self organizing maps – Expectation maximization – Gaussian mixture models – Principal component analysis – Bayesian learning: Bayes optimal classifier, Naïve Bayes classifier, Bayesian belief networks.</p> <p><b>Activities:</b> Customer Segmentation, Image Processing and any other similar ones suitable for unsupervised and probabilistic learning</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>					

**References:**

1. Stuart Russell and Peter Norvig Artificial Intelligence - A Modern Approach, Prentice Hall, 3rd edition, 2011.
2. Elaine Rich, Kevin Knight and Shiv Shankar B. Nair, Artificial Intelligence, 3rd edition, Tata McGraw Hill, 2009.
3. Tom Mitchell, "Machine Learning", McGraw Hill, 3rd Edition, 1997.
4. Ethem Alpaydin, "Introduction to Machine Learning", MIT Press, Prentice Hall of India, Third Edition 2014.
5. Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar "Foundations of Machine Learning", MIT Press, 2012.

**E-resources:**

1. Nptel Course, An Introduction to Artificial Intelligence by Prof Mausam
2. Nptel Course, Machine Learning, By Prof. Carl Gustaf Jansson

<b>CO</b>	<b>Description of CO</b>	<b>PO</b>	<b>PSO</b>
<b>CO1</b>	Describe the fundamental concepts, scope, and applications of artificial intelligence and machine learning in solving real-world problems.	--	--
<b>CO2</b>	Analyze various artificial intelligence and machine learning models to understand their behavior, performance, and suitability for different problem domains.	PO1 (3)	PSO1 (3)
<b>CO3</b>	Evaluate machine learning algorithms and artificial intelligence techniques based on accuracy, efficiency, and applicability to specific use cases.	PO3 (2)	PSO2 (2)
<b>CO4</b>	Design intelligent systems by selecting and integrating appropriate machine learning models and AI techniques to address complex real-world applications.	PO2 (1)	PSO1 (3)

BD25202	Python for e-Governance Practices	L	T	P	C
		3	0	2	4

**Course Objective:**

The course is to practice E-governance tasks. The major part is to automate the process, such as data collection, storage, analysis, and visualization, using Python

**Python essentials**

Functions - Strings - Lists - Tuples - Sets – Dictionary

**Practical:**

1. Create a function that takes two numbers and returns their sum, difference, product, and quotient.
2. Count the number of vowels and consonants in a given string.
3. Print all even numbers from a list using a loop.
4. Count the occurrences of a specific element in a tuple.
5. Iterate through a dictionary and print keys and values separately.

**Large Data Management and Basic Analysis**

Pandas - data manipulation, data cleaning, data analysis, data frames, NumPy - handling arrays and matrices, statistical analysis. Case study - E-Governance Data Set management of the existing system.

**Practical:**

1. Create a DataFrame from a dictionary and display it. Handle missing values in a DataFrame.
2. Calculate basic statistics of a DataFrame column and Add a new column based on existing columns.
3. Create a 2D array and perform basic operations and Calculate mean and standard deviation of an array.
4. Load CSV, explore first few records, and find total entries.

**Large Data Analysis and Visualization**

SciPy - math functions, optimization algorithms, ScikitLearn - ML algorithms - regression, classification, clustering, and Time Series Forecasting Assignment - E-Governance Data Set analysis.

**Practical:**

1. Calculate factorial, square root, and exponential using SciPy.
2. Find the minimum of a simple function.

3. Perform linear regression on sample data.
4. Classify points using K-Nearest Neighbors (KNN).
5. Perform K-Means clustering on 2D data.
6. Forecast using a simple moving average.
7. **Assignment:** E-Governance Dataset Analysis

**Experiment:**

1. Load the dataset using Pandas.
2. Explore and clean the data.
3. Apply **linear regression** to predict a numeric feature (e.g., number of services used).
4. Perform **clustering** to group regions based on usage patterns.
5. Visualize trends with **line plots, bar charts, or histograms**.

**Data Visualization and Web Application Development**

Matplotlib and SeaBorn - charts and plots Django Framework - Building complex applications, user authentication, database interactions, API Development; Flask Framework - lightweight application development Mini project: Development of an e-governance application.

**Practical:**

1. Create a simple line chart using Matplotlib and a bar plot using Seaborn
2. i) Create a Django project and app ,ii) Implement user registration and login (Authentication), **iii)** Connect to a database (SQLite/MySQL) and perform CRUD operations using models. iv) Create a simple API endpoint using Django REST framework.
3. Create a basic web app using Flask Framework (Lightweight Applications)
4. Mini Project – E-Governance Application
  - (i). **Front-end:** HTML/CSS forms for citizen registration, feedback, or service requests.
  - (ii). **Back-end:**
    - Use Django or Flask to manage user authentication.
    - Create database models for citizens and services.
  - (iii). **Data Visualization:**
    - Use Matplotlib / Seaborn to display statistics (e.g., number of services used per region).

(iv). **API Development:**

- Create REST APIs to fetch citizen data for dashboard visualization.

(v). **Deployment:**

- Run locally and test functionalities (CRUD, login, dashboards).

**E-governance Essentials**

Data Governance - IT Governance - Model Governance - Project Governance - Risk Governance Case Study: TN Govt. E-Governance Portal.

**Practical**

1. Analyze sample dataset for completeness and consistency.
2. Map IT services of a government portal (login system, forms submission, database) and check **uptime/logging metrics**.
3. Check a simple model for accuracy before deployment
4. Create a simple **Gantt chart** for a portal development project.
5. Evaluate risks in a sample dataset.

**Weightage:**

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

**Assessment Methodology:**

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

**References:**

1. Samir Madhavan, Mastering Python for Data Science, 2015
2. Nathan E. Myers, Gregory Kogan, Self-Service Data Analytics and Governance for Managers, Wiley, 2021
3. Chester Alexis C. Buama, Best Practices in E-Governance, Society Publishing, 2018
4. Alex Berson, Larry Dubov, Master Data Management and Data Governance. MGH, 2011
5. Dr. Devaiah N G, Keerthiraj, E-Governance in India, OrangeBooks Publication, 2021

**E-Resources:**

Swayam courses

- Machine Learning with Python for Managers - Dr. David Campbell, Manipal University
- Python for Data Science - Prof. Ragnathan Rengasamy, IIT Madras.

CO	Description of CO	PO	PSO
CO1	Apply the various Python programming essentials to solve given problems.	--	--
CO2	Apply the various data management techniques to analyze the given e-governance dataset.	PO1 (3)	PSO1 (3)
CO3	Apply machine learning and optimization algorithms to evaluate the given dataset and derive insights.	PO3 (2)	PSO2 (2)
CO4	Design and develop an e-governance application for the given scenario using Django/Flask framework.	PO2 (1)	PSO1 (3)
CO5	Apply a suitable e-governance process to solve the given problem effectively.	PO1 (2)	PSO1 (3)

CP25C07	Quantum Computing	L	T	P	C
		2	0	0	2
<b>Course Objective:</b>					
<ol style="list-style-type: none"> <li>1. To provide a mathematical foundation for Quantum Computing and provide the basics of working</li> <li>2. To interpret the various aspects and applications of quantum computing.</li> <li>3. To examine the factors that affect Quantum computation</li> </ol>					
<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>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).		
<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>		

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.	--	--

<b>CO2</b>	Analyze quantum algorithms and quantum circuit models to understand their computational advantages, limitations, and performance characteristics.	PO1 (3)	PSO1 (3)
<b>CO3</b>	Evaluate quantum computing paradigms, error correction techniques, and hardware technologies for their suitability in solving complex computational problems.	PO3 (2)	PSO2 (2)
<b>CO4</b>	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)

<b>BD25001</b>	<b>Business Statistics and Data Modelling</b>	L	T	P	C
		3	0	0	3
<b>Course Objectives:</b>					
<ul style="list-style-type: none"> <li>To understand statistical techniques essential for business decision-making.</li> <li>To develop data modelling skills to analyze and solve business problems.</li> <li>To apply tools and techniques to extract insights from data for informed decisions.</li> </ul>					
<b>Descriptive Statistics and Data Visualization:</b> Types of data: Categorical, Numerical- Frequency distribution, Measures of central tendency (Mean, Median, Mode)-Measures of dispersion (Range, Variance, Standard Deviation)-Graphical representation: Bar charts, Histograms, Pie charts, Boxplots-Data summarization using Excel/R/Python					
<b>Activity:</b> Flipped class room, quiz					
<b>Probability and Business Applications:</b> Basics of probability and counting techniques-Conditional Probability and Bayes' Theorem-Discrete and Continuous distributions: Binomial, Poisson, Normal-Business decision scenarios involving risk and uncertainty					
<b>Activity:</b> Case study					
<b>Inferential Statistics:</b> Sampling techniques and sampling distributions- Estimation: Point and interval estimates-Hypothesis testing: Z-test, t-test, chi-square test-One-way and two-way ANOVA-Applications in quality control, market research					
<b>Activity:</b> Tutorial, Quiz					
<b>Business Data Modelling:</b> Introduction to data modelling and model types- Correlation and simple linear regression-Multiple linear regression and model					

<p>diagnostics-Time series components and forecasting models-Model evaluation metrics (R<sup>2</sup>, MSE, RMSE)</p> <p><b>Activity:</b> Flipped classroom</p>
<p><b>Predictive Analytics and Business Insights:</b> Introduction to classification (logistic regression)-Decision trees and business use cases-Clustering basics (K-means) for customer segmentation-Scenario analysis and what-if modelling</p> <p>Case studies using software (Excel Solver, R, Python, or Power BI)</p> <p><b>Activity:</b> Case study, seminar</p>
<p><b>Weightage:</b> Continuous Assessment: 40%, End Semester Theory Examination: 60%</p>
<p><b>Mandated Activities with marks:</b></p> <p>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>1. Richard Levin &amp; David Rubin, Statistics for Management, Pearson New International Edition, 2013</li> <li>2. James R. Evans – Business Analytics,3rd edition, Pearson, 2021</li> <li>3. Andy Field – Discovering Statistics Using R,SAGE,2012</li> <li>4. Albright &amp; Winston – Business Analytics: Data Analysis and Decision Making, 8th edition, Cengage,2024.</li> </ol>
<p><b>E-Resources:</b></p> <ol style="list-style-type: none"> <li>1. <a href="https://openstax.org">https://openstax.org</a></li> <li>2. <a href="https://nptel.ac.in">https://nptel.ac.in</a></li> <li>3. <a href="https://ocw.mit.edu">https://ocw.mit.edu</a></li> </ol>

CO	Description of CO	PO	PSO
CO1	Summarize and visualize business data using descriptive and graphical statistical methods to support decision making.	PO1 (2)	PSO1 (2)
CO2	Apply probability theory—including Bayes’ theorem—and appropriate sampling techniques (binomial, normal, Poisson) to assess and forecast business scenarios.	PO1 (3), PO2 (2)	PSO1 (3)
CO3	Conduct hypothesis testing, confidence interval estimation, ANOVA, and linear regression analysis to support business control and strategy decisions.	PO1 (3), PO2 (2), PO3 (2)	PSO1 (3) PSO2 (2)
CO4	Build and interpret predictive data models—multiple regression, logistic regression, and time series forecasting—for business and forecasting use cases.	PO1 (3), PO2 (3), PO3 (2)	PSO1 (3) PSO2 (2)
CO5	Design conceptual, logical, and physical data models (ER diagrams, star schema) to structure business data in decision support systems.	PO2 (3), PO3 (3)	PSO1 (3) PSO2 (2)
CO6	Implement statistical and data models using appropriate software tools (Python/R/Excel/TIBCO) and communicate results through dashboards or reports.	PO1 (2), PO2 (2), PO3 (2)	PSO1 (3) PSO2 (2)

BD25C01	Prompt Engineering	L	T	P	C
		3	0	0	3
<p><b>Course Objective:</b>  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.</p>					
<p><b>Introduction to Prompt Engineering:</b> 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.</p>					
<p><b>Activities</b></p> <ol style="list-style-type: none"> <li>1. Presentation on Generative AI and its industrial applications</li> <li>2. Try basic prompts in ChatGPT to observe varied outputs</li> </ol>					
<p><b>Advanced Prompting Techniques:</b> 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.</p>					
<p><b>Activities</b></p> <ol style="list-style-type: none"> <li>1. Case study discussion on in-context learning and zero-shot prompting</li> <li>2. Design prompt chains and analyze outputs using OpenAI Playground</li> </ol>					
<p><b>Advanced Tools, Frameworks, and Automation:</b> 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.</p>					
<p><b>Activities</b></p> <ol style="list-style-type: none"> <li>1. Group demo on LangChain use cases</li> <li>2. Discussion on scripting approaches for large-scale prompt workflows</li> </ol>					
<p><b>Evaluation and Ethical Considerations:</b> 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.</p>					

**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)

<b>BD25002</b>	<b>Digital Marketing Analytics</b>	L	T	P	C
		3	0	0	3

**Course Objective:**  
This course equips students with advanced analytical tools and techniques to evaluate digital marketing strategies. Students will gain hands-on experience in analyzing online consumer behavior, campaign performance, and ROI using modern tools and real-world data.

**Introduction to Digital Marketing Analytics and Web Analytics**  
Evolution of Digital Marketing - Role of Analytics in Digital Strategy - Types of Digital Data (Owned, Earned, Paid) - Metrics vs KPIs - Google Analytics: Setup and Fundamentals - Traffic, Bounce Rate, Sessions, Pageviews - Goal Setting and Funnel Visualization - Attribution Models

**Activity:**  
Analyze Website Performance Using Google Analytics (GA4 Demo Account). Students will explore and analyze traffic metrics using the Google Analytics 4 (GA4) Demo Account to understand core web analytics concepts.

**Instructions:**

1. Log into the GA4 Demo Account.
2. Navigate to the Reports > Acquisition > Traffic acquisition section.
3. Analyze and record the following for the past 30 days:
  - Total sessions
  - Bounce rate
  - Traffic sources (Direct, Organic, Paid, Social)
  - Avg. session duration
4. Identify one traffic source with high bounce rate and suggest 2 improvements.

**Search Engine Optimization (Seo) & Sem Analytics**  
Keyword Research and Analytics - On-page & Off-page SEO Metrics - Google Search Console - Google Ads Analytics (PPC Campaign Metrics)

**Activity:**  
Google Search Console Report Analysis. Use a sample Google Search Console report (or screenshots) to interpret SEO performance.

**Instructions:**

1. Review a given screenshot or CSV of GSC Performance report.
2. Identify:
  - Top 5 queries with highest CTR

- **Pages with declining impressions**
- **Any mobile usability issues**

**3. Suggest 2 on-page SEO improvements based on findings.**

**Email, Content Marketing, Campaign and ROI Analysis**

Metrics: Open Rate, CTR, Conversion, Unsubscribe - A/B Testing for Emails - Content Engagement Metrics - Tools: Mailchimp, HubSpot - Marketing Funnels and Conversion Tracking - CAC, CLV, ROAS, and ROI Calculation - Marketing Mix Modeling - Cohort and Funnel Analysis

**Activity:**

Calculate CAC, CLV, ROAS from Campaign Data. Use real or provided marketing data to compute campaign ROI metrics.

**Instructions:**

**1. Given a dataset with the following:**

- **Spend per channel**
- **No. of new customers**
- **Avg. order value**
- **Repeat purchase frequency**
- **Revenue from each channel**

**2. Calculate:**

- **CAC for each channel**
- **CLV**
- **ROAS per channel**
- **Overall marketing ROI**

**Customer Segmentation and Predictive Analytics**

RFM Analysis - Cluster Analysis and Persona Development - Predictive Modeling for Churn and Conversion - Recommendation Engines Overview

**Activity:**

RFM Segmentation Using Excel or Python, Perform RFM segmentation on a given transaction dataset.

**Instructions:**

**1. Use a CSV dataset containing:**

- **Customer ID, Purchase Date, Amount**

**2. Calculate:**

- **Recency: Days since last purchase**
- **Frequency: Number of purchases**

- **Monetary: Total spending**

3. **Assign RFM scores (1–5) and create customer segments using RFM quartiles.**

### **Visualization and Reporting Tools**

Dashboards with Google Data Studio / Power BI - Automated Reporting - Executive Summary Writing - Visual Best Practices

#### **Activity:**

Executive Report and Visual Audit. Write an executive summary with a supporting visual, applying best practices.

#### **Instructions:**

1. **Use insights from a prior campaign or simulated dataset (provided by instructor).**
2. **Prepare:**
  - **1-page executive summary (targeted at CMOs)**
  - **1-page dashboard screenshot or visualization**
  - **Apply formatting best practices (clear titles, color coding, and legends).**

#### **References**

1. Chuck Hemann & Ken Burbary, “Digital Marketing Analytics”, Que Publishing, 2018
2. Mike Grigsby, Marketing Analytics: A practical guide to real marketing science, Kogan Page, 2015.

#### **E- Resource:**

1. Google Analytics Certification Resources (GA4) - [\[GA4\] Google Analytics 4 account training guide and support - Analytics Help](#)

**Weightage:** Continuous Assessment: 40%, Activity: 10%, Internal Theory Examination: 30%.

#### **Assessment Methodology:**

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

<b>CO</b>	<b>Description of CO</b>	<b>PO</b>	<b>PSO</b>
<b>CO1</b>	Understand the key metrics and key performance indicators (KPIs) used in digital marketing.	PO3 (2)	PSO1 (2)
<b>CO2</b>	Analyze web and social media analytics to drive informed digital marketing decisions.	PO1 (3), PO3 (2)	PSO1 (3)
<b>CO3</b>	Use analytics platforms such as Google Analytics, Facebook Insights, and SEO tools for marketing performance evaluation.	PO1 (2), PO3 (3)	PSO1 (3)
<b>CO4</b>	Apply A/B testing, sentiment analysis, and customer segmentation techniques for marketing optimization.	PO1 (3), PO3 (2)	PSO1 (3), PSO2 (2)
<b>CO5</b>	Make data-driven decisions for paid and organic digital marketing campaigns.	PO1 (3), PO2 (2), PO3 (2)	PSO1 (3), PSO2 (2)

BD25003	Web Analytics and Development	L	T	P	C
		3	0	0	3

**Course Objective:**

1. Understand the key concepts of web analytics: traffic metrics (visitors, sessions, page views), KPIs, referrers, and how to interpret them.
2. Use tools like Google Analytics to collect and segment site data, define goals, and navigate reports (Audience, Acquisition, Behavior, Conversion).
3. Instrument websites for analytics via tagging (JavaScript, pixel tracking, log parsing), A/B testing, and heatmap/flow tools.
4. Apply analytical techniques (segmentation, cohort analysis, funnel analysis, content grouping) to extract actionable insights.
5. Develop dashboards and reports that translate web data into strategic business metrics, and optimize user experience and marketing ROI using testing results.

**Principles of Web Analytics and Instrumentation Strategies:**

Definition, history, purpose; digital KPIs and success metrics; difference between log-based and tag-based collection, JS tagging, use of tracking pixels, cookie/session logic, tag management systems

**Activities:**

1. Group discussion on privacy, consent, and ethical data collection.
2. Quiz on terminology: KPIs, sessions, cookies, pixels, tag containers.
3. Viva or presentation on instrumentation best practices and challenges.

**Google Analytics & Core Tools and Clickstream & Flow Visualization:**

Account setup, tracking code, user dimensions vs. metrics, segmentation, channels, Paths analysis, behaviour flow, exit funnels, bounce attribution

**Activities:**

1. Create custom dashboards showing traffic flow and engagement.
2. Visualize user journeys using flow reports.
3. Prepare short insight reports summarizing clickstream findings.
4. Recommend UX or content improvements based on flow analysis.

**A/B and Multivariate Testing, Content and SEO Analytics:**

Test set-up, sample sizing, interpreting significance, platform tools, Internal search tracking, SEO dashboards, crawl data, page performance.

**Activities:**

1. Evaluate content performance using engagement metrics such as page views, scroll depth, and time-on-page across user segments.
2. Perform SEO analytics by analyzing keyword performance, organic traffic, and technical SEO indicators using SEO tools.
3. Optimize digital content and campaigns by integrating insights from testing and SEO analytics and presenting actionable recommendations through reports or dashboards.

**Social and Campaign Tagging and Customer Segmentation and Sentiment Analysis:**

UTM tracking, referral analysis, integration with social dashboards, Cohort analysis, sentiment mining via APIs or plugins

**Activity:.**

1. Integrate segmentation and sentiment insights to optimize targeting, messaging, and campaign strategies and communicate findings through reports or dashboards.

**Web Application Instrumentation, Data Visualization and Dashboard Design:**

Embedding analytics events into web pages, APIs, single-page apps, Tools like Looker Studio/Power BI; storytelling with metrics Analysis to action: UX A/B, funnel drops, campaign attribution

**Activity:**

1. Present dashboards and findings through reports or presentations with data-backed conclusions.

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

**Assessment Methodology:**

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

**References:**

1. Web Analytics 2.0: The Art of Online Accountability & Science of Customer Centricity — Avinash Kaushik (Sybex, 2009).
2. Advanced Web Metrics with Google *Analytics* (2nd ed.) — Brian Clifton (Wiley, 2010).
3. Successful Analytics: Gain Business Insight by Managing Google Analytics — Brian Clifton (2015).
4. Django for Beginners: Build websites with Python and Django — William S. Vincent (2023 edition).
5. Django 4 by Example — Antonio Mele (2022).

**E-Resources:**

1. Google Analytics Academy (via [Skillshop](#))
2. Google Tag Manager Fundamentals
3. MDN Web Docs – Learn Web Development Guides
4. Flask & Django Official Documentation

CO	Description of CO	PO	PSO
CO1	Define and interpret digital analytics metrics and key performance indicators (KPIs).	PO3 (2)	PSO1 (2)
CO2	Implement tagging mechanisms and data collection strategies for digital platforms.	PO1 (3), PO3 (2)	PSO1 (3)
CO3	Analyze user behavior data and conduct testing to evaluate digital performance.	PO1 (3), PO3 (2)	PSO1 (3), PSO2 (2)
CO4	Instrument web applications and configure custom events for advanced analytics.	PO1 (3), PO3 (3)	PSO1 (3)
CO5	Build interactive dashboards and visualizations to communicate analytics insights.	PO2 (3), PO3 (2)	PSO1 (3)
CO6	Optimize digital campaigns and user experience using analytics-driven insights.	PO1 (3), PO2 (2), PO3 (2)	PSO1 (3), PSO2 (2)

<b>BD25004</b>	<b>Optimization Techniques for</b>	L	T	P	C
	<b>Analytics</b>	3	0	0	3

  

**Course Objective:**

- To understand the basics of optimization and different types of problems like linear, nonlinear, integer, and dynamic programming.
- To learn methods such as the Simplex method and Newton’s method for solving optimization problems.
- To solve real-life problems using combinatorial and graph-based optimization techniques.
- To explore metaheuristic methods like genetic algorithms and differential evolution for complex problems.
- To apply optimization in analytics and machine learning, including regression and forecasting using ARIMA models.

  

**Introduction to Optimization Problem**

Introduction - The Basic Optimization Problem - General Structure of Optimization Algorithms – Constraints-The Feasible Region. Optimization Problems: Linear Programming, Quadratic Programming, Conic Optimization, Integer Programming, and Dynamic Programming. Optimization with Data Uncertainty: Stochastic Programming - Robust Optimization.

**Activity:**

*Pre-Class:* In this flipped classroom activity, students first watch videos on optimization basics and problem types like LP, QP, and IP. They review reading materials and complete a short quiz before class. In class, they solve an LP problem using Python and explore how constraints affect solutions. They classify case studies into different optimization types and discuss uncertainty in models. After class, they complete a mini-project applying optimization with real-world uncertainty.

**Linear Programming (LP) and Nonlinear Programming (NLP)**

Linear Programming: Introduction - Duality - Optimality Conditions - The Simplex Method: Basic Solutions - Simplex Iterations - The Tableau Form of the Simplex Method - Graphical Interpretation - The Dual Simplex Method - Alternatives to the Simplex Method. Nonlinear Programming: Introduction - Univariate Optimization: Binary search - Newton’s Method - Approximate Line Search, Unconstrained

Optimization: Steepest Descent -Newton's Method, Constrained Optimization: The generalized reduced gradient method - Sequential Quadratic Programming.

**Activity :**

Students create a concept map linking LP and NLP methods, showing connections between simplex, duality, and gradient-based approaches in constrained and unconstrained problems. Each group presents how specific algorithms (e.g., Simplex, GRG, SQP) solve different classes of optimization problems with examples.

**Integer and Combinatorial Optimization**

Integer Programming: Introduction - Modelling Logical Conditions - Solving Mixed Integer Linear Programs: Linear Programming Relaxation - Branch and Bound - Cutting Planes - Branch and Cut. Combinatorial Optimization: Shortest Path – Dijkstra Algorithm, Bellman-Ford Algorithm. Greedy Algorithms - Minimum Spanning Tree.

**Activity:**

Students will work in teams to solve a real-world integer or combinatorial optimization problem (e.g., delivery routing, workforce scheduling, or facility location planning).

The project must include:

- Modelling logical conditions using Integer Programming
- Implementation of at least one exact method (e.g., Branch and Bound, Cutting Planes)
- Application of a combinatorial algorithm (e.g., Dijkstra's or Kruskal's Algorithm)
- Visualization of the optimization process or solution space (e.g., decision trees, network graphs).

**Metaheuristics Optimization**

Introduction - Single-State Methods: Hill-Climbing- Simulated Annealing- Tabu Search -Iterated Local Search. Population Methods: Genetic Algorithm- Differential Evolution - Particle Swarm Optimization.

**Activity :**

Choose a research paper that applies a metaheuristic algorithm (e.g., Genetic Algorithm or Simulated Annealing) to solve an optimization problem. Replicate the algorithm using Python or MATLAB, and apply it to a similar or small-scale real-world problem (e.g., TSP, job scheduling). Analyse the results, compare

performance with other techniques, and reflect on tuning strategies used in the original study.

### **Optimization in Machine Learning and Analytics**

Regression Analysis via Least Squares: Introduction – Linear Least Squares: Pseudo-Inverse – Brief Discussion of Probabilistic Interpretation, Regularized Linear Least Squares. Forecasting: Stationary Data and Differencing: Autocorrelation, ARIMA Models: Autoregression Models – Moving Average Models – ARIMA Model Structure.

#### **Activity:**

Create a digital model that demonstrates the process of regression using least squares and regularization, visualizing error minimization and the effect of overfitting. Include time series forecasting components with AR, MA, and ARIMA models, using real or simulated data to showcase trends, autocorrelation, and predictions.

### **References**

1. Practical Optimization: Algorithms and Engineering Applications, Andreas Antoniou , Wu-Sheng Lu, Springer, 3rd Edition 2021.
2. Optimization Methods in Finance, Gérard Cornuéjols, Javier Peña, Reha Tütüncü, Cambridge University Press, 2nd Edition, 2018.
3. Introduction to Combinatorial Optimization, Ding-Zhu Du, Panos M. Pardalos, Xiaodong Hu, Weili Wu, Springer, 2022.
4. Essentials of Metaheuristics A Set of Undergraduate Lecture Notes, Sean Luke Department of Computer Science George Mason University, Second Edition, Online Version 2.3, February, 2016.
5. An Introduction to Optimization With Applications in Machine Learning and Data Analytics, Jeffrey Paul Wheeler, CRC Press, Taylor & Francis Group, LLC, 2024.

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

#### **Assessment Methodology:**

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

<b>CO</b>	<b>Description of CO</b>	<b>PO</b>	<b>PSO</b>
<b>CO1</b>	Model and formulate various optimization problems considering constraints and data uncertainty using linear, integer, nonlinear, and stochastic formulations.	PO1 (3), PO3 (2)	PSO1 (3)
<b>CO2</b>	Apply solution methods such as linear, nonlinear, integer, and dynamic programming to solve optimization problems effectively.	PO1 (3), PO3 (3)	PSO1 (3)
<b>CO3</b>	Solve integer and combinatorial optimization problems using techniques such as branch and bound, cutting planes, and graph algorithms.	PO1 (3), PO3 (3)	PSO1 (3), PSO2 (2)
<b>CO4</b>	Utilize metaheuristic techniques including genetic algorithms, differential evolution, and particle swarm optimization for complex search problems.	PO1 (3), PO3 (3)	PSO1 (3), PSO2 (2)
<b>CO5</b>	Incorporate optimization techniques for regression and forecasting tasks by selecting appropriate methods for real-world analytics applications.	PO1 (3), PO2 (2), PO3 (2)	PSO1 (3), PSO2 (2)

<b>BD25005</b>	<b>Data Lake Architecture and Applications</b>	L	T	P	C
		3	0	0	3
<b>Course Objective:</b>					
<p>The objective of this course is to provide learners with a comprehensive understanding of Data Lake concepts, architecture, design principles, and technologies. The course aims to enable participants to implement and manage scalable and cost-effective data lake solutions that support modern data processing needs, including big data analytics, machine learning, and real-time data processing.</p>					
<b>Introduction to Data Lakes and Lakehouse architecture</b>					
<p>Describe the value of data lakes -Compare data lakes and data warehouses. Data lake characteristics and Core components of a data lake - Recognize common architectures built on data lakes. Traditional architecture and modern data platforms- Real-world applications and use cases</p> <p><b>Activity:</b> Hands-on exploration on cloud-based data lakes.</p>					
<b>Data Ingestion and Storage</b>					
<p>Layers in Data Lake Architecture: Ingestion Layer - Storage Layer -Processing Layer - Metadata Management Layer - Consumption Layer. Data Lake Zones: Raw Zone - Cleansed Zone- Curated Zone-Sandbox/Exploration Zone - Data Governance and Cataloging - Security and Access Control.</p> <p><b>Activity:</b> Reinforce learning through a fun quiz.</p>					

### **Building and Managing a Data Lake**

Data Ingestion Methods (Batch, Stream) - ETL vs ELT in Data Lake- Tools for Data Ingestion (Apache NiFi, Kafka, AWS Glue, Azure Data Factory) - Data Transformation and Processing (Apache Spark, AWS EMR) - Managing Metadata (Apache Atlas, AWS Glue Data Catalog)- Data Lake Optimization (partitioning, compression, indexing)

#### **Activities:**

1. Implement data ingestion pipelines by comparing batch and stream ingestion methods and configuring ETL/ELT workflows using tools such as Apache NiFi, Kafka, AWS Glue, or Azure Data Factory.
2. Manage and optimize a data lake environment by performing data transformation using Apache Spark/AWS EMR, maintaining metadata with catalog tools, and applying optimization techniques such as partitioning, compression, and indexing.

### **Advanced Concepts and Real-World Applications**

Data Lake house Concept - Integration with BI and ML Tools (Power BI, Tableau, Sage Maker) - Monitoring and Logging (Cloud Watch, Azure Monitor) - Performance Tuning and Cost Optimization - Case Studies: Data Lake Implementation in Enterprises - Future Trends: Serverless Lakes, Data Mesh, AI on Data Lakes

#### **Activities:**

1. Analyze enterprise data lake and lakehouse architectures by studying integration with BI and ML tools, monitoring and logging mechanisms, and performance tuning and cost optimization strategies.
2. Evaluate real-world case studies and emerging trends by examining enterprise data lake implementations, serverless data lakes, data mesh architectures, and AI-driven analytics on data lakes.

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

#### **Assessment Methodology :**

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

## References

1. Gaurav Ashok Thalpati, "Practical Lakehouse Architecture", O'Reilly Media, 2024, ISBN: 9781098153014
2. Denny Lee, Tristen Wentling, Scott Haines, Prashanth Babu "Delta Lake: The Definitive Guide", O'Reilly Media, 2024, ISBN: 9781098151942
3. Bill Inmon Data Lake Architecture: Designing the Data Lake and Avoiding the Garbage Dump, Technics Publications LLC, 2016
4. Rukmani Goplan, "The Cloud Data Lake: A Guide to Building Robust Cloud Data Architecture", O'Reilly Media, 2022
5. Alex Gorelik, "The Enterprise Big Data Lake: Delivering the Promise of Big Data and Data Science", O'Reilly Media, 2019.

## E-Resources

1. <https://aws.amazon.com/big-data/datalakes-and-analytics/>
2. <https://learn.microsoft.com/en-us/azure/data-lake/>

CO	Description of CO	PO	PSO
CO1	Describe the architecture, conceptual understanding, and key components of modern data lake systems.	--	--
CO2	Analyze data lake designs including zones, ingestion flows, and pipeline processes to understand data management and processing requirements.	PO1 (3)	PSO1 (3)
CO3	Evaluate pipeline implementations, governance, metadata management, and security practices to ensure reliability, compliance, and efficiency.	PO3 (2)	PSO2 (2)
CO4	Design optimized data lake solutions integrating cost/performance considerations, analytics frameworks, and end-to-end workflow strategies.	PO2 (1)	PSO1 (3)

BD25006	Data Modeling and Integration	L	T	P	C
		3	0	0	3
<p><b>Course Objectives:</b></p> <p>The objective of the course is to equip postgraduate students with</p> <ul style="list-style-type: none"> <li>• A comprehensive understanding of how data is structured, modelled, and integrated across diverse systems.</li> <li>• The course aims to develop the ability to design and implement conceptual, logical, and physical data models that support efficient data storage and retrieval.</li> <li>• It covers techniques and tools for integrating data from multiple sources using ETL, big data, and cloud platforms.</li> <li>• The course also emphasizes data quality, governance, and security to ensure reliable and compliant data integration processes.</li> </ul>					
<p><b>Introduction to Data Modelling:</b> Definition and Importance - Types of Data Models- Entity-Relationship (ER) Modelling Basics-Enhanced ER (EER) Modelling- Relational Data Model Overview-Functional Dependencies and Anomalies- Normalization Role of Data Modelling in Big Data Analytics.</p> <p><b>Activity:</b> Design an ER model for a university or hospital database.</p>					
<p><b>Advanced Data Modelling Techniques:</b> Dimensional Data Modelling-Slowly Changing Dimensions (SCD)-Data Vault Modelling-Temporal Data Modelling- Spatial Data Modelling-Semantic Data Modelling-Graph-Based Data Document-Oriented Model (NoSQL)-Key-Value Data Model (Riak)-Column-Family Data Model (HBase)-Types of Consistency</p> <p><b>Activity:</b> Build a document-oriented model using MongoDB (e.g., product catalog or user profile).</p>					
<p><b>Data Integration Concepts:</b> Introduction to Data Integration-Types of Data Sources-ETL (Extract, Transform, Load) Process-ELT (Extract, Load, Transform) Approach-Data Virtualization and Federation-Master Data Management (MDM)- MapReduce-pig-DAG-Cascading-Spark-Kafka-CouchDB- NEO4J- Redis</p> <p><b>Activity:</b> Simulate an ETL pipeline using Talend Open Studio or Python (Pandas). Integrate data from two different sources (CSV and MySQL) into a staging table.</p>					
<p><b>Big Data Integration Tools and Platforms:</b> Introduction-Hadoop Ecosystem for Data Integration-Apache Spark for Unified Data Integration-Real-time Data Ingestion and Integration Tools-ETL and Data Pipeline Tools-Cloud-based</p>					

Integration Platforms-Data Lake and Warehouse Integration-Metadata and Data Catalog Tools-Blockchain-Quantum Computing

**Activity:** Process a dataset using Apache Spark (e.g., perform aggregations or joins on large CSV files).

**Data Quality, Governance, and Security:** Introduction to Data Quality-Data Cleaning and Standardization-Data Governance Frameworks-Master and Reference Data Management-Big Data Security-Data Security Fundamentals-Regulatory Compliance and Privacy-Auditing, Monitoring, and Data Lineage

**Activity:** Apply encryption and access control on a database (e.g., MySQL user privileges, MongoDB role-based access).

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

**Assessment Methodology :**

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

**References:**

1. Designing Data-Intensive Applications by Martin Kleppmann, O'Reilly Media, 2017
2. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling by Ralph Kimball & Margy Ross, Wiley, 2025
3. Data Governance by John Ladley, Elsevier/Morgan Kaufmann, 2019
4. Database System Concepts by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill Education, 2013

**eBooks**

1. Data Modeling Essentials (3rd Edition) by Graeme Simsion & Graham Witt
2. Data Integration Blueprint and Modelling by Anthony David Giordano
3. The Ultimate Guide to Data Integration by Fivetran

**E- resources**

1. An industry-focused guide on modern data integration practices and tools. NPTEL & MOOC Courses
2. Database Management System by IIT Kharagpur

<b>CO</b>	<b>CO Description</b>	<b>PO</b>	<b>PSO</b>
<b>CO1</b>	Design and implement conceptual, logical, and physical data models for structured and semi-structured data sources.	PO1, PO3	PSO1
<b>CO2</b>	Apply advanced data modeling techniques such as dimensional modeling and NoSQL schema design to real-world scenarios.	PO1, PO3	PSO1, PSO2
<b>CO3</b>	Develop and manage data integration workflows using ETL/ELT tools and platforms across heterogeneous data sources.	PO1, PO3	PSO1, PSO2
<b>CO4</b>	Use big data integration frameworks such as Hadoop, Spark, Kafka, and cloud services for handling large-scale data processing.	PO1, PO3	PSO1
<b>CO5</b>	Assess and improve data quality, implement governance policies, and apply security and privacy measures in data integration processes.	PO1, PO3	PSO2

BD25007	<b>Data Acquisition and Production Management Practices</b>	L	T	P	C
		3	0	0	3

**Course Objectives:**

- Understand the principles and architecture of data acquisition systems in industrial environments.
- Apply knowledge of sensors, data loggers, and communication protocols in production contexts.
- Integrate real-time data acquisition with production planning and management.
- Develop analytical and problem-solving skills for optimizing production performance using data.

**Data Acquisition Fundamentals:** Introduction to Data Acquisition Systems - Signal conditioning and interfacing - Types of sensors and transducers (temperature, pressure, flow, etc.) - Analog-to-Digital Conversion (ADC) and Digital-to-Analog Conversion (DAC) - Data loggers and SCADA systems.

**Activity:**

**Scenario:** Calibrate a temperature sensor (e.g., LM35 or thermistor) in a smart manufacturing setup.

**Tasks:**

- Design a calibration routine.
- Apply filtering (moving average/low-pass filter).
- Scale sensor output to °C.

**Data Communication and Networking:** Serial and parallel communication protocols (RS-232, RS-485) - Protocols: Modbus, CAN, Ethernet, OPC UA - Wired and Wireless Connectivity: ZigBee, Wi-Fi - IoT Platforms: ThingSpeak, Blynk, AWS IoT - Cloud-based Acquisition and Real-time Data Transmission & Security.

**Activity:**

**Scenario:** Build a real-time data logging system for a smart agriculture environment.

**Tasks:**

- Use Arduino/Raspberry Pi with DHT11, soil moisture sensors.
- Log data to CSV or cloud.
- Display values on LCD or monitor.
- Optionally upload data to ThingSpeak/Google Sheets.

**Production Planning and Control:** Overview of Production Systems - Demand forecasting and capacity planning - Inventory management: EOQ, JIT - Scheduling: Gantt Charts, Johnson's Rule - Lean manufacturing, Six Sigma

**Activity:**

**Scenario:** Design a real-time dashboard for a smart manufacturing unit.

**Tasks:**

- Collect machine data (e.g., temp, speed, count).
- Use SCADA (Factory I/O, Ignition) or IoT (Node-RED, Grafana).

- Implement threshold-based alerts and documentation.

### **Integration of Data Acquisition with Production Systems**

Real-time monitoring and control - Data for quality assurance - Preventive and predictive maintenance - MES (Manufacturing Execution Systems) and ERP System interfaces.

#### **Activity:**

**Scenario:** Perform time-series forecasting of monthly production.

#### **Tasks:**

- Clean and visualize historical data.
- Apply forecasting (Moving Average, ARIMA, Holt-Winters).
- Evaluate with RMSE, MAPE.
- Predict future outputs and generate insights.

### **Analytics and Case Studies in Production Management**

Statistical Process Control (SPC) - Analytics with Python, R, or MATLAB - Industry 4.0 and Smart Factories - Case Studies: Manufacturing, Automotive, Electronics

#### **Mini-Project:**

**Scenario:** Prototype a smart production line.

#### **Tasks:**

- Design and simulate a conveyor/packaging line.
- Integrate sensors (proximity, IR, load cell).
- Implement acquisition and logging with Arduino/RPi.
- Build dashboards (Blynk, Grafana).
- Enable alert-based control (e.g., temperature limits).
- Present system with demo, documentation, and insights.

### **References**

1. Measurement Systems: Application and Design, Ernest O. Doebelin, McGraw-Hill, 5<sup>th</sup> edition, 2002
2. Advanced Information-Measuring Technologies and Systems, Volodymyr Eremenko & Artur Zaporozhets, Springer 2023
3. Operations Management: Theory and Practice, B. Mahadevan Pearson Paperback, Third Edition, 2015
4. Automation, Production Systems, and Computer-Integrated Manufacturing, Mikell P. Groover, 4<sup>th</sup> edition, Pearson, 2016
5. Internet Of Things (IOT): Architecture and Design Principles, Raj Kamal, 2nd Edition, McGraw-Hill, 2022

### **E-Resources**

1. NPTEL – Industrial IoT - [https://onlinecourses.nptel.ac.in/noc20\\_cs69/preview](https://onlinecourses.nptel.ac.in/noc20_cs69/preview)
2. Coursera – Smart Manufacturing - <https://www.coursera.org/specializations/digital-technology-in-manufacturing>

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

**Assessment Methodology :**

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

<b>CO</b>	<b>CO Description</b>	<b>PO</b>	<b>PSO</b>
<b>CO1</b>	Design and implement data acquisition systems for industrial applications.	PO1, PO3	PSO1
<b>CO2</b>	Understand and apply production management tools and methods.	PO1, PO3	PSO1
<b>CO3</b>	Collect and analyze production data using digital tools.	PO1, PO3	PSO1, PSO2
<b>CO4</b>	Develop IoT and cloud-based solutions for smart manufacturing.	PO1, PO3	PSO1
<b>CO5</b>	Solve real-time production challenges using data-driven approaches.	PO1, PO3	PSO1, PSO2

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)

CP25C17	Edge and Fog Computing	L	T	P	C
		3	0	0	3
<p><b>Course Objective:</b></p> <p>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.</p>					
<p><b>Cloud to Edge Computing Paradigms:</b> 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</p>					
<p><b>Suggested Activities:</b></p> <ul style="list-style-type: none"> <li>• Case Study: IoT solution using fog or edge</li> <li>• Group Discussion: Business opportunities in edge markets</li> </ul>					
<p><b>Edge Offloading and System Optimization:</b> 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</p>					

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

BD25008	Data Intensive Computing	L	T	P	C
		3	0	0	3
<p><b>Course Objective:</b></p> <ol style="list-style-type: none"> <li>1. Understand data-intensive computing and be able to analyze, design, and implement effective solutions for data-intensive applications with very large-scale data sets.</li> <li>2. Able to design and develop solutions using data-intensive computing models.</li> <li>3. Understand the approaches that address challenges in managing and utilizing ultra-scale data and the methods for transforming big data into discoveries and intelligence for human understanding and decision making.</li> </ol>					

**Basics and Platform of Data Intensive Computing:** Data Intensive computing Paradigms-types, need and use – Supercomputing, Grid Computing, Cloud Computing, Many-core Computing. Parallel Programming Systems-MapReduce-Hadoop, Workflows-Swift, MPI-MPICH, OpenMP, Multi-Threading-PThreads.

**Activities:**

1. Compare data-intensive computing paradigms by analyzing Supercomputing, Grid, Cloud, and Many-core computing in terms of architecture, scalability, and application suitability.
2. Demonstrate parallel programming systems by implementing a simple data processing task using any one platform such as Hadoop MapReduce, MPI (MPICH), OpenMP, PThreads, or Swift workflows.

**Scheduling and File Systems:** Job Management Systems- Batch scheduling, Lightweight Task Scheduling. Storage Systems-File Systems- EXT3, Shared File Systems -NFS, Distributed File Systems-HDFS, FusionFS, Parallel File Systems-GPFS, PVFS, Lustre, Distributed NoSQL Key/Value Stores-Cassandra, MongoDB, ZHT, Relational Databases-MySQL.

**Activities:**

1. **Analyze job scheduling and task management systems** by comparing batch scheduling and lightweight task scheduling approaches used in data-intensive computing environments.
2. **Explore storage and file system technologies** by studying the architecture and use cases of local (EXT3), shared (NFS), distributed (HDFS, FusionFS), parallel file systems (GPFS, PVFS, Lustre), NoSQL key–value stores (Cassandra, MongoDB, ZHT), and relational databases (MySQL).

**Derived and Distributed Data Systems:** Data Models and Query Languages - Storage and Retrieval - Encoding and Evolution – Replication - Partitioning - The Trouble with Distributed Systems - Consistency and Consensus - Batch Processing - Stream Processing. Data-Intensive Computing with GPUs and databases, many-core computing era, and new challenges, Case studies on open research questions in data-intensive computing.

1. **Analyze distributed data system design principles** by examining data models, query languages, storage and retrieval mechanisms, encoding and evolution, replication, partitioning, and consistency/consensus challenges in real-world systems.
2. **Evaluate modern data-intensive computing approaches** by studying batch vs. stream processing frameworks, GPU-accelerated databases, many-core computing architectures, and selected case studies highlighting open research challenges.

**Course Outcomes:**

1. Design & implement data-acquisition systems for industrial applications.
2. Understand & apply production management tools and methods
3. Collect & analyse production data using digital tools
4. Develop IoT & cloud-based solutions for smart manufacturing
5. Solve real-time production challenges using data-driven approach

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

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

**References:**

1. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems - Martin Kleppmann 1<sup>st</sup> Edition, O'Reilly Media. 2017.
2. Doing Data Science: Straight Talk from the Frontline, 1st Edition Author(s): Cathy O'Neil and Rachel Schutt ISBN: 978 1449358655 Publisher: O'Reilly Media.

3. Data intensive computing: architectures, algorithms and applications - Tevfik Kosar, Ian Gorton & Deborah Grácio 1st Edition, ISBN: 1615209719 / 978-1615209712 Information Science Reference (IGI Global) 2012.
4. The Fourth Paradigm: Data-Intensive Scientific Discovery - Tony Hey, Stewart Tansley & Kristin Tolle, 1st Edition ISBN-10: 0982544200 / ISBN-13: 978-0982544204 Microsoft Research 2009.
5. Joel Grus. Data Science from Scratch: First Principles with Python 1st Edition, 978 1491901427, O'Reilly Media. 2015.

**E-Resource**

1. <https://dataintensive.net>
2. <https://www.mcs.anl.gov/~itf/dbpp/>
3. <https://www.elsevier.com>

CO	CO Description	PO	PSO
CO1	Design and implement data acquisition systems for industrial applications.	PO1, PO3	PSO1
CO2	Understand and apply production management tools and methods.	PO1, PO3	PSO1
CO3	Collect and analyse production data using digital tools.	PO1, PO3	PSO1, PSO2
CO4	Develop IoT and cloud-based solutions for smart manufacturing.	PO1, PO3	PSO1
CO5	Solve real-time production challenges using data-driven approaches.	PO1, PO3	PSO1, PSO2

<b>BD25009</b>	<b>Deep Learning Ecosystem</b>	L	T	P	C
		3	0	0	3
<p><b>Course Objective:</b>  This course provides an in-depth understanding of the deep learning ecosystem, including the theoretical foundations, practical implementations, tools, frameworks, and applications in various domains.</p>					
<p><b>Introduction to Deep Learning:</b> Overview of machine learning vs. deep learning - Historical context and evolution of deep learning - Applications of deep learning in industry and research - Neural Networks: structure and function - Activation functions and loss functions - Forward and backward propagation - Gradient descent and optimization techniques</p> <p><b>Activity:</b>  <b>Quizzes</b> - To quickly assess understanding of core concepts, terminology, and foundational knowledge in deep learning</p>					
<p><b>Deep Learning Frameworks:</b> Introduction to Tensor Flow and PyTorch - Setting up the environment: Anaconda, Jupyter Notebooks - Building the first neural network.</p> <p><b>Activity:</b>  <b>Project-based learning</b> - To assess students' practical skills in implementing deep learning models and solving real-world problems.</p>					
<p><b>Convolutional Neural Networks and Recurrent Neural Networks:</b>  Understanding convolution, pooling layers, and architectures - Applications to image classification and object detection - Transfer learning and pre-trained models - Introduction to recurrent architectures - Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs) - Applications in natural language processing (NLP) and time series analysis</p> <p><b>Activity:</b>  <b>Concept Maps</b> - Students create concept maps or diagrams illustrating the flow of information through a neural network or the relationships between different deep learning algorithms.</p>					
<p><b>Advanced Deep Learning Techniques:</b> Generative Adversarial Networks (GANs) – Auto encoders and Variational Auto encoders (VAEs) - Reinforcement Learning overview and its relation to deep learning</p>					

<p><b>Activity:</b></p> <p><b>Reproduction of Research Paper</b> - Students keep a journal to document their learning experiences, identify challenges, and reflect on their understanding of the material.</p>
<p><b>Application Domains:</b> Computer Vision: Image segmentation, style transfer, and facial recognition - Natural Language Processing: Text generation, sentiment analysis, and machine translation - Speech Recognition and Generation</p> <p><b>Activity:</b></p> <p><b>Case Studies</b> - Present students with real-world case studies of deep learning applications.</p>
<p><b>References</b></p> <ol style="list-style-type: none"> <li>1. Deep Learning by Ian Goodfellow, Yoshua Benjio, Aaron Courville, The MIT Press, 2016.</li> <li>2. Deep Learning: A Practitioner's Approach by Josh Patterson and Adam Gibson, O'Reilly Media, Inc., 2017</li> <li>3. Learn Keras for Deep Neural Networks by Jojo Moolayil, Apress, 2018</li> <li>4. Deep Learning Projects Using TensorFlow 2 by Vinita Silaparasetty, Apress, 2020</li> <li>5. Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms by Nikhil Buduma, O'Reilly Publications, 2017.</li> </ol>
<p><b>E-Resource:</b></p> <ul style="list-style-type: none"> <li>• <a href="https://onlinecourses.nptel.ac.in/noc20_cs22">https://onlinecourses.nptel.ac.in/noc20_cs22</a></li> <li>• <a href="https://onlinecourses.nptel.ac.in/noc20_cs62">https://onlinecourses.nptel.ac.in/noc20_cs62</a></li> </ul>
<p><b>Weightage:</b> Continuous Assessment: 40%, End Semester Theory Examination: 60%</p>
<p><b>Assessment Methodology:</b></p> <p>Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE &amp; IES questions (25).</p>

CO	CO Description	PO	PSO
CO1	Understand the core concepts and components of deep learning.	PO3	PSO1
CO2	Implement deep learning algorithms using popular	PO1,	PSO1

	frameworks.	PO3	
<b>CO3</b>	Evaluate and optimize deep learning models.	PO1, PO3	PSO1, PSO2
<b>CO4</b>	Apply advanced deep learning techniques to real-world problems across various domains.	PO1, PO3	PSO1, PSO2
<b>CO5</b>	Stay updated with various application domains and emerging trends in deep learning.	PO2, PO3	PSO2

CP25C19	Cognitive Computing	L	T	P	C
		3	0	0	3
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>To understand the core principles of cognitive computing and how they emulate human thought processes.</li> <li>To apply cognitive computing to solve complex, unstructured, and real-world problems.</li> <li>Develop cognitive computing applications.</li> </ul>					
<p>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.</p> <p><b>Activities:</b></p> <ol style="list-style-type: none"> <li>Concept Quizzes: Human cognition and Cognitive Computing</li> <li>Research paper review on cognitive systems</li> </ol>					
<p>Cognitive architectures: ACT-R, Soar, OpenCog - Knowledge Representation and Reasoning - Machine Learning models in cognitive systems - Human-like Learning: Supervised, Unsupervised, Reinforcement</p> <p><b>Activities:</b> Demo on IBM Watson APIs (Language Translator)</p>					
<p>Role of NLP in a Cognitive System - NLP pipeline: POS tagging, NER, parsing - Sentiment Analysis and Question Answering</p> <p>Debate: Sentiment analysis and its challenges in Regional Languages.</p> <p>Semantic Web - Semantic Understanding: Ontologies and Knowledge Graphs</p> <p><b>Activities:</b></p> <p>Ideathon: Innovative solution to health care problems.</p>					
<p>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</p> <p><b>Activities:</b> Mini project: Develop health care chatbot. Prototype Demonstrations.</p>					
<p>Process of Building a Cognitive Application - IBM's Watson as a Cognitive System- Business Implications of Cognitive Computing</p> <p><b>Activities:</b> Research Paper Review: Cognitive Computing</p>					

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>”

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

**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>	

**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**

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, technologies, and components of mixed reality and their applications in immersive digital environments.	--	--
CO2	Analyze mixed reality systems and interaction techniques to understand performance, usability, and user experience in immersive applications.	PO1 (3)	PSO1 (3)
CO3	Evaluate mixed reality platforms, devices, and development frameworks to assess their suitability for various application domains.	PO3 (2)	PSO2 (2)
CO4	Design mixed reality applications by selecting appropriate hardware, software frameworks, and interaction models for real-world use cases.	PO2 (1)	PSO1 (3)

BD25010	Data Manipulation and Visualization	L	T	P	C
		3	0	0	3
<p><b>Course Objective:</b></p> <p>The course aims to</p> <ul style="list-style-type: none"> <li>• Develop advanced competencies in data manipulation and visualization, focusing on the efficient handling, transformation, and integration of diverse datasets.</li> <li>• Students will learn to create clear, accurate, and interactive visual representations using tools like Python, R, and Tableau to support data analysis and decision-making.</li> <li>• Emphasis is placed on ethical data communication, reproducible workflows, and applying visualization techniques to complex, real-world problems.</li> </ul>					
<p><b>Foundation for Data Manipulation</b></p> <p>Need for data manipulation- Series: Create series object from a list and dictionary - The head and Tail methods - Passing Series to Python Built-In Functions – Methods for Data sorting; Data frame: Methods and Attributes between Series and Data Frames - Fill in Missing Values - Filtering data and methods in Data frame - Data Extraction in data frames -Working with Text Data - Merging Data frames; Data Mining - Data Processing and Modelling.</p> <p><b>Activities</b></p> <ul style="list-style-type: none"> <li>• <i>Before Class:</i> Students watch a tutorial video on creating and manipulating Series and DataFrames using pandas in Python.</li> <li>• <i>In Class:</i> Students complete hands-on coding exercises where they load a dataset, clean missing values, filter data, and perform sorting and merging operations with instructor support</li> </ul> <p><b>Data Wrangling</b></p> <p>Data Sources - Data Loading, Storage and File Formats - Reading and Writing Data in Text Format, Web Scraping, Binary Data Formats, interacting with Web APIs, Interacting with Databases - Data Wrangling - Hierarchical Indexing, Combining and Merging Data Sets Reshaping and Pivoting Tools for Data Wrangling Data Cleaning and Preparation Handling Missing Data, Data Transformation, String Manipulation</p> <p><b>Activity:</b></p> <p>Hierarchical Indexing and Data Reshaping Techniques: Poster illustrates how multi-level indexing and pivoting/unpivoting data structures help organize and analyze</p>					

complex datasets

### **Introduction to Data Visualization and Visual Analytics**

Overview of data visualization - Data Abstraction - Task Abstraction - Analysis: Four Levels for Validation - Scalar and Point techniques – Color maps – Contouring – Height Plots - Vector visualization techniques –Vector Glyphs – Vector Color Coding – Matrix visualization techniques. Visual Analytics: Visual Variables- Networks and Trees - Map Color and Other Channels- Manipulate View- Heat Map. Streaming Data Visualization: Presenting streaming data, streaming visualization techniques, streaming analysis. Geo Spatial Visualization: Chloropleth map, Hexagonal Binning, Dot map, Cluster map, cartogram map

### **Project-based Learning Activity [Example]:**

Students will work in teams to select a complex dataset (e.g., climate data, social media analytics, public health statistics, or transportation data) and develop a set of visualizations that illustrate key patterns, trends, and relationships. The project must include at least:

- One scalar or point visualization (e.g., contour plot or height map)
- One vector/matrix technique (e.g., glyphs, heatmaps, matrix display)
- One geospatial visualization (e.g., chloropleth, dot map, or cartogram)
- One streaming data or simulated streaming scenario, if applicable

### **Time Series and Multivariate Data Visualization**

Time Series Visualization - Individual time series - Multiple time series and dose–response curves - Two or more response variables- Visualizing trends – Smoothing. Multivariate data visualization – Geometric projection techniques - Icon-based techniques - Pixel-oriented techniques - Hierarchical techniques - Scatterplot matrix - Hyperbox – Trellis display - Parallel coordinates

### **Reproduction of Research Paper Activity:**

Choose a study presenting multivariate visualizations of patient health metrics (e.g., blood pressure, cholesterol, glucose levels) over time. Replicate icon-based or pixel-oriented techniques shown in the paper and analyze the trends or clusters identified.

### **DATA Visualization Tools, Best Practices in Data Visualization**

Tools: Tableau functions and logics: Marks and Channels-Arrange Tables-Arrange

Spatial Data- Facets into multiple views. Best Practices: Common pitfalls of color use - Redundant coding - Multi-panel figures - Titles, captions, and tables - Balance the data and the context - Use larger axis labels - Choosing the effective visualization - Issues of Cognition, Perception, and Reasoning. Issues of System Design Evaluation, Hardware, and Applications

**Model Making Activity:**

Develop a physical or digital display that contrasts ethical vs. misleading visualizations, showing examples of distortion, cherry-picking, poor labeling, and exaggerated axes

**References**

1. Data Analytics using Python, Bharti Motwani, Wiley, 2nd edition, 2020
2. Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython, Wes McKinney, O'Reilly, 2nd Edition, 2017
3. Data Visualization: A Practical Introduction, Kieran Healy, Princeton University Press, 2nd Edition, 2023
4. Information Visualization Perception for Design, Colin Ware, Morgan Kaufmann Publishers, 4th edition, 2021
5. Interactive Visual Data Analysis, Christian Toninski, Heidrun Schumann, CRC Press publication, 2020

**E-Resources**

1. [https://onlinecourses.nptel.ac.in/noc25\\_cs131](https://onlinecourses.nptel.ac.in/noc25_cs131)
2. [https://onlinecourses.nptel.ac.in/noc25\\_cs104](https://onlinecourses.nptel.ac.in/noc25_cs104)

**Course Outcomes:**

1. Demonstrate proficiency in applying data manipulation techniques for the structured transformation, cleaning, and preparation of datasets for analysis.
2. Implement advanced data wrangling techniques to clean, restructure, and integrate data from multiple sources for analysis and visualization.
3. Design and execute a diverse array of data visualizations and visual analytics methodologies to effectively communicate complex data insights
4. Perform comprehensive analysis and visualization of time series and multivariate data to uncover underlying patterns, trends, and interrelationships.
5. Effectively use contemporary data visualization tools and apply best practices in design, color theory, and user cognition to create clear, accurate, and

ethically sound visual representations.		
<b>Weightage:</b>	Continuous Assessment: 40%	End Semester Theory Examination: 60%
	(i). Activities: 15% (ii). Internal Theory Examinations: 25%	
<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		

<b>CO</b>	<b>CO Description</b>	<b>Mapped POs</b>	<b>Mapped PSOs</b>
<b>CO1</b>	Apply data manipulation techniques for structured transformation, cleaning, and preparation of datasets for analysis.	PO1, PO3	PSO1
<b>CO2</b>	Implement advanced data wrangling techniques to clean, restructure, and integrate data from multiple sources for analysis and visualization.	PO1, PO3	PSO1, PSO2
<b>CO3</b>	Design and execute diverse data visualizations and visual analytics methods to effectively communicate complex data insights.	PO2, PO3	PSO1
<b>CO4</b>	Perform comprehensive analysis and visualization of time series and multivariate data to identify patterns, trends, and interrelationships.	PO1, PO3	PSO1, PSO2
<b>CO5</b>	Use contemporary data visualization tools and apply best practices in design, color theory, user cognition, and ethics to create clear and accurate visual representations.	PO2, PO3	PSO2

BD25C02	Exploratory Data Analysis	L	T	P	C
		3	0	0	3
<p><b>COURSE OBJECTIVES:</b> 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.</p>					
<p><b>INTRODUCTION</b> 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.</p> <p><b>ACTIVITIES:</b></p> <ol style="list-style-type: none"> <li>1. Conduct a Think Pair Share activity on dataset exploration and question formulation.</li> <li>2. Conduct a Jigsaw activity on Data type and quality issue identification.</li> </ol>					
<p><b>DATA WRANGLING</b> 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.</p> <p><b>ACTIVITIES:</b></p> <ol style="list-style-type: none"> <li>1. Conduct a Peer Debugging Session on handling missing values, duplicates, and type inconsistencies.</li> <li>2. Assign a Mini Project on real-world dataset wrangling, focusing on merging and reshaping data.</li> <li>3. Conceptual discussion on outlier identification.</li> </ol>					
<p><b>MULTIVARIATE ANALYSIS, RELATIONSHIP EXPLORATION AND CAUSAL INFERENCE</b> 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</p> <p><b>ACTIVITIES:</b></p> <ol style="list-style-type: none"> <li>1. Conduct a <i>Group Analysis Task</i> to explore correlation matrices and identify multicollinearity.</li> <li>2. Use a <i>Role Play Simulation</i> to explain Simpson's Paradox and how it affects decision-making.</li> <li>3. Conduct a Visual Inference Exercise where students use pair plots and heatmaps to explore relationships and patterns within a dataset.</li> </ol>					

**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%	

**Mandated Activities with marks:**

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

**Internal Examinations:** TWO tests

**REFERENCES:**

1. Mukhiya, S. K., & Ahmed, U. (2020). Hands-On Exploratory Data Analysis with Python. Packt Publishing.
2. Wes McKinney (2022), *Python for Data Analysis*, O'Reilly Media (3rd Edition)
3. Grus, J. (2019). Data science from scratch: First principles with Python. O'Reilly Media.

**E-RESOURCES:**

1. Deep Reinforcement Learning – Prof. Shivaram Kalyanakrishnan, IIT Madras – <https://nptel.ac.in/courses/106106179>
2. <https://www.youtube.com/user/joshstarmar>
3. <https://www.kaggle.com/learn>
4. <https://www.coursera.org/specializations/jhu-data-science>
5. <https://www.analyticsvidhya.com/blog/2021/06/exploratory-data-analysis-eda-step-by-step/>

**Course Outcomes:**

**CO1:** Apply Exploratory Data Analysis (EDA) techniques using Python libraries on real-world datasets.

**CO2:** Analyze data using statistical measures and derive meaningful patterns through appropriate visualizations.

**CO3:** Design and implement data wrangling workflows and perform time-series transformations.

**CO4:** Evaluate statistical hypotheses and build predictive models using regression techniques.

**CO5:** Build and deploy basic data science models and interpret the results to derive business insights.

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

<b>BD25011</b>	<b>Geospatial Analytics</b>	L	T	P	C
		3	0	0	3

### **Course Objectives**

By the end of this course, students will be able to:

1. Understand the fundamental concepts of geospatial data, coordinate systems, and spatial relationships.
2. Explore data acquisition techniques including GPS, remote sensing, and open geospatial data platforms.
3. Apply geospatial data processing and analytics using GIS tools and Python libraries. Develop the ability to analyze spatial patterns and present insights using maps, dashboards, and visualizations.
4. Evaluate real-world case studies involving geospatial solutions in domains like urban planning, agriculture, disaster management, and public health.

### **Foundations of Geospatial Analytics**

Introduction to Geospatial Analytics: Definition, scope, and relevance – Types of geospatial data: Vector and Raster – Coordinate systems and map projections – Spatial relationships: topology, distance, buffering, overlays – Components of GIS and their functions – Types of GIS: Desktop, Web, and Mobile – Spatial resolution, scale, and accuracy – Basics of Map Composition and Cartography Principles.

### **Geospatial Data Acquisition and Management**

Remote sensing fundamentals – GPS and GNSS: Working principles and applications – Open data portals: OpenStreetMap, USGS Earth Explorer, Copernicus – Spatial data formats: Shapefile, Geo JSON, Geo TIFF, KML – Data wrangling: Geo referencing, reprojection, format transformation – Web-based APIs for spatial data access (Google Maps, Overpass API) – Introduction to QGIS/ArcGIS: Layer import, editing, management – Data versioning, cataloging, and licensing models.

### **Spatial Data Analysis and Visualization**

Data Ingestion Methods (Batch, Stream) - ETL vs ELT in Data Lake- Tools for Data Ingestion (Apache NiFi, Kafka, AWS Glue, Azure Data Factory) - Data Transformation and Processing (Apache Spark, AWS EMR) - Managing Metadata (Apache Atlas, AWS Glue Data Catalog)- Data Lake Optimization (partitioning, compression, indexing)

## **Advanced Concepts and Real-World Applications**

Data Lakehouse Concept - Integration with BI and ML Tools (Power BI, Tableau, SageMaker) - Monitoring and Logging (CloudWatch, Azure Monitor) - Performance Tuning and Cost Optimization - Case Studies: Data Lake Implementation in Enterprises - Future Trends: Serverless Lakes, Data Mesh, AI on Data Lakes.

## **References**

1. Paul Bolstad, "GIS Fundamentals: A First Text on Geographic Information Systems", XanEdu Publishing, 6th Edition, 2019.
2. Kang-Tsung Chang, "Introduction to Geographic Information Systems", McGraw-Hill Education, 9th Edition, 2023
3. Nathalie Japkowicz, Mohak Shah, "Big Data Analytics with Applications in GIS", CRC Press, 2020. Jonathan Raper, "Multidimensional Geographic Information Science", CRC Press, 2000.
4. Ian Heywood, Sarah Cornelius, Steve Carver, "An Introduction to Geographical Information Systems", Pearson Education, 4th Edition, 2011
5. Alan MacEachren, "How Maps Work: Representation, Visualization, and Design", Guilford Press, 2004
6. Michael N. DeMers, "Fundamentals of Geographic Information Systems", Wiley, 5th Edition, 2018
7. Chris Brunsdon, Lex Comber, "An Introduction to R for Spatial Analysis and Mapping", SAGE Publications, 3rd Edition, 2022.

## **Cos.**

1. Understand core geospatial data concepts — coordinate systems, map projections, spatial data types (vector, raster), and GI Science principles.
2. Implement advanced data wrangling techniques to clean, restructure, and integrate data from multiple sources.
3. Apply spatial analysis — overlay, buffer, spatial statistics, interpolation, and raster algebra — to discover patterns.
4. Automate geospatial workflows and custom geo processing through Python scripting and libraries (GeoPandas, Rasterio, PostGIS).
5. Design geospatial visualizations and maps — static thematic maps, interactive dashboards, and story maps with advanced cartographic techniques.
6. Complete a real-world geospatial analytics project from problem definition through reproducible analysis and reporting.

<b>Weightage:</b>	Continuous Assessment: 40%	End Semester Theory Examination: 60%
	(i). Activities: 15% (ii). Internal Theory Examinations: 25%	
<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		

<b>CO</b>	<b>CO Description</b>	<b>Mapped POs</b>	<b>Mapped PSOs</b>
<b>CO1</b>	Understand core geospatial data concepts including coordinate systems, map projections, spatial data models, and GIScience principles.	PO3	PSO1
<b>CO2</b>	Implement advanced data wrangling techniques to clean, restructure, and integrate data from multiple sources.	PO1, PO3	PSO1, PSO2
<b>CO3</b>	Apply spatial analysis techniques such as overlay, buffering, spatial statistics, interpolation, and raster algebra to identify patterns.	PO1, PO3	PSO1
<b>CO4</b>	Automate geospatial workflows and custom geoprocessing using Python libraries and spatial databases.	PO1, PO3	PSO1, PSO2
<b>CO5</b>	Design geospatial visualizations including thematic maps, interactive dashboards, and story maps using advanced cartography	PO2, PO3	PSO1

CP25C25	Federated Learning	L	T	P	C
		3	0	0	3
<p><b>Course Objective:</b></p> <ol style="list-style-type: none"> <li>1. Understand the fundamentals, architecture, and real-world relevance of Federated Learning.</li> <li>2. Learn core FL algorithms, privacy techniques, and system challenges in decentralized settings.</li> <li>3. Apply FL frameworks and tools to build practical applications through hands-on project work.</li> </ol>					
<p><b>Introduction to Federated Learning</b>            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.</p> <p><b>Activity</b>            Create a Quiz for understanding Federated Learning Basics</p> <p><b>Federated Learning Algorithms</b>            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.</p> <p><b>Activity</b>            Create a flipped class room for Understanding federated learning algorithms</p> <p><b>Privacy and Security in FL</b>            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.</p> <p><b>Activity</b>            Ask students to submit an assignment on Privacy and security in FL</p> <p><b>System &amp; Communication Challenges</b>            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.</p> <p><b>Activity</b>            Create a flipped class room communication challenges</p>					

**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)

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and operational principles of federated learning and privacy-preserving machine learning systems.	--	--
CO2	<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)

<b>CP25C24</b>	<b>Vibe Coding</b>	L 3	T 0	P 0	C 3
<p><b>Course Objective:</b> This course introduces the basics of AI-assisted coding and Vibe Coding. Participants will learn to use AI tools, build simple applications, and understand the roles and ethics involved. By the end, they will create their own AI-powered projects.</p>					
<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>• Students take on roles such as AI, prompt engineer, and developer to simulate collaborative coding.</li> <li>• 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>• Students explore AI tools (e.g., Cursor or v0) at home and build a basic app in class.</li> <li>• Groups design a simple development workflow using AI tools and explain tool selection and purpose.</li> </ul>					

### **AI Prompt Design, Oversight, and Ethical Coding Practices**

6Vibe 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

#### **Activities:**

- Team presents a complete AI workflow, including prompt design, context handling, and oversight strategy.
- Class discussion on accountability, code quality, and risks of over-reliance on AI in software development.

### **The Unbundled Programmer and the Future of Coding Roles**

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**

- Showcase how AI transforms developer roles across the SDLC.
- 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:**

- Refactor example AI-generated code and answer related questions on quality and security.
- 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**

- Build a working prototype (e.g., to-do list, resume builder) using AI-assisted tools.
- 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

- Design and deploy a full AI-assisted web application as a team.
- Teams demo their solutions and reflect on human-AI collaboration challenges.

<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. 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

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)

CP25C20	Agentic AI	L	T	P	C
		3	0	0	3
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Understand the foundations and architecture of intelligent agents.</li> <li>• Design agents capable of autonomous and rational decision-making.</li> <li>• Apply reinforcement learning for adaptive agent behavior.</li> <li>• Explore coordination in multi-agent and human-AI systems.</li> <li>• Analyse the ethical implications of agentic systems.</li> <li>• Develop real-world applications using modern agent-based AI frameworks.</li> </ul>					
<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>					
<p><b>Agent Decision-Making and Planning</b></p> <p>Rationality and Utility Theory - Task Environment Analysis - AI Planning Techniques: STRIPS, Classical Planning - Decision-Making under Uncertainty: MDPs, Bayesian Models, Game Theory</p> <p><b>Activity:</b></p> <ul style="list-style-type: none"> <li>• Design a simple application for making decision using AI in python</li> </ul>					
<p><b>Learning in Agentic Systems</b></p> <p>Reinforcement Learning: Q-Learning, SARSA - Value and Policy Iteration - Deep RL: DQNs, Policy Gradient Methods, Actor-Critic - Tools: OpenAI Gym, PettingZoo, TensorFlow RL</p> <p><b>Activity:</b></p> <ul style="list-style-type: none"> <li>• Write a case study for Tools about OpenAI and TensorFlow with proper application</li> </ul>					
<p><b>Multi-Agent Systems and Collaboration</b></p> <p>Multi-Agent Coordination: Cooperation, Competition - Agent Communication and Negotiation - Human-AI Collaboration: Interactive and Mixed-Initiative Systems - Platforms: NetLogo, RoboCup Simulation</p> <p><b>Activity:</b></p> <ul style="list-style-type: none"> <li>• Create Autonomous agents using NetLogo platform</li> </ul>					

## 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>	
<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>		

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and principles of agentic artificial intelligence and autonomous agent systems.	--	--
CO2	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)