MRK INSTITUTE OF TECHNOLOGY

AN AUTONOMOUS INSTITUTION

(Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai)

KATTUMANNARKOIL- 608 301, CUDDALORE DISTRICT, TAMILNADU, INDIA

CHOICE BASED CREDIT SYSTEM

Programme: M.E. COMPUTER SCIENCE AND ENGINEERING **Regulations:** 2025

PROGRAMME OUTCOMES (POs):

PO	Programme Outcomes
PO1	An ability to independently carry out research /investigation and development work to solve practical problems
PO2	An ability to write and present a substantial technical report/document.
PO3	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 (PSOS)

PSO	Programme Specific Outcomes
PSO1	Advanced Computing and Problem Solving: Analyze, design, and implement advanced algorithms, architectures, and computational models to develop sustainable and scalable solutions, aligning with industry and societal needs to solve complex problems in diverse domains.
PSO2	Research and Innovation Competence: Undertake independent research and apply advanced tools and methodologies to propose innovative solutions for real-world and interdisciplinary computing challenges, demonstrating research aptitude.

MRK Institute of Technology 608 301. Kattumannarkoil.

Dr. K. ANANDAVELU, B.E., M.E., Ph.D.
D.I.S., I.G.C.O.H.S.,M.I.S.T.E.,M.S.E.A (T).,
PRINCIPAL
MRK Institute of Technology
Kattumannarkoil - 608 301.

MRK INSTITUTE OF TECHNOLOGY

AN AUTONOMOUS INSTITUTION

(Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai)

KATTUMANNARKOIL- 608 301, CUDDALORE DISTRICT, TAMILNADU, INDIA

Abbreviations:

BS –Basic Science (Mathematics)

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

SD – Skill Development

TCP –Total Contact Period(s)

L –Laboratory Course

T – Theory

LIT –Laboratory Integrated Theory

PW – Project Work

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.	CP25C02	Advanced Database Technologies	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.	CP25101	Technical Seminar	-	0	0	2	2	1	SD
Total Credits							**22**	**19**	

Semester II

S. No.	Course Code	Course Title	Type	Periods per week			TCP	Credits	Category
				L	T	P			
1.	CP25201	Multicore Architectures	LIT	3	0	2	5	4	ES (PC)
2.	CP25C05	Artificial Intelligence and Machine Learning	T	3	0	0	3	3	ES (PC)
3.	CP25C06	Cloud and Big Data Analytics	T	3	0	0	3	3	ES (PC)
4.	CP25C07	Quantum Computing	T	2	0	0	2	2	ES (PC)
5.		Programme Elective I	T	3	0	0	3	3	ES (PE)
6.		Programme Elective - II	T	3	0	0	3	3	ES (PE)
7.		Self-Learning Course	-	-	-	-	-	1	-
						TOTAL	**19**	**19**	

Semester III

S. No.	Course Code	Course Title	Type	Periods per week			TCP	Credits	Category
				L	T	P			
1.		Programme Elective-III	T	3	0	0	3	3	ES (PE)
2.		Programme Elective-IV	T	3	0	0	3	3	ES (PE)
3.		Programme Elective-V	T	3	0	0	3	3	ES (PE)
4.		Industry Oriented Course II	-	1	0	0	1	1	SD
5.	CP25301	Project Work I	-	0	0	12	12	6	SD
						TOTAL	**22**	**16**	

Semester IV

S. No.	Course Code	Course Title	Type	Periods per week			TCP	Credits	Category
				L	T	P			
1.	CP25401	Project Work II	---	0	0	24	24	12	SD
						TOTAL	**24**	**12**	

MRK Institute of Technology ★ Kattumannarkoil. 608 301.

PROGRAMME ELECTIVE COURSES (PE)

S. No.	Course Code	Course Title	Periods Per Week			Total Contact Periods	Credits
			L	T	P		
1.	CP25C08	Advanced Software Testing and Quality Assurance	3	0	0	3	3
2.	CP25C09	Agile Methodologies	3	0	0	3	3
3.	CP25C10	Web of Things	3	0	0	3	3
4.	CP25001	Text and Speech Processing	3	0	0	3	3
5.	CP25C11	Advanced Deep Learning and Neural Networks	3	0	0	3	3
6.	CP25C12	Quantum Cryptography	3	0	0	3	3
7.	CP25C13	Quantum Machine Learning	3	0	0	3	3
8.	CP25C14	AI in IoT	3	0	0	3	3
9.	CP25C15	Web 3.0	3	0	0	3	3
10.	CP25C16	Advanced Large Language Models	3	0	0	3	3
11.	CP25C17	Edge and Fog Computing	3	0	0	3	3
12.	CP25C18	Green Computing and Sustainability	3	0	0	3	3
13.	CP25C19	Cognitive Computing	3	0	0	3	3
14.	CP25C20	Agentic AI	3	0	0	3	3
15.	CP25C21	Mixed Reality	3	0	0	3	3
16.	CP25C22	Blockchains Architecture and Design	3	0	0	3	3
17.	CP25C23	Human-Centered AI	3	0	0	3	3
18.	CP25C24	Vibe Coding	3	0	0	3	3
19.	CP25C25	Federated Learning	3	0	0	3	3
20.	CP25002	Deep Learning for Computer Vision	3	0	0	3	3

SEMESTER I

MA25C07	Advanced Mathematical Methods (CSIE)	L	T	P	C
		3	1	0	4

Course Objectives:

- Develop an in-depth understanding of advanced concepts in linear algebra, multivariate analysis, and number theory for computer science applications.
- Apply mathematical tools such as eigenvalue decomposition, SVD, and multivariate statistical methods to real-world computing and data-driven problems.
- Analyze and implement number-theoretic techniques for cryptography, security, and algorithmic problem-solving in computer science.

UNIT I – ADVANCED LINEAR ALGEBRA 12

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.

UNIT II – MULTIVARIATE ANALYSIS 12

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.

UNIT III – ELEMENTARY NUMBER THEORY 12

Division algorithm, divisibility and the Euclidean algorithm, fundamental theorem of arithmetic, modular arithmetic and basic properties of congruences, principles of mathematical induction and well ordering principle.

UNIT IV – COMPUTATIONAL NUMBER THEORY AND CRYPTOGRAPHIC MATHEMATICS 12

Primality testing algorithms, Chinese Remainder Theorem, quadratic congruence, discrete logarithm, factorization methods, and advanced number-theoretic techniques for algorithmic problem-solving.

UNIT V – SECURITY AND INFORMATION-THEORETIC FOUNDATIONS 12

Side channel attacks, Shannon theory, perfect secrecy, semantic security, and applications of number-theoretic and information-theoretic concepts to cryptography and secure computing.

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).

TOTAL: 60 PERIODS

References:

1. Gilbert Strang, *Linear Algebra and Its Applications*, Cengage Learning.
2. Richard A. Johnson & Dean W. Wichern, *Applied Multivariate Statistical Analysis*, Pearson.
3. Neal Koblitz, *A Course in Number Theory and Cryptography*, Springer.
4. Victor Shoup, *A Computational Introduction to Number Theory and Algebra*, Cambridge University Press.

E-resources:

1. https://ocw.mit.edu/courses/18-06-linear-algebra
2. https://nptel.ac.in/courses/111105041
3. https://crypto.stanford.edu/pbc/notes/numbertheory

MRK Institute of Technology ★ 608 301. Kattumannarkoil. ★

CP25C01	Advanced Data Structures and Algorithms	L	T	P	C
		3	0	4	5

Course Objectives:

1. To explore advanced linear, tree, and graph data structures and their applications.
2. To design efficient algorithms using appropriate algorithmic paradigms.
3. To evaluate computational complexity and identify tractable vs. intractable problems.

UNIT I – LINEAR DATA STRUCTURES AND MEMORY OPTIMIZATION 18

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.

Practical:

1. Implement skip lists and measure performance compared with balanced BST.
2. Experiment with cache-aware data structures and analyze memory utilization.

UNIT II – ADVANCED TREE DATA STRUCTURES 18

Balanced Trees: AVL, Red-Black Trees, Splay Trees, Treaps. Multi-way Trees: B-Trees, B+ Trees, R-Trees. Range and String Structures: Segment Trees, Fenwick Trees, Suffix Trees and Tries for string processing. Applications in indexing, text retrieval, and computational geometry.

.**Practical:**

1. Implement B+ tree for database indexing use-case.
2. Design a suffix tree-based algorithm for DNA sequence matching.

UNIT III – GRAPH DATA STRUCTURES AND ALGORITHMS 18

Graph 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.

Practical:

1. Implement Johnson’s algorithm for sparse graph shortest paths.
2. Demonstration of Maximum flow in traffic or network routing simulation.

UNIT IV – Algorithm Design and Paradigms: 18

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:

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

UNIT V – COMPUTATIONAL COMPLEXITY, APPROXIMATION AND EMERGING ALGORITHMS 18

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. 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:

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

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

TOTAL: 90 PERIODS

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

References:

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

E-resources:

1. https://www.theiotacademy.co/blog/data-structures-and-algorithms-in-c/
2. https://github.com/afrid18/Data_structures_and_algorithms_in_cpp
3. https://www.udemy.com/course/introduction-to-algorithms-and-data-structures- in-c/?srsltid=AfmBOorEZlkgV7QzaEh6IqzAaKLjC-IpFU1NGgWFoHMLhOos- uDVKjCK

CO	Description of CO	PO	PSO
CO1	Describe data structures and implement algorithmic solutions for complex computational problems.	--	--
CO2	Analyze the time complexity and Efficiency of algorithms for various computing problems.	PO1(3)	PSO1(3)
CO3	Evaluate algorithmic techniques and data structures to etermine 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)
CO5	Apply advanced, approximation, randomized, parallel, distributed, and streaming algorithms to solve complex real-world computational problems.	PO4(2) PO5(2)	PSO1(3), PSO2(3)

CP25C02	Advanced Database Technologies	L	T	P	C
		3	0	0	3

Course Objectives:

- To strengthen the understanding of enhanced ER models and their transformation into relational models with indexing and file structures.
- To understand object-oriented and object-relational database concepts and querying using OQL.
- To explore techniques in query processing, execution, and optimization strategies.

UNIT I – ENTITY RELATIONSHIP AND ENHANCED ENTITY RELATIONAL MODELS 9

Entity Relationship Model: Entity Relationship Model Revised – Subclasses, Superclasses and Inheritance – Specialization and Generalization – Union Types – Aggregation.Enhanced Entity Relational Model: Relational Model Revised – Converting ER and EER Model to Relational Model – SQL and Advanced Features – File Structures – Hashing – Indexing.

Activity:

1. Design an ER Model for a specific use case.
2. Demonstration of SQL implementation.
3. Design an ER Model for a specific use case

UNIT II – OBJECT-RELATIONAL DATABASES 9

Object Relational Databases: Object Database Concepts – Object Database Extensions to SQL – The ODMG Object Model and ODL – Object Database Conceptual Design – Object Query Language (OQL) – Language Binding in the ODMG Standard.

Activity:

1. Demonstration of Object Query Language.

UNIT III – QUERY PROCESSING, EXECUTION AND OPTIMIZATION 9

Object Database Concepts-Object Database Extensions to SQL, The ODMG Object Model and ODL, Object Database Conceptual Design-Object Query Language OQL-Language Binding in the ODMG Standard.

Activity:

1. Demonstration of Object Query Language.

UNIT IV – DISTRIBUTED DATABASES AND NOSQL SYSTEMS 9

Distributed Databases: Real-Time Bidding, E-mail Marketing, Affiliate Marketing, Social Marketing, Mobile Marketing – Distributed Database Concepts – Data Fragmentation, Replication and Allocation – Distributed Database Design Techniques – Distributed Database Architectures.NOSQL Systems and Big Data: Introduction to NOSQL Systems – CAP Theorem – Document-Based NOSQL Systems – Key-Value Stores – Column-Based or Wide Column NOSQL Systems – NOSQL Graph Databases and Neo4j.

Activity:

1. Demonstration of concurrency and transactions.
2. Design an application with MongoDB.

UNIT V – ADVANCED DATABASE MODELS, SYSTEMS AND APPLICATIONS 9

Advanced Database Models, Systems and Applications: Active Database Concepts and Triggers – Temporal Database Concepts – Spatial Database Concepts – Multimedia Database Concepts – Deductive Database Concepts – Introduction to Information Retrieval and Web Search.Big Data Infrastructure: Demonstration of Hadoop infrastructure for Big Data Analytics.

Activity:

1. Demonstration of Hadoop infrastructure for Big Data Analytics.

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

TOTAL: 45 PERIODS

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

References:

1. Elmasri, R., & Navathe, S. B. (2016). Fundamentals of database systems. Pearson Education.
2. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). Database system concepts, McGraw Hill Education.
3. Ceri, S., & Pelagatti, G. Distributed databases: Principles and systems. McGraw Hill.
4. Ramakrishnan, R., & Gehrke, J. (2004). Database management systems.McGraw Hill.

E-resources:

1. https://www.edx.org/learn/sql/stanford-university-databases-advanced-topics-in-sql
2. https://www.coursera.org/courses?query=sql&productDifficultyLevel=Advanced

	Description of CO	PO	PSO
CO1	Elaborate different database models for effective database design.	--	--
CO2	Implement advanced database features for optimized data retrieval.	PO1(3)	PSO1(3)
CO3	Evaluate query processing and optimization strategies	PO3(2)	PSO2(2)
CO4	Design solutions using advanced database models to address complex data-intensive	PO2(1)	PSO1(3)
CO5	Apply distributed, NoSQL, big data, and specialized database technologies to develop solutions for complex data-intensive applications.	PO4(2)	PSO1(3), PSO2(3)

CP25C03	Advanced Operating Systems	L	T	P	C
		3	0	0	3

Course Objectives:

- To analyze the architectures and design issues of advanced operating systems.
- To develop the model for process synchronization and recovery in complex environments.
- To evaluate algorithms for distributed coordination, resource management, fault tolerance, and security.

UNIT I – ADVANCED PROCESS, THREAD AND MEMORY MANAGEMENT 9

Advanced Process and Thread Management: Multithreading models, thread pools, context switching, synchronization issues and solutions: semaphores, monitors, lock-free data structures, CPU scheduling in multi-core systems.

Memory and Resource Management in Modern OS: Virtual memory, demand paging, page replacement policies, huge pages, NUMA-aware memory management, resource allocation in cloud-native environments.

Activity:

1. CPU scheduler simulation for multicore systems.
2. Simulate demand paging and page replacement algorithms.

UNIT II – VIRTUALIZATION AND CONTAINERIZATION 9

Virtualization and Containerization: Hypervisors (Type I and Type II), KVM, QEMU, Xen. Containers: Docker, LXC, systemd-nspawn. OS-level virtualization and namespaces.

Activity:

1. Deploy and configure Docker containers with various images.

UNIT III – DISTRIBUTED OPERATING SYSTEMS AND FILE SYSTEMS 9

Distributed Operating Systems and File Systems: Distributed scheduling, communication, and synchronization. Distributed file systems: NFS, GFS, HDFS. Transparency issues and fault tolerance.

Activity:

1. Simulate distributed process synchronization.

UNIT IV – SECURITY, TRUST AND REAL-TIME OPERATING SYSTEMS 9

Security and Trust in Operating Systems: Access control models: DAC, MAC, RBAC. OS hardening techniques, sandboxing, SELinux, AppArmor, secure boot, rootkit detection, trusted execution environments.

Real-Time and Embedded Operating Systems: Real-time scheduling algorithms (EDF, RM), POSIX RT extensions, RTOS architecture, TinyOS, FreeRTOS case studies.

Activity:

1. Implement Role-Based Access Control (RBAC) using Linux user and group permissions.

UNIT V – EDGE, CLOUD AND FUTURE OPERATING SYSTEM PARADIGMS 9

Edge and Cloud OS – Future Paradigms: Serverless OS, unikernels, lightweight OS for edge computing, mobile OS internals (Android, iOS), and OS for quantum and neuromorphic computing (introduction).

Activity:

1. Analyze Android's system architecture using emulator tools.

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

TOTAL: 45 PERIODS

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
2. Prof. Rajiv Misra, "Cloud Computing and Distributed Systems", IIT Patna, NPTEL, https://nptel.ac.in/courses/106104182

CO	Description of CO	PO	PSO
CO1	Describe operating system concepts for memory and resource management.	--	--
CO2	Analyse virtualization and distributed OS mechanisms for scalability and performance.	PO1 (3)	PSO1 (3)
CO3	Evaluate OS security and resource handling strategies in diverse environments.	PO3 (2)	PSO2 (2)
CO4	Design innovative OS solutions using modern tools and	PO2 (1)	PSO1 (3)
CO5	Design and evaluate innovative operating system solutions for edge, cloud, mobile, serverless, and emerging computing	PO4 (2)	PSO1 (3),

		L	T	P	C
CP25C04	**Advanced Compiler Design**	3	0	0	3

Course Objective:

- To analyze the theory and principles of modern compiler design and advanced optimization techniques.
- To design and implement efficient front-end and back-end compiler components for programming languages.
- To evaluate code optimization strategies and runtime environment management in contemporary architectures.

UNIT I – Intermediate Representations and Control Flow Analysis: 9

Static single assignment (SSA) form- Context-Free Grammer (CFG) construction-dominance relations- Intermediate Representation (IR) design for functional and imperative languages-Static single assignment and def-use chains

Activities:

1. Convert source code to SSA form using LLVM IR.
2. Visualize control flow graphs from SSA using LLVM tools.

UNIT II – PROGRAM ANALYSIS AND ADVANCED OPTIMIZATION 9

Data flow analysis- live variable analysis- reaching definitions-Alias analysis and dependence analysis-Loop optimizations and transformations Polyhedral model for loop nests-Tiling, skewing, fusion, and vectorization-Profile-guided and feedback-directed optimizations

Activities:

1. Perform loop unrolling and strength reduction.
2. Conduct live variable analysis and visualize data flow graphs.
3. Implement loop tiling and loop skewing on a matrix multiplication program.
4. Analyze the effect on loop-intensive code with LLVM optimization flags.

UNIT III – JUST-IN-TIME COMPILATION AND MACHINE LEARNING IN COMPILERS 9

JIT compilation models: tracing, method-based-GraalVM architecture, Java HotSpot internals-LLVM JIT and dynamic language support ML for phase ordering, auto-tuning, and IR prediction-Reinforcement learning for optimization passes-Dataset creation and benchmarking for compiler ML

Activities:

1. Develop a basic JIT-enabled interpreter with LLVM or GraalVM.
2. Implement dynamic dispatch using LLVM JIT API.
3. Train an ML model to predict optimization passes.
4. Use reinforcement learning for pass selection in toy compiler.

UNIT IV – Domain-Specific Languages (DSLs) and Compiler Extensions: 9

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.

UNIT V – Security, Verification, and Future Trends: 9

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%

TOTAL: 45 PERIODS

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”

	Description of CO	PO	PSO
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)
CO5	Evaluate secure compilation, formal verification, compiler fuzzing, and emerging compiler technologies for quantum, multi-target, and neuromorphic systems.	PO4(2)	PSO1(3), PSO2(3)

SEMESTER II

		L	T	P	C
CP25201	**Multicore Architecture**	3	0	2	4

Course Objective:

This course enables students to learn parallel programming models, tools, and techniques using shared and distributed memory systems. The course emphasizes the implementation and performance evaluation of parallel applications using OpenMP and MPI.

UNIT I – Multicore Systems and Architectures: 15

Evolution from Single-core to Multicore Systems - SIMD vs MIMD Architectures - Shared and Distributed Memory Systems - Cache Coherence and Memory Consistency

Practical:

1.Write a program to create and execute multiple threads using Pthreads.

2. Write a program to demonstrate mutex locks using std::mutex.

UNIT II – Thread Management and Synchronization Techniques: 15

Thread Management and Synchronization Techniques: Synchronization primitives: Mutex - Locks - Semaphores - Data races and deadlocks - Inter-thread communication: Condition variables- Pipes

Practical:

1.Write a program to compute the Fibonacci series using OpenMP tasks.

2. Write a Rust program to spawn threads and use channels for communication.

UNIT III – PARALLEL PROGRAMMING USING OPENMP AND MPI 15

Parallel Programming using OpenMP for Shared Memory: OpenMP architecture and memory model – Directives and work-sharing constructs – Parallelizing loops and task management – Performance optimization.Message Passing Programming with MPI: MPI programming concepts – Point-to-point and collective communication – Derived data types and synchronization – Performance tuning.

Practical:

1. Write a program using MPI to calculate the average of an array using collective communication.
2. Write a program using the threading module to demonstrate data race and synchronization.
3. Write a program to perform matrix multiplication using OpenMP.

UNIT IV – PARALLEL PERFORMANCE AND SCALABILITY 15

Evaluating Parallel Performance and Scalability: Measuring and analyzing performance – Scalability metrics – Use of profiling tools for performance bottlenecks.

Practical:

1. Write a Rust program to parallelize a for loop using the rayon crate.
2. Write a program and profile its performance using gprof or perf.

UNIT V – ADVANCED CONCEPTS AND PRACTICAL CASE STUDIES 15

Tree search – N-body simulation – Graph algorithms – Comparative analysis of OpenMP and MPI implementations – Trends in multicore and many core processors.

Practical:

1. Mini Project: Write a C program to implement parallel merge sort or breadth-first search using both OpenMP and MPI and compare complexity.

Course Outcomes:

- Explain the architectural features and classifications of multicore processors
- Demonstrate the use of synchronization primitives to ensure correctness in parallel code
- Implement shared memory parallel programs using OpenMP
- Develop scalable MPI programs for distributed memory architectures
- Analyze and compare performance of OpenMP and MPI implementations for real-world problems

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

TOTAL: 75 PERIODS

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

References:

1. Michael J. Quinn, Parallel Programming in C with MPI and OpenMP, McGraw- Hill, 2004.
2. Yan Solihin, Fundamentals of Parallel Multicore Architecture, CRC Press, 2015.
3. Victor Alessandrini, Shared Memory Application Programming, Morgan Kaufmann, 2015.
4. Rohit Chandra et al., Parallel Programming in OpenMP, Morgan Kaufmann,

E-Resources:

1. NPTEL Course: https://nptel.ac.in/courses/106104189
2. OpenMP Official Documentation: https://www.openmp.org
3. MPI Forum: https://www.mpi-forum.org

CO-PO-PSO Mapping

CO	**Description of CO**	**PO**	**PSO**
CO1	Describe the fundamental concepts, evolution, and omponents of multicore processor architectures and their role in modern computing systems.	--	--
CO2	Analyze the performance, parallelism, and scalability issues of multicore architectures using appropriate architectural and programming models.	PO1 (3)	PSO1 (3)
CO3	Evaluate different multicore design techniques, cache coherence protocols, and interconnection networks for their effectiveness in specific computing scenarios.	PO3 (2)	PSO2 (2)
CO4	Design efficient multicore-based solutions by selecting suitable architectures, synchronization mechanisms, and parallel execution strategies for real-world applications.	PO2 (1)	PSO1 (3)
CO5	Design and compare multicore and manycore solutions for real-world applications using parallel algorithms and appropriate OpenMP and MPI implementations.	PO4(2)	PSO1(3), PSO2(3)

CP25C05	Artificial Intelligence and Machine Learning	L	T	P	C
		3	0	0	3

Course Objectives:

• To understand the fundamentals of Artificial Intelligence and its core problem-solving techniques using intelligent agents, search strategies, and logic-based reasoning.

• To comprehend the theoretical foundations and algorithmic frameworks of

UNIT I – PROBLEM SOLVING AND KNOWLEDGE REPRESENTATION 9

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.

Activities:

1. Develop a simple Intelligent Agent that navigates a grid world using search algorithms.

UNIT II – UNCERTAIN KNOWLEDGE AND REASONING 9

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.

Activities:

Develop a simple Intelligent Agent that makes decisions in adversarial conditions.

UNIT III – MACHINE LEARNING FUNDAMENTALS 9

Types of learning – hypothesis space – inductive bias – evaluation – cross validation – bias variance – bias variance trade-off.

Activities:

1. Develop a simple Intelligent Agent that uses logical reasoning to infer hidden facts or constraints.

UNIT IV – SUPERVISED LEARNING **9**

Linear Regression – Logistic Regression – Decision Trees: classification and regression trees – Neural networks and multilayer perceptron – Support Vector Machines with linear and non-linear kernel functions – K nearest neighbours – Ensemble learning: bagging and boosting.

Activities:

1. Behaviour Modelling, Disease Diagnosis and similar applications suitable for supervised learning.

UNIT V – UNSUPERVISED AND PROBABILISTIC LEARNING **9**

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.

Activities:

1. Customer Segmentation, Image Processing and similar applications suitable for unsupervised and probabilistic learning.

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

TOTAL: 45 PERIODS

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

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

E-resources:

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

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, scope, and applications of artificial intelligence and machine learning in solving real-world problems.	--	--
CO2	Analyze various artificial intelligence and machine learning models to understand their behavior, performance, and suitability for different problem domains.	PO1 (3)	PSO1 (3)
CO3	Evaluate machine learning algorithms and artificial intelligence techniques based on accuracy, efficiency, and applicability to specific use cases.	PO3 (2)	PSO2 (2)
CO4	Design intelligent systems by selecting and integrating appropriate machine learning models and AI techniques to address complex real-world applications.	PO2 (1)	PSO1 (3)
CO5	Design and evaluate unsupervised and probabilistic learning solutions using clustering, dimensionality reduction and Bayesian learning techniques.	PO4 (2)	PSO1 (3), PSO2 (3)

		L	T	P	C
CP25C06	**Cloud and Big Data Analytics**	3	0	0	3

Course Objectives:

- To understand cloud computing paradigms and service models.
- To study big data characteristics, technologies and storage.
- To explore Hadoop ecosystem and processing tools.
- To analyze, store, and visualize large-scale data.
- To develop and deploy cloud-based big data applications.

UNIT I – CLOUD COMPUTING FUNDAMENTALS **9**

Introduction to Cloud Computing – Service Models: IaaS, PaaS, SaaS – Deployment Models: Public, Private, Hybrid, Community – Virtualization: Hypervisors, VM provisioning – Cloud providers: AWS, Azure, Google Cloud overview.

Activities:

1. Create a free-tier AWS account and explore EC2 and S3.

UNIT II – BIG DATA ECOSYSTEM AND HADOOP **9**

Big Data characteristics: Volume, Velocity, Variety, Veracity, Value – challenges in Big Data – Hadoop Architecture and HDFS – MapReduce Programming Model – YARN.

Activities:

1. Watch a Hadoop tutorial and submit a summary report.

UNIT III – DATA STORAGE AND PROCESSING TECHNIQUES **9**

NoSQL Databases: MongoDB, Cassandra, HBase – Data Ingestion: Sqoop, Flume – Hive, Pig: architecture and querying – data cleaning and preparation.

Activities:

1. Perform HiveQL assignments on datasets from Kaggle.

UNIT IV – ADVANCED BIG DATA ANALYTICS **9**

Introduction to Apache Spark – Spark vs. MapReduce – Spark SQL – Data Frames – RDDs – Machine Learning using Spark MLlib – real-time processing: Kafka and Spark Streaming.

Activities:

1. Develop a mini-project using PySpark on a COVID or traffic dataset.

UNIT V – CLOUD INTEGRATION, SECURITY AND APPLICATIONS **9**

Data on Cloud – storage and compute services – data security and privacy in Cloud – cloud-based data analytics use cases – Cloud-Native vs Traditional Applications – case studies: Smart Cities and Healthcare Analytics.

Activities:

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

TOTAL: 45 PERIODS

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

References:

1. Rajkumar Buyya, James Broberg, Andrzej Goscinski, Cloud Computing: Principles and Paradigms, Wiley.
2. Tom White, Hadoop: The Definitive Guide, O'Reilly Media.
3. Vignesh Prajapati, Big Data Analytics with R and Hadoop, Packt Publishing.
4. Kleppmann, Martin, Designing Data-Intensive Applications, O'Reilly.
5. Alan Gates, Programming Pig, O'Reilly.
6. Jure Leskovec et al., Mining of Massive Datasets, Cambridge University Press.

E-resources:

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

CO	Description of CO	PO	PSO
CO1	Explain cloud computing architectures, service models, deployment models and virtualization technologies.	--	--
CO2	Analyze Big Data characteristics, Hadoop ecosystem components and distributed processing models for scalable data management.	PO1 (3)	PSO1 (3)
CO3	Apply NoSQL, data ingestion, Hive/Pig and data preparation techniques to store and process large-scale datasets.	PO3 (2)	PSO2 (2)
CO4	Implement and evaluate Spark, MLlib, Kafka and Spark Streaming techniques for advanced and real-time Big Data analytics.	PO2 (1)	PSO1 (3)
CO5	Design secure, scalable cloud-based Big Data solutions by integrating cloud services, analytics platforms and real-world application requirements.	PO4 (2)	PSO1 (3), PSO2 (3)

CP25C07	Quantum Computing	L	T	P	C
		2	0	0	2

Course Objective:

1. To provide a mathematical foundation for Quantum Computing and provide the basics of working
2. To interpret the various aspects and applications of quantum computing.
3. To examine the factors that affect Quantum computation

UNIT I – PHYSICAL PROPERTIES AND MATHEMATICAL FOUNDATIONS 6

Double Slit Experiment – Light: Particle vs Wave – Heisenberg Uncertainty Principle – vector spaces, basis, inner product, outer product, tensor product and linear operators.

Activities:

1. Simulate the experiment using an interactive virtual lab; construct simple operators and visualize their action on vectors.

UNIT II – QUANTUM COMPUTING POSTULATES AND GATES 6

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.

Activities:

1. Group quiz to match postulates to physical implications; visualize states with Bloch sphere simulators.

UNIT III – QUANTUM COMPUTING CIRCUITS 6

Dirac notation for quantum computing – computational basis – orthonormality – Hadamard and Phase gates – building quantum circuits.

Activities:

1. Use IBM Q Composer to build and simulate custom circuits; apply X, H and Z gates and observe results.

UNIT IV – FUNDAMENTAL QUANTUM ALGORITHMS 6

Deutsch–Jozsa Algorithm – Grover search algorithm – problem definition – amplitude amplification – Grover oracle – diffuser – multiple solutions in the search space.

Activities:

1. Construct a Deutsch–Jozsa circuit for a 3-bit input function; simulate Grover's algorithm with multiple marked elements.

UNIT V – PROGRAMMING ON A REAL QUANTUM COMPUTER 6

Coding a real-time quantum computer via IBMQ – basic quantum measurement and state analysis – 1- and 2-qubit circuits – quantum noise and comparison of real and simulated results.

Activities:

1. Connect Qiskit with IBMQ; create 1- and 2-qubit circuits and compare real and simulated results.

Text Books:

1. Chuck Easttom, "Quantum Computing Fundamentals", 1st edition, Published by
Addison-Wesley Professional (June 1st 2021)
2. Qiskit TextBook - https://qiskit.org/textbook/preface.html (2022)

References:

1. Kasirajan, Venkateswaran. Fundamentals of quantum computing. Springer
International Publishing, 2021.
2. Chris Bernhardt, Quantum Computing for Everyone, The MIT Press, Cambridge,
2020
3. Nielsen, Michael A., and Isaac L. Chuang, "Quantum Computation and

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

TOTAL: 30 PERIODS

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

CO	Description of CO	PO	PSO
CO1	Describe the fundamental principles, postulates, and computational models of quantum computing and their significance in next-generation computing systems.	--	--
CO2	Analyze quantum algorithms and quantum circuit models to understand their computational advantages, limitations, and performance characteristics.	PO1 (3)	PSO1 (3)
CO3	Evaluate quantum computing paradigms, error correction techniques, and hardware technologies for their suitability in solving complex computational problems.	PO3 (2)	PSO2 (2)
CO4	Design quantum circuits and algorithmic solutions by selecting appropriate quantum gates, qubit architectures, and computational models for real-world problem scenarios.	PO2 (1)	PSO1 (3)
CO5	Design and execute basic quantum programs on real or simulated quantum platforms and evaluate measurement results and quantum noise.	PO4 (2)	PSO1 (3), PSO2 (3)

MRK Institute of Technology
608 301.
Kattumannarkoil.

SEMESTER III

CP25301	Project Work I	L	T	P	C
		0	0	12	6

Course Objectives:

- The main learning objective of this course is to prepare the students for identifying a specific problem for the current need of the society and or industry, through detailed review of relevant literature, developing an efficient methodology to solve the identified specific problem.

Note: A project topic must be selected by the students in consultation with their guides. The progress of the project is evaluated based on a minimum of three reviews. The review committee may be constituted by the Head of the Department. A project report is required at the end of the semester. The project work is evaluated jointly by external and internal examiners constituted by the Head of the Department based on oral presentation and the project report.

SEMESTER IV

CP25401	Project Work II	L	T	P	C
		0	0	24	12

Course Objectives:

- The main learning objective of this course is to prepare the students for solving the specific problem for the current need of the society and or industry, through the formulated efficient methodology, and to develop necessary skills to critically analyse and discuss in detail regarding the project results and making relevant conclusions.

Note: A project topic must be selected by the students in consultation with their guides. The progress of the project is evaluated based on a minimum of three reviews. The review committee may be constituted by the Head of the Department. A project report is required at the end of the semester. The project work is evaluated jointly by external and internal examiners constituted by the Head of the Department based on oral presentation and the project report.

PROGRAMME ELECTIVE COURSES

26SD303	Project Work I	L	T	P	C
		0	0	16	8

Course Objectives:

- The main learning objective of this course is to prepare the students for identifying a specific problem for the current need of the society and or industry, through detailed review of relevant literature, developing an efficient methodology to solve the identified specific problem.

Note: A project topic must be selected by the students in consultation with their guides. The progress of the project is evaluated based on a minimum of three reviews. The review committee may be constituted by the Head of the Department. A project report is required at the end of the semester. The project work is evaluated jointly by external and internal examiners constituted by the Head of the Department based on oral presentation and the project report.

		L	T	P	C
CP25C08	**Advanced Software Testing and Quality Assurance**	3	0	0	3

Course Objectives:

1. Provide an in-depth understanding of advanced software testing techniques and their role in quality assurance.
2. Explore automated testing frameworks, tools, and methods for test design and test process improvement.
3. Enable students to design test plans, develop test cases, and assess test results critically.

UNIT I – FOUNDATIONS, TEST LIFECYCLE AND RISK-BASED TESTING 9

Evolution of software testing in Agile and DevOps ecosystems – Software Test Life Cycle (STLC) from requirements analysis to test closure – quality attributes: reliability, maintainability, testability – testing types and classifications – unit, integration, system, acceptance, regression, smoke and sanity testing – risk-based testing and test prioritization.

Activities:

1. Map STLC to Agile/DevOps pipelines; design a test strategy for risk-based scenarios.

UNIT II – ADVANCED TEST DESIGN TECHNIQUES 9

Black-box and white-box test strategies – BVA – ECP – decision tables – control/data flow – state transition and use-case testing – model-based testing – combinatorial testing – cause-effect graphing – mutation testing – fault injection – fuzz testing – test case minimization and prioritization.

Activities:

1. Create test cases for a real-world problem using model-based design.

UNIT III – TEST AUTOMATION AND CONTINUOUS TESTING 9

Reusable and maintainable automation frameworks – Selenium, TestNG, JUnit, Robot Framework, Cypress – performance testing with JMeter and basic LoadRunner concepts – Jenkins and GitHub integration for continuous testing – BDD and TDD using Cucumber.

Activities:

1. Implement a hybrid automation framework for a web-based application.

UNIT IV – SOFTWARE QUALITY ASSURANCE FRAMEWORKS AND METRICS 9

Quality planning, control, assurance and improvement – reviews, audits, inspections – quality models: McCall, Boehm, ISO 9126/ISO 25010 – metrics: test coverage, defect density, MTBF, test effort – cost of quality and defect leakage.

Activities:

1. Analyze QA strategy in enterprise software such as ERP or banking.

UNIT V – TEST PROCESS IMPROVEMENT AND EMERGING TRENDS 9

Test maturity models: TMMi, CMMI-Dev from QA perspective – test process audits, benchmarking and reviews – empirical methods and defect prediction models – AI/ML-based testing: self-healing tests, anomaly detection – shift-left/right testing – visual and mobile testing.

Activities:

1. Design a test improvement roadmap based on TMMi for a sample company.

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

TOTAL: 45 PERIODS

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

References:

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

E-Resources:

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

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, principles, and processes involved in advanced software testing and quality	--	--
CO2	Analyze software systems and testing strategies to identify defects, assess risks, and ensure compliance with	PO1 (3)	PSO1 (3)
CO3	Evaluate software testing techniques, quality models, and assurance frameworks to determine their	PO3 (2)	PSO2 (2)
CO4	Design comprehensive testing and quality assurance strategies by selecting appropriate testing tools, automation frameworks, and quality metrics for real-world	PO2 (1)	PSO1 (3)
CO5	Design continuous test improvement and AI/ML-enabled quality assurance strategies for modern Agile and DevOps environments.	PO4 (2)	PSO1 (3), PSO2

		L	T	P	C
CP25C09	**Agile Methodologies**	3	0	0	3

Course Objectives:

- Introduce students to the principles, values, and practices of Agile software development.
- Provide hands-on experience with Agile tools and techniques used in industry.
- Develop the ability to plan, manage, and deliver software in Agile environments.

UNIT I – AGILE FOUNDATIONS AND MINDSET 9

Evolution of software development methodologies – challenges in traditional models – Agile Manifesto – Agile values and principles – Agile mindset in individuals and teams – characteristics of Agile teams and leaders – Lean thinking – cultural shift from traditional to Agile.

Activities:

1. Compare Agile and Waterfall in large-scale government or startup projects;

UNIT II – AGILE PROCESS MODELS AND FRAMEWORKS 9

Scrum: roles, artifacts and events – Extreme Programming: pair programming, test-first, refactoring – Kanban: flow, WIP and pull systems – SAFe – Disciplined Agile Delivery – Feature-Driven Development – Crystal – LeSS – Nexus.

Activities:

1. Conduct a team simulation using Scrum ceremonies.

UNIT III – AGILE PROJECT MANAGEMENT AND PLANNING 9

User stories and story mapping – estimation: planning poker, T-shirt sizing – product backlog grooming and sprint planning – Agile metrics: velocity, burndown chart, cumulative flow diagram – risk management in Agile.

Activities:

1. Tool demonstration using Jira, Trello and Azure DevOps.

UNIT IV – AGILE ENGINEERING PRACTICES **9**

Test-Driven Development and Behaviour Driven Development – Continuous Integration and Continuous Delivery – DevOps principles and relationship with Agile – refactoring, code smells and clean code – design patterns in Agile context.

Activities:

1. Automate testing and CI/CD for a sample application.

UNIT V – AGILE AT SCALE AND INDUSTRY APPLICATIONS **9**

Agile scaling frameworks and challenges – distributed Agile teams and remote collaboration – Agile in education, HR and manufacturing – Agile contracting and governance – case studies from Google, Spotify, IBM and startups.

Activities:

1. End-to-end Agile project using Scrum/Kanban and a CI/CD pipeline.

References:

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

E-Resources:

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

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

TOTAL: 45 PERIODS

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

CO	Description of CO	P	PS
CO1	Describe the fundamental principles, values, and frameworks of agile methodologies and their role in modern software development practices.	--	--
CO2	**Analyze** agile processes and practices to understand their effectiveness in managing requirements, collaboration, and iterative software development.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** agile models, project management techniques, and team practices to assess their suitability for different organizational and project contexts.	PO3 (2)	PSO2 (2)
CO4	**Design** agile-based project workflows by selecting appropriate methodologies, roles, and practices to deliver high-quality software solutions.	PO2 (1)	PSO1 (3)
CO5	Design and manage scalable Agile workflows for distributed teams and real-world organizations using appropriate frameworks, tools and practices.	PO4 (2)	PSO1 (3),

		L	T	P	C
CP25C10	**Web of Things**	3	0	0	3

Course Objective:

To equip students with the knowledge and skills to integrate Internet of Things (IoT) devices with web technologies using the Web of Things (WoT) framework. This course enables learners to design interoperable, secure, and scalable smart applications by leveraging standardized WoT architectures, Thing Descriptions, and scripting interfaces.

UNIT I – INTRODUCTION TO WEB OF THINGS 9

Evolution from IoT to WoT – WoT architecture – Thing Description – Binding – Servient – protocol bindings – web protocols: HTTP, WebSockets, MQTT, CoAP – WoT Scripting and interaction models: Properties, Actions, Events.

Activities:

1. Mind-map comparison of IoT vs WoT; analyze WoT reference architecture.

UNIT II – WOT STANDARDS AND THING DESCRIPTION 9

W3C WoT Architecture and Building Blocks – Thing Description syntax and vocabulary – JSON-LD format – TD schema validation and interpretation – semantic annotation in TD.

Activities:

1. Write a Thing Description for a smart device in JSON-LD; validate it using WoT tools.

UNIT III – IMPLEMENTATION OF WOT SERVIENTS 9

WoT Servient structure – exposed vs consumed Things – Node-WoT framework – WoT Scripting API – protocol bindings using HTTP, MQTT and CoAP.

Activities:

1. Build a WoT Servient with Node-WoT; demonstrate the WoT Scripting API.

UNIT IV – INTEROPERABILITY, SECURITY AND PRIVACY 9

Interoperability challenges – HTTPS – OAuth 2.0 – access control – privacy and anonymization – trust and identity in WoT ecosystems.

Activities:

Present security issues in WoT; design a secure architecture for a smart home.

UNIT V – APPLICATIONS, INTEGRATION AND FUTURE TRENDS 9

Smart Cities – Smart Homes – e-Health – IIoT – Cloud and Edge integration – data analytics and AI in WoT – digital twins – Metaverse – integration of Arduino, BeagleBone, Intel Edition and other embedded systems with WoT.

Activities:

Present real-world WoT case studies; create a project integrating an embedded device with WoT.

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

TOTAL: 45 PERIODS

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

References

1. Michael Kuniavsky, *Smart Things: Ubiquitous Computing User Experience Design*, Morgan Kaufmann, 2010.
2. Dom Guinard & Vlad Trifa, *Building the Web of Things*, Manning Publications, 2016.
3. akima Chaouchi, *The Internet of Things: Connecting Objects to the Web*, Wiley, 2010.
4. Peter Waher, *Mastering Internet of Things*, Packt Publishing, 2018.
5. Alasdair Gilchrist, *Industry 4.0: The Industrial Internet of Things*, Apress,

Web resource

W3C WoT Working Group Documents: https://www.w3.org/WoT/H

CO	Description of CO	PO	PSO
CO1	Explain Web of Things concepts, architectures, Thing Descriptions, interaction models and enabling web technologies.	--	--
CO2	Analyze WoT standards, protocol bindings, servient frameworks and communication mechanisms for interoperability and scalability.	PO1 (3)	PSO 1(3)
CO3	Apply Thing Description, JSON-LD, Node-WoT and scripting interfaces to implement interoperable WoT applications.	PO3 (2)	PSO 2(2)
CO4	Evaluate security, privacy, trust and identity mechanisms for developing secure Web of Things ecosystems.	PO2 (1)	PSO 1(3)
CO5	Design scalable WoT solutions integrating embedded devices, cloud/edge platforms, analytics and AI for real-world smart applications.	PO4 (2)	PSO 1 (3), PSO

CP25001	Text and Speech Processing	L	T	P	C
		3	0	0	3

Course Objective:

To introduce the fundamental principles and practices of text and speech processing, equipping students with theoretical understanding and practical skills in natural language and speech technologies, including parsing, information extraction, speech recognition, and synthesis

UNIT I – INTRODUCTION TO TEXT AND SPEECH PROCESSING 9

NLP pipeline overview; human speech production and acoustic phonetics; text normalization: tokenization, stemming, lemmatization; basics of speech signal: sampling, frequency and noise.

Activities:

1. Speech waveform analysis using Praat.
2. Tokenization and stemming with NLTK.

UNIT II – TEXT PROCESSING TECHNIQUES 9

Part-of-speech tagging; parsing techniques: CFG and dependency parsing; Named Entity Recognition (NER); chunking; language modeling: N-gram models.

Activities:

1. POS tagging using spaCy.
2. Building a rule-based NER system.

UNIT III – SPEECH SIGNAL PROCESSING AND ASR 9

Feature extraction: MFCC, PLP, LPC; acoustic modeling: HMM, GMM; basics of Automatic Speech Recognition (ASR); Hidden Markov Models for speech.

Activities:

1. Extract MFCC features using Python.
2. Basic ASR using pre-trained models.

UNIT IV – SPEECH SYNTHESIS AND APPLICATIONS 9

Text-to-speech (TTS) systems; speech databases and corpora; evaluation metrics: WER and CER; applications: voice assistants and dictation systems.

Activities:

1. TTS demonstration using pyttsx3.
2. Compare open-source TTS tools.

UNIT V – NLP AND SPEECH IN PRACTICE 9

Sentiment analysis; chatbots and dialog systems; multilingual and low-resource NLP; ethics in speech and text technology; RNNs and LSTM; Transformer; Large Language Models; Masked Language Models.

Activities:

1. Build a chatbot using Rasa.
2. Debate and compare the advantages and limitations of different NLP
3. models: discuss ethical issues in speech surveillance.

Weightage: Continuous Assessment 40%, End Semester Theory aminations:60%

TOTAL: 45 PERIODS

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

References:

1. Daniel Jurafsky and James H. Martin, Speech and Language Processing, Pearson, 3rd edition draft, 2023.
2. Lawrence Rabiner and Biing-Hwang Juang, Fundamentals of Speech Recognition, Pearson, 2011.
3. Christopher D. Manning and Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.
4. Thomas Schultz and Katrin Kirchhoff, Multilingual Speech Processing, Academic Press, 2006.
5. Jacob Eisenstein, Introduction to Natural Language Processing, MIT Press, 2019.

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, models, and techniques involved in text and speech processing for natural	--	--
CO2	**Analyze** text and speech processing algorithms to understand their performance, accuracy, and limitations in real-world language and audio processing tasks.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** text and speech processing methods, models, and systems to assess their effectiveness for various	PO3 (2)	PSO2 (2)
CO4	**Design** text and speech processing solutions by selecting and integrating appropriate models, features, and	PO2 (1)	PSO1 (3)
CO5	Design intelligent text and speech applications by integrating suitable NLP, speech and modern language models for real-world use cases.	PO4 (2)	PSO1 (3), PSO2

MRK Institute of Technology 608 301. Kattumannarkoil.

CP25C11	Advanced Deep Learning and Neural Networks	L	T	P	C
		3	0	0	3

Course Objective:

This course aims to provide a comprehensive understanding of neural network architectures and their training methodologies. The course explores optimization techniques, regularization methods, and state-of-the-art architectures including encoder-decoder models, attention mechanisms, variational auto encoders, GANs, and transformer-based models.

UNIT I – NEURAL NETWORK FUNDAMENTALS AND REGULARIZATION 9

Perceptron; multilayer feed-forward networks; back propagation; activation functions; loss function; regularization; data augmentation; noise robustness; early stopping; bagging; dropout; batch normalization.

Activities:

1. Understand and demonstrate the working principle of neural networks.

UNIT II – CNN AND RNN ARCHITECTURES 9

Convolutional Neural Networks; convolution operation; input, convolutional, pooling and fully connected layers; Recurrent Neural Networks; LSTM; bidirectional RNNs; RNN language models; word-level RNN; deep recurrent and recursive neural networks.

Activities:

1. Reproduce a research paper related to CNN and RNN.

UNIT III – OPTIMIZATION AND DEEP NETWORK TUNING 9

Non-convex optimization; stochastic optimization; generalization; spatial transformer networks; tuning deep networks; matching input data and network architecture; model goals and output layers; layer and parameter count; memory; weight initialization; activation and loss functions; overfitting.

Activities:

1. Develop a simple three-layer deep neural network application.

UNIT IV – ENCODER-DECODER, ATTENTION AND GENERATIVE MODELS 9

Encoder-decoder models; attention mechanism; attention over images; hierarchical attention; variational autoencoders; autoregressive models; NADE; MADE; PixelRNN; GANs; Graph Convolution Networks; Deep Belief Networks.

Activities:

1. Design a regression, generative or graph model.

UNIT V – ADVANCED DEEP LEARNING APPLICATIONS 9

Object detection: R-CNN, Faster R-CNN and YOLO; MobileNet; DarkNet; object tracking; Audio WaveNet; NLP with Word2Vec; joint detection; face recognition; scene captioning; language transformer models; deep learning for computer vision, time series, text and generative applications.

Activities:

1. Design a face recognition model.
2. Explore medical image segmentation and time-series data analysis.

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

TOTAL: 45 PERIODS

Assessment Methodology: Assignments (30), Quiz (10), Virtual Demo (25),

References

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

MRK Institute of Technology 608 301. Kattumannarkoil.

Online Courses

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

CO-PO-PSO Mapping

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and learning paradigms of deep learning and neural networks used in advanced intelligent systems.	--	--
CO2	Analyze neural network architectures, learning behavior, regularization and optimization characteristics.	PO1 (3)	PSO1 (3)
CO3	Apply training, optimization and tuning techniques to develop effective deep learning models.	PO3 (2)	PSO2 (2)
CO4	Evaluate advanced generative, attention-based, convolutional and recurrent models for complex data-driven applications.	PO2 (1)	PSO1 (3)
CO5	Design deep learning solutions by integrating appropriate architectures, training methods and advanced models for real-world applications.	PO4 (2)	PSO1 (3), PSO2

CP25C12	Quantum Cryptography	L	T	P	C
		3	0	0	3

Course Objective:
The objective of this course is to introduce the principles of quantum information and computation, along with key quantum algorithms and their practical applications. It also aims to provide a clear understanding of post-quantum and advanced quantum cryptographic techniques essential for secure communication in the quantum era.

UNIT I – QUANTUM INFORMATION AND QUANTUM CIRCUITS 9

Qubits; single and multiple qubits; mathematical model for quantum mechanics; quantum measurements; no-cloning; phase shift; bit flips; Hadamard transform; arbitrary transforms; entanglement; quantum gates and circuits: CNOT, CCNOT, reversible and universal gates; quantum parallelism.

Activities

- Build and simulate quantum circuits using Qiskit (or another quantum simulator)
- Create a Bell state using a Hadamard gate followed by a CNOT gate.

UNIT II – QUANTUM ALGORITHMS 9

Deutsch algorithm; phase kickback; Deutsch-Jozsa algorithm; Simon's algorithm; Quantum Fourier Transform; Grover's algorithm; Shor's algorithm.

Activities

- Simulate both Deutsch’s algorithm (1-bit) and the Deutsch-Jozsa algorithm (n-bit) using Qiskit.

UNIT III – QUANTUM CRYPTANALYSIS 9

Quantum order finding; factoring; discrete logarithms; hidden subgroup problems; Diffie-Hellman problems; computational and decisional DH; IND-CPA; applications to RSA and Elgamal encryption.

Activities:

1. Simulate Shor's algorithm to factor small numbers using Qiskit or another quantum simulator.

UNIT IV – POST-QUANTUM CRYPTOGRAPHY 9

Post-quantum cryptography; lattices, codes and isogenies; lattice problems; Learning With Errors (LWE); Short Integer Solution (SIS); dihedral hidden subgroup problem; public-key encryption from LWE; Fully Homomorphic Encryption (FHE).

Activities:

1. Implement a simplified LWE-based public-key encryption scheme using Python or SageMath.

UNIT V – ADVANCED QUANTUM CRYPTOGRAPHY **9**

Quantum Key Distribution and bit commitment; random oracles; Quantum One-Time Pad and encryption; quantum PKE; quantum FHE; quantum indifferentiability; quantum money; QKD with imperfect devices; beyond point-to-point QKD; device-independent quantum cryptography.

Activities:

1. Simulate BB84 for quantum key distribution using Qiskit.
2. Simulate QKD between two IoT-based devices using Python and Qiskit.

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

TOTAL: 45 PERIODS

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

References:

1. Quantum Computation and Quantum Information, M. A. Nielsen and I. Chuang, Cambridge University Press, 2012.
2. Quantum Computing from the Ground Up, Riley Tipton Perry, World Scientific Publishing Ltd., 2012.
3. Quantum algorithms via linear Algebra Primer, Richard J. Lipton Kenneth W. Regan, The MIT Press Cambridge, 2014.
4. Quantum Computing: An Applied Approach, Jack D. Hidary, 1st Edition, Springer, 2019.

E-resources:

1. Quantum Algorithms and Cryptography (Video) – NPTEL
2. Introduction to Quantum Computing: Quantum Algorithms and Qiskit (Video) – NPTEL
3. Practical Quantum Computing with IBM Qiskit for Beginners (Video) – Coursera

CO-PO-PSO Mapping

CO	Description of CO	PO	PSO
CO1	Describe the fundamental principles, protocols, and security concepts of quantum cryptography and their significance in secure communication systems.	--	--
CO2	Analyze quantum cryptographic protocols and security mechanisms to understand their robustness, vulnerabilities, and performance characteristics.	PO1 (3)	PSO1 (3)
CO3	Evaluate quantum cryptographic techniques and implementations to assess their effectiveness in ensuring confidentiality and secure key distribution.	PO3 (2)	PSO2 (2)
CO4	Design secure communication systems by applying appropriate quantum cryptographic protocols, architectures, and security strategies for real-world applications.	PO2 (1)	PSO1 (3)

CO5	Design secure communication systems using appropriate quantum, post-quantum and advanced cryptographic protocols for real-world applications.	PO4 (2)	PSO1 (3), PSO2 (3)

CP25C13	Quantum Machine Learning	L	T	P	C
		3	0	0	3

Course Objective:

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.

UNIT I – QUANTUM COMPUTING FOUNDATIONS FOR QML 9

Qubits, superposition and entanglement; quantum gates, circuits and measurement; Grover's and Shor's algorithms; introduction to QML; classical versus quantum ML; applications of QML.

Activities:

1. Implement a Bell state using Qiskit.
2. Implement and visualize Grover's search algorithm.

UNIT II – QUANTUM INFORMATION THEORY AND ALGORITHMS 9

Quantum entropy and information; quantum teleportation; superdense coding; no-cloning theorem; quantum cryptography; Quantum Fourier Transform; Quantum Phase Estimation.

Activities:

1. Implement a 3-qubit QFT using Qiskit.
2. Simulate quantum teleportation.

UNIT III – QUANTUM ALGORITHMS FOR MACHINE LEARNING 9

Quantum Support Vector Machines (QSVM); Quantum Principal Component Analysis (QPCA); Quantum k-Nearest Neighbors (QkNN); Quantum Neural Networks (QNN).

Activities:

1. Implement QSVM on a small dataset such as Iris.
2. Perform QPCA on a synthetic dataset.

UNIT IV – QUANTUM OPTIMIZATION AND HYBRID MODELS 9

Variational Quantum Eigensolver (VQE); Quantum Approximate Optimization Algorithm (QAOA); hybrid quantum-classical algorithms; applications of quantum optimization in machine learning.

Activities:

1. Implement VQE using Qiskit.
2. Implement QAOA for a simple Max-Cut problem.

UNIT V – APPLICATIONS, ADVANCED QML AND FUTURE TRENDS 9

QML applications in finance, healthcare, NLP and drug discovery; Qiskit and TensorFlow Quantum; future challenges and opportunities; quantum mechanics and data-driven physics; kernelizing quantum mechanics; qubit maps; transverse-field Ising models; variational algorithms; neural network quantum states; quantum reservoir computing.

Activities:

1. Build a quantum classifier using TensorFlow Quantum.
2. Create a hybrid quantum-classical recommendation system.

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

TOTAL: 45 PERIODS

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/
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)	PSO 1(3)
CO3	Evaluate quantum machine learning techniques and frameworks to assess their effectiveness for complex	PO3 (2)	PSO 2(2)
CO4	Design quantum machine learning solutions by selecting appropriate quantum circuits, learning models, and	PO2 (1)	PSO 1(3)
CO5	Design quantum machine learning solutions using suitable quantum circuits, learning models and hybrid architectures for real-world applications.	PO4 (2)	PSO 1 (3), PSO

CP25C14	AI in IoT	L	T	P	C
		3	0	0	3

Course Objective:

- To gain a deep understanding of AI concepts, IoT architectures, and their convergence for intelligent systems.
- To build and train machine learning and deep learning models suited for IoT data processing using TensorFlow / Keras.
- To apply data normalization, filtering, and feature engineering to enhance AI model accuracy
- To deploy AI models on edge devices for low-latency decision-making in IoT environments.
- To implement AI-based cyber security solutions for intrusion detection and anomaly detection in IoT networks.

UNIT I – INTRODUCTION TO AI IN IOT **9**

Fundamentals of AI and IoT - IoT Architecture and AI Integration - AI Models for IoT (Machine Learning, Deep Learning) - AI-Driven IoT Applications (Smart Cities, Healthcare, Industrial IoT) - Ethical Considerations in AI- Enabled IoT – Case Study: Smart Agriculture with AI-Enabled IoT Sensors.

Activities:

1. Analytical study of AI integration within IoT ecosystems, focusing on architectural layers, data flow, and intelligence distribution across device, edge, and cloud.
2. • Simulation-based generation and statistical analysis of IoT sensor data using computational tools (e.g., Jupyter Notebook) to interpret trends, patterns, and variability.

UNIT II – Data Acquisition & Pre-processing in AI-Enabled IoT: **9**

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

Activities:

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

UNIT III – AI Models for IoT Data Processing & Decision-Making: 9

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

Activities

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

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

Activities:

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

UNIT V – INDUSTRY TRENDS, ETHICS AND FUTURE AI-IOT 9

AI in smart cities, healthcare and industrial IoT; scaling large-scale deployments; sustainability and scalability; AI business models and monetization; Green AI; IoT regulations and standardization; Edge AI advances; Quantum AI and IoT interoperability; sustainable smart-city case study.

Activities:

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

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

TOTAL: 45 PERIODS

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

MRK Institute of Technology 608 301. Kattumannarkoil.

References

1. Fadi Al-Turjman, "Artificial Intelligence in IoT", Springer, 2019
2. Ambika Nagaraj," The Role of AI in Enhancing IoT-Cloud Applications", Bentham Books, 2023
3. Rajkumar Buyya and Amir Vahid Dastjerdi, "Internet of Things: Principles and Paradigms", Elsevier, 2016
4. Jason Ioannou, "Data Analytics for IoT and Smart Systems", Springer, 2020
5. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, "Deep Learning", MIT Press, 2016
6. Weisong Shi, "Edge Computing: Vision and Challenges", Springer, 2020
7. Mauro Conti, "Security and Privacy in the Age of IoT", Springer, 2020
8. François Chollet, "Deep Learning with Python" (2nd Edition), Manning Publications, 2021
9. Stuart Russell & Peter Norvig, "Artificial Intelligence: A Modern Approach (4th Edition), Pearson, 2020
10. Arshdeep Bahga & Vijay Madisetti, "Internet of Things: A Hands-On Approach" 2nd Edition), Universities Press, 2016

Web References

https://www.springer.com/gp/campaign/artificial-intelligence/artificial-intelligence-in-iot https://iotworldmagazine.com/2025/01/16/2675/list-of-top-50-best-ai-automation-tools- in-2025
https://www.opensourceforu.com/2025/02/open-source-ai-frameworks-integrating-ai-with-iot/

CO	Description of CO	P	PSO
CO1	Describe the fundamental concepts, architectures, and applications of integrating artificial intelligence techniques with Internet of Things (IoT) systems.	--	--
CO2	Analyze AI-enabled IoT systems to understand data processing,learning models, and decision-making mechanisms in distributed environments.	PO1 (3)	PSO1 (3)
CO3	Evaluate AI techniques and IoT platforms to assess their effectiveness, scalability, and reliability for intelligent connected applications.	PO3 (2)	PSO2 (2)
CO4	Design intelligent IoT solutions by selecting appropriate AI models, sensing strategies, and system architectures for real- world applications.	PO2 (1)	PSO1 (3)
CO5	Design intelligent and sustainable IoT solutions by integrating appropriate AI models, sensing strategies, edge deployment and regulatory considerations.	PO4 (2)	PSO1 (3), PSO2 (3)

CP25C15	Web 3.0	L	T	P	C
		3	0	0	3

Course Objective:

- To understand the progression from Web 1.0 and Web 2.0 to Web 3.0, with an emphasis on decentralization, semantic interoperability, and enhanced privacy.
- To gain hands-on expertise in block chain technology, consensus algorithms, smart contract development, and dApp design by employing open standards.
- To deploy NFT tokenization models and digital identity systems in accordance with W3C standards.
- To develop the ability to integrate emerging technologies (IoT, AI, Semantic Web)
while ensuring accessibility, security, and global interoperability.
- To critically assess decentralized governance models and regulatory challenges, proposing innovative solutions based on industry best

UNIT I – WEB 3.0 FOUNDATIONS 9

Evolution from Web 1.0 to Web 2.0 and Web 3.0; decentralization; semantic interoperability; privacy; layered structure: blockchain, decentralized storage, P2P networks and Semantic Web protocols.

Activities:

1 Present historical transitions and discuss related research papers.

UNIT II – BLOCKCHAIN AND DISTRIBUTED LEDGER SYSTEMS 9

Blockchain architecture; consensus mechanisms; Proof-of-Work, Proof-of-Stake and hybrid models; advantages and limitations; Directed Acyclic Graphs; RESTful API design following open standards.

Activities:

1. Compare consensus mechanisms.
2. Reproduce comparative consensus studies using simulation tools.

UNIT III – SMART CONTRACTS, DAPPS AND DIGITAL ASSETS 9

Smart contract design, deployment and management; decentralized applications; blockchain back-end integration with front-end systems; HTML5, semantic HTML and ARIA/WCAG; NFT tokenization; ERC-721/1155; JSON-LD metadata; oracle services; decentralized storage; IPFS and Filecoin.

Activities:

1. Develop a team-based dApp.
2. Develop an NFT smart contract and explore decentralized storage.

UNIT IV – SECURITY, DIGITAL IDENTITY AND P2P NETWORKING 9

Security, scalability and interoperability; digital identity; decentralized identity (DIDs); self-sovereign identity; W3C DID and verifiable credentials; P2P networking and decentralized storage.

Activities:

1. Develop and test decentralized storage and identity modules.
2. Analyze research on decentralized identity and storage.

UNIT V – DECENTRALIZED GOVERNANCE AND TECHNOLOGY 9INTEGRATION 9

DAOs; token-based governance; DAO structure and voting; regulatory and economic frameworks; Semantic Web protocols RDF, OWL and SPARQL; AI for data curation; IoT for real-time data collection; open-standard compliance.

Activities:

1. Analyze DAO governance and regulatory research.
2. Prototype DAO governance and IoT integration solutions.

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

TOTAL: 45 PERIODS

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

References

1. Prabhat Kumar Srivastav, Prateek Singhal, Basudeo Singh, Nitin Sharma, "Web 3.0 | The Next Generation's Internet and Understanding the Concept", CRC Press, 2024.

2. Imran Bashir, "Mastering Blockchain: Unlocking the World of Cryptocurrencies", Latest Edition, Packt Publishing, 2023.

3. Andreas M. Antonopoulos and Gavin Wood, "Mastering Ethereum: Building Smart Contracts and DApps", 2nd Edition, O'Reilly Media, 2018.

4. Alex Tapscott, "Web3: Charting the Internet's Next Economic and Cultural Frontier, Portfolio", 2023.

Web References

1. https://ipfs.tech/
2. https://docs.filecoin.io/
3. https://identity.foundation/
4. https://www.w3.org/TR/did-1.0/
5. https://www.aragon.org/

CO	**Description of CO**	**PO**	**PSO**
CO1	Describe the fundamental concepts, technologies, and architectural principles of Web 3.0 and their impact on next-generation internet applications.	--	--
CO2	Analyze Web 3.0 platforms and decentralized technologies to understand their functionality, performance, and security implications.	PO1 (3)	PSO1 (3)
CO3	Evaluate Web 3.0 technologies, protocols, and applications to assess their suitability for decentralized and user-centric systems.	PO3 (2)	PSO2 (2)
CO4	Design Web 3.0–based applications by selecting appropriate decentralized architectures, protocols, and technologies for real- world use cases.	PO2 (1)	PSO1 (3)
CO5	Design interoperable Web 3.0 solutions by selecting appropriate decentralized architectures, protocols and technologies for real-world use cases.	PO4 (2)	PSO1 (3), PSO2 (3)

		L	T	P	C
CP25C16	**Advanced Large Language Models**	3	0	0	3

Course Objective:

This course provides the basic knowledge to understand and harness the capabilities of cutting-edge language models. Large Language Models (LLM) have revolutionized the way we interact with text, enabling us to communicate, analyse. and generate content with unprecedented sophistication. This course will help the students to learn LLM from the introduction to the uses and limitations, the inner workings of LLMs, equipping the students with thc tools to explore their versatile applications in content creation, code generation, sentiment analysis, and more.

UNIT I – BUILDING BLOCKS OF MODERN LLMS 9

Large Language Model basics; LLM architecture; Transformer architecture; Attention; encoder-decoder models.

Activities:

1. Study LLM building blocks and Transformer architecture.
2. Analyze a real-world LLM application.

UNIT II – PRE-TRAINING AND LANGUAGE MODELING 9

Pre-training and language modeling; autoregressive and auto-encoder language modeling; early decoder and auto-encoder pretraining; transformer decoding pretraining by denoising; masked language modeling; BERT pretraining; pretraining data and clean-text preprocessing; scaling web data; decoding strategies.

Activities:

1. Implement auto-regressive and auto-encoder language modeling flow.
2. Implement and experiment with decoding strategies.

UNIT III – PARAMETER-EFFICIENT TUNING AND EXPLAINABILITY 9

Parameter Efficient Tuning Methods: prefix tuning, prompt tuning, adapters, compacters, layer freezing, bias terms, fine-tuning, diff pruning and LoRA; advantages and pitfalls; explainability and LLMs; ethical considerations.

Activities:

1. Implement LoRA on a standard dataset and analyze the results.

UNIT IV – IN-CONTEXT LEARNING, PROMPT ENGINEERING AND ALIGNMENT 9

In-context learning; fine-tuning; zero-shot and few-shot learning; prompting; instruction prompting; chain-of-thought prompting; prompt selection; automatic prompt design; visual question answering; multilingual sentiment training; CLIP.

Activities:

1. Build a question-answering system.
2. Perform multilingual sentiment analysis.
3. Experiment with zero-shot/few-shot learning and prompt selection.

UNIT V – GENERATION, EVALUATION AND APPLICATIONS 9

Generation-based automatic evaluation; human evaluation; intrinsic versus extrinsic evaluation; ranking and multiple metrics; GLUE-GEM; quantitative and qualitative evaluation; human annotation; reporting; evaluation challenges and metrics; practical LLM applications in content creation, code generation and sentiment analysis.

Activities:

1. Analyze GLUE-GEM scores for training and testing datasets.

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

TOTAL: 45 PERIODS

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

References

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

E-Resources

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

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and capabilities of advanced large language models and	--	--
CO2	**Analyze** large language model architectures and training methodologies to understand their performance,	PO1 (3)	PSO1 (3)
CO3	**Evaluate** large language models and deployment strategies to assess their effectiveness, ethical	PO3 (2)	PSO2 (2)
CO4	**Design** applications and systems leveraging advanced large language models by selecting appropriate	PO2 (1)	PSO1 (3)
CO5	Design LLM-based applications by selecting suitable architectures, tuning strategies, prompting methods and evaluation frameworks.	PO4 (2)	PSO1 (3), PSO2

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

Course Objective:

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

UNIT I – CLOUD TO EDGE COMPUTING PARADIGMS 9

Evolution from cloud to fog and edge computing; IoT architecture requirements and latency limitations; fog vs edge vs cloud; Multi-access Edge Computing (MEC); multi-tier distributed architectures.

Activities:

1. Case study: IoT solution using fog or edge.
2. Discuss business opportunities in edge markets.

UNIT II – EDGE OFFLOADING AND RESOURCE OPTIMIZATION 9

Edge offloading strategies and placement issues; latency, energy efficiency, QoS and privacy; queuing theory; system performance metrics; optimization using machine learning for slice orchestration; cloud-to-fog-to-things integration; network slicing for 5G/6G; resource virtualization with SDN, NFV and containers; resource-management modeling and simulation.

Activities:

1. Compare cloud and edge performance.
2. Simulate latency or offloading.
3. Study 5G network slicing for vertical applications.

UNIT III – EDGE MIDDLEWARE AND DISTRIBUTED SYSTEMS 9

Middleware design goals: heterogeneity, discovery and scalability; service orchestration; lightweight Docker/Kubernetes containers on constrained devices; edge cluster formation and node coordination; MQTT, CoAP and microservices; distributed storage, synchronization and fault recovery.

Activities:

1. Prototype edge middleware architecture.
2. Deploy containers using Docker/Kubernetes.
3. Analyze vulnerabilities in edge middleware.

UNIT IV – INTELLIGENCE, ANALYTICS, SECURITY AND PRIVACY 9

Fog-layer data management: caching, placement and lifecycle; federated learning and distributed ML; stream and real-time analytics; smart-city, healthcare and smart-home use cases; ML-based threat detection; behavioral analysis; differential privacy and federated anonymization; blockchain for resource security; software-defined and intent-based edge networks; green edge computing.

Activities:

1. Implement predictive analytics using Kafka and Flink.
2. Develop an IoT threat-detection model.
3. Study privacy-preserving fog computing and edge security.

UNIT V – FUTURE DIRECTIONS AND EMERGING TECHNOLOGIES 9

Edge AI and TinyML; neural compression; quantization; model deployment; 6G vision; ultra-low latency; terahertz communications; digital twins; serverless edge; cognitive networks; sustainable and secure edge architectures.

Activities:

1. Develop a research project integrating emerging technologies with edge computing.
2. Present future research directions.

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

TOTAL: 45 PERIODS

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	PO2, PO3	PSO1
CO2	Design and optimize network slicing strategies for integrated cloud–fog–edge systems using mathematical modeling and optimization	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

		L	T	P	C
CP25C18	**Green Computing and Sustainability**	3	0	0	3

Course Objective:

- To understand the principles of green computing.
- To assess environmental impacts of computing technologies and need for green computing.
- To explore sustainable IT practices and energy-efficient hardware system for sustainability growth.

UNIT I – INTRODUCTION TO GREEN COMPUTING 9

Scope; environmental impact of computing; sustainable computing; power-aware resource management; Green IT; OCED Green IT Framework; greening IT; benefits and challenges; quantifying IT energy efficiency.

Activities:

1. Review e-waste management.

UNIT II – GREEN HARDWARE AND SOFTWARE 9

Hardware life cycle; reuse, recycle and dispose; green design and manufacturing of computers; green software; energy-saving software techniques.

Activities:

1. Demonstrate DAQ tools.

UNIT III – SUSTAINABLE SOFTWARE AND GREEN DATA CENTERS 9

Sustainable software attributes, metrics and methodology; energy-aware software engineering; green algorithms; green data-center infrastructure for energy efficiency; server power management.

Activities:

1. Analyze a project's carbon footprint.
2. Present energy issues in data centers.

UNIT IV – GREEN DATA STORAGE AND GREEN IT MANAGEMENT 9

Energy-efficient data-storage practices; optical interconnects for green computers and data centers; energy-efficient cloud computing; green initiatives; implementation; Green IT laws, standards and protocols; regulatory environment and IT manufacturers.

Activities:

1. Case study on energy-efficient cloud computing.
2. Demonstrate Green IT laws in India.
3. Debate the role of IT professionals in Green Computing.

UNIT V – GOVERNMENT INITIATIVES, REPORTING AND GREEN AUDITS 9

Green building standards; sustainability reporting; green audits; nonregulatory government initiatives; energy assessment and audit practices for computing facilities.

Activities:

1. Perform an energy audit in a computer laboratory using software tools.

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

TOTAL: 45 PERIODS

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

References

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

E-Resources

1. Prof. Deepu Philip, Prof. Amandeep Singh | IIT Kanpur

Link http://onlinecourses.nptel.ac.in/noc21_mg85/preview

2. Prof. Sayak Banerjee | IIT Hyderabad

Link “https://onlinecourses.nptel.ac.in/noc23_me138/preview”

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, principles, and practices of green computing and sustainability in modern information technology systems.	--	--

CO2	Analyze computing systems and technologies to assess their energy efficiency, environmental impact, and sustainability metrics.	PO1 (3)	PSO1 (3)
CO3	Evaluate green computing strategies, sustainable IT practices, and regulatory frameworks for their effectiveness in reducing environmental impact.	PO3 (2)	PSO2 (2)
CO4	Design sustainable computing solutions by selecting energy- efficient architectures, technologies, and practices for environmentally responsible systems.	PO2 (1)	PSO1 (3)
CO5	Design sustainable computing solutions using energy-efficient architectures, technologies, audits and responsible IT practices.	PO4 (2)	PSO1 (3), PSO2 (3)

		L	T	P	C
CP25C19	**Cognitive Computing**	3	0	0	3

Course Objectives:

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

UNIT I – FOUNDATIONS OF COGNITIVE COMPUTING 9

Human cognition; introduction to Cognitive Computing; elements of a cognitive system; design principles for cognitive systems; Artificial Intelligence, Natural Language Processing and Machine Learning in Cognitive Computing.

Activities:

1. Concept quiz on Human Cognition and Cognitive Computing.
2. Research paper review on cognitive systems.

UNIT II – COGNITIVE ARCHITECTURES, KNOWLEDGE AND LEARNING 9

Cognitive architectures: ACT-R, Soar and OpenCog; knowledge representation and reasoning; machine learning models in cognitive systems; human-like learning: supervised, unsupervised and reinforcement learning.

Activities:

1. Demonstration of IBM Watson APIs (Language Translator).

UNIT III – NLP AND SEMANTIC UNDERSTANDING 9

Role of NLP in a cognitive system; NLP pipeline: POS tagging, NER and parsing; sentiment analysis; question answering; semantic Web; semantic understanding; ontologies and knowledge graphs.

Activities:

1. Debate on sentiment analysis and its challenges in regional languages.
2. Ideathon on innovative solutions to healthcare problems.

UNIT IV – COGNITIVE AGENTS AND APPLICATION DEVELOPMENT 9

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; process of building a cognitive application; IBM Watson as a cognitive system.

Activities:

1. Mini project: Develop a healthcare chatbot.
2. Prototype demonstrations.

UNIT V – APPLICATIONS, BUSINESS IMPLICATIONS AND EMERGING TRENDS 9

Cognitive computing applications in healthcare, finance, education, legal and robotics; building a cognitive healthcare application; business implications of cognitive computing; cognitive cloud; cognitive IoT; cognitive chatbots; neuromorphic computing and brain-inspired models.

Activities:

1. Prototype demonstrations.

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

TOTAL: 45 PERIODS

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

E-Resources

- Prof. Sharba Bandyopadhyay, Prof. Rajlakshmi Guha | IIT Kharagpur, IIT Kharagpur
Link 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)
CO5	Design cognitive computing solutions by integrating suitable agents, knowledge models, learning techniques and application technologies for real-world domains.	PO4 (2)	PSO1 (3), PSO2 (3)

CP25C20	Agentic AI	L	T	P	C
		3	0	0	3

Course Objectives:

- Understand the foundations and architecture of intelligent agents.
- Design agents capable of autonomous and rational decision-making.
- Apply reinforcement learning for adaptive agent behavior.
- Explore coordination in multi-agent and human-AI systems.
- Analyse the ethical implications of agentic systems.
- Develop real-world applications using modern agent-based AI frameworks.

UNIT I – FOUNDATIONS OF INTELLIGENT AGENTS 9

Introduction to Agentic AI: history, motivation and applications; agents and environments; sensors and actuators; environment types; simple reflex, goal-based, utility-based and learning agents; reactive, deliberative, hybrid and subsumption architectures.

Activities:

1. Prepare a report on types of agents, their roles and architectures.

UNIT II – AGENT DECISION-MAKING AND PLANNING 9

Rationality and utility theory; task environment analysis; AI planning techniques including STRIPS and classical planning; decision-making under uncertainty using Markov Decision Processes, Bayesian models and game theory.

Activities:

1. Design a simple AI decision-making application in Python.

UNIT III – LEARNING IN AGENTIC SYSTEMS 9

Reinforcement Learning: Q-Learning and SARSA; value and policy iteration; deep reinforcement learning: DQNs, policy gradient methods and actor-critic; tools including OpenAI Gym, PettingZoo and TensorFlow RL.

Activities:

1. Case study using OpenAI and TensorFlow tools with an appropriate application.

UNIT IV – MULTI-AGENT SYSTEMS AND COLLABORATION 9

Multi-agent coordination: cooperation and competition; agent communication and negotiation; human-AI collaboration; interactive and mixed-initiative systems; platforms: NetLogo and RoboCup Simulation.

Activities:

1. Create autonomous agents using the NetLogo platform.

UNIT V – LLM AGENTS, EMBODIED COGNITION, ETHICS AND APPLICATIONS 9

Language-based agents: tool use, planning and reasoning; memory-augmented and situated agents; agents in robotics and smart environments; Unity ML-Agents and Hugging Face tools; AI alignment, fairness and explainability; social and environmental implications; case studies in autonomous vehicles and conversational agents; course project evaluation and presentations.

Activities:

1. Design a simple LLM-based tool-using agent application.
2. Develop 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

TOTAL: 45 PERIODS

Weightage:	**Continuous Assessment: 40%** **(i) Activity: 10%**	**End Semester Theory**
	(ii) Internal Theory Examination:	

Mandated Activities with Marks:
Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Internal Examinations: Two Tests

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)
CO3	Evaluate agentic AI frameworks, learning strategies, and coordination techniques to assess their effectiveness in complex and dynamic environments.	PO3 (2)	PSO2 (2)
CO4	Design agentic AI solutions by selecting appropriate agent architectures, learning methods, and coordination mechanisms for real-world applications.	PO2 (1)	PSO1 (3)
CO5	Design ethical and effective agentic AI solutions by integrating suitable architectures, learning methods, tools and coordination mechanisms for real-world applications.	PO4 (2)	PSO1 (3), PSO2 (3)

CP25C21	Mixed Reality	L	T	P	C
		3	0	0	3

Course Objective:
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.

UNIT 1 - Defining Virtual and Augmented Reality 9
Four Key Elements of Virtual Reality Experience Looking at Some Other Types of
Virtual and Augmented Reality.

Activity:
- Assignment about Virtual Reality Experience for real world applications

UNIT 2 - Exploring the Current State of Virtual Reality & Augmented Reality 9
Looking at the Available Form Factors, Focusing on Features. Room-scale versus
stationary experience. Inside-out tracking. Tracking devices Haptic feedback, Audio, Considering Controllers. Mobile devices AR headsets. AR glasses. Hand tracking. Motion controllers Hand and Gesture Tracking, Whole Body Tracking
Activity:
1. Create a flipped class room for exploring the current state of Augmented Reality

UNIT 3 - Simple VR Environment Development 9
Interacting with the Virtual World, Interactions: Manipulation, Navigation, and Communication, **Creating** 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.

Activity:
1. Create a Simple Virtual Environment for a simple real - world application

UNIT 4 – Creating 2D/3DProjects 9
Getting started with Unity, Installing Unity Hub and Unity Editor, Editor interface, Creating Projects with templates, Asset store, creating a simple 2d game using game objects, Scripting in c#, Visual Scripting, creating a simple 3D game by importing game objects from asset store, Building the projects to desktop OS/ Mobile OS

Activity:
1. Create a simple 3D game using Visual Scripting

UNIT 5 - Setting Up for AR Development **9**

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:

1. Write a case study about setting up for AR development

TOTAL: 45 PERIODS

References:

1. Jonathan Lenowes, Augmented Reality with Unity AR Foundation,2021 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

Weightage:	Continuous Assessment: 40%	End Semester Theory Examinations: 60%
	(i) Activity: 10% (ii) Internal Theory Examination:	

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)
CO5	Evaluate mixed reality platforms, devices, and development frameworks to assess their suitability for various application domains	PO3 (2)	PSO2 (2)

CP25C22	**Blockchains Architecture and Design**	L	T	P	C
		3	0	0	3

Course Objective:

The objective of this course is to study the Blockchain Technology and its uses across a range of fields, with an emphasis on Blockchain fundamentals, Smart contracts, Consensus processes and Security issues.

UNIT I – BLOCKCHAIN FUNDAMENTALS AND CRYPTOGRAPHIC STRUCTURES 9

Introduction to Blockchain; layers and types of blockchain; generic elements; cryptographic primitives; hash functions; Merkle tree; structure of a block; linking blocks; mining; validating a new block; assembling and selecting chains.

Activities:

Quiz on hashing, signing data and building a Merkle tree.

UNIT II – BITCOIN AND CONSENSUS MECHANISMS 9

Bitcoin overview; Bitcoin scripts; transactions; network and payments; consensus mechanisms; types of consensus mechanisms; Proof of Work; consensus protocols for permissionless blockchain environments; consensus protocols for permissioned blockchain environments.

Activities:

1. Debate and discussion on the advantages and limitations of consensus protocols.
2. Identify and set up an Ethereum testnet and transfer ethers between

UNIT III – SMART CONTRACTS AND ETHEREUM 9

Smart contracts; definition and Ricardian contracts; introduction to Ethereum; Ethereum network; Ethereum components; Ethereum Virtual Machine (EVM); Ethereum development environment; Solidity language.

Activities:

1. Prepare a presentation on smart contracts and Ethereum concepts using

UNIT IV – WEB3 AND HYPERLEDGER FABRIC 9

Exploring Web3 with Geth; contract deployment; interacting with contracts through frontends; Truffle; introduction to Hyperledger; reference architecture; Hyperledger Fabric; Hyperledger Sawtooth; Corda.

Activities:

1. Explore and discuss research papers related to Hyperledger and blockchain technology.

UNIT V – BLOCKCHAIN APPLICATIONS, CHALLENGES AND FUTURE DIRECTIONS 9

Blockchain applications in government: border control, voting and citizen identification; healthcare; finance, insurance, post-trade settlement and financial crime prevention; scalability; privacy; emerging trends; limitations of current blockchain platforms; open research challenges.

Activities:

1. Research and present use cases such as supply chain and digital identity.

MRK Institute of Technology ★ 608 301. ★ Kattumannarkoil.

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

TOTAL: 45 PERIODS

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

References:

1. Imran Bashir, "Mastering Blockchain: A deep dive into distributed ledgers, consensus protocols, smart contracts, Dapps, cryptocurrencies, Ethereum,and more", Third Edition, Packt Publishing, 2020.

2. Andreas M. Antonopoulos, David A. Harding, "Mastering Bitcoin: Programming the open Blockchain", Third Edition, O'Reilly Publishing, 2023.

3. Ahmed Banafa, "Blockchain Technology and Applications", First Edition, River Publishers, 2024.

4. Melanie Swan, "Blockchain: Blueprint for New Economy", First Edition, O'Reilly Publishing, 2015.

5. Bikramaditya Singhal, Gautam Dhameja, Priyansu Sekhar Panda, "Beginning

6. Blockchain : A Beginner's Guide to Building Blockchain Solutions" , 1st edition, Apress, NewYork, 2018.

E-Resources:

1. Zero to Blockchain - An IBM Redbooks course, by Bob Dill, David Smits https://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/crse0401.html

2. Hyperledger Fabric - https://www.hyperledger.org/projects/fabric

3. https://www.tutorialspoint.com/blockchain/index.htm

4. https://www.nptel.ac.in/courses/106105184 by Prof.S.Chakraborty and P.Jayachandran,IIT Kharagpur

MRK Institute of Technology
608 301.
Kattumannarkoil.

CO–PO–PSO Mapping:

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, components, and operational principles of blockchain architectures and distributed ledger technologies.	--	--
CO2	Analyze blockchain consensus mechanisms, data structures, and network architectures to understand security, performance, and scalability issues	PO1 (3)	PSO1 (3)
CO3	Evaluate blockchain platforms, smart contract frameworks,and architectural designs to assess their suitability for various decentralized applications.	PO3 (2)	PSO2 (2)
CO4	Design blockchain-based systems by selecting appropriate architectures, consensus protocols, and smart contract models for real-world use cases.	PO2 (1)	PSO1 (3)
CO5	Design secure and scalable blockchain-based solutions by selecting appropriate architectures, consensus mechanisms, smart contracts and deployment	PO4 (2)	PSO1 (3), PSO2

CP25 C23	Human - Centered AI	L	T	P	C
		3	0	0	3

Course Objective:

The course aims to introduce the fundamentals of Human-Centered Artificial Intelligence (HCAI), focusing on its principles, frameworks, and societal impact. This course includes fairness, privacy, reliability, and ethical considerations in the development of intelligent systems. Additionally, it explores governance structures, best practices, and regulatory measures to ensure safe and trustworthy AI deployment.

UNIT I – INTRODUCTION TO HUMAN-CENTERED AI 9

Overview of AI development; defining Human-Centered AI (HCAI); difference between traditional AI and HCAI; design thinking process; cognitive models; human-AI collaboration; human factors; HCAI principles; impact of AI on society; importance of human values.

Activities:

1. Explore key innovations and challenges in human-AI collaboration across sectors.

2. Explore the impact of AI on ethics and human-centric approaches.

UNIT II – FAIRNESS, BIAS AND PRIVACY 9

Sources of bias; group fairness; exploratory data analysis and dataset limitations; individual fairness; counterfactual fairness; attack models; privacy-preserving learning; differential privacy; federated learning.
Activities:
1. Use AI platforms to detect and analyze different types of bias.
2. Explore the use of randomized response/coin-flip mechanisms in sensitive surveys for privacy.

UNIT III – HCAI FRAMEWORKS, SAFETY AND EXPLAINABILITY 9

Levels of automation; reliable, safe and trustworthy systems; two-dimensional HCAI framework; life-critical systems; golden rules for design; application examples; model explainability using LIME and SHAP.
Activities:
1. Prepare a presentation on Human-Centered AI and model explainability using LIME and SHAP

UNIT IV – HUMAN-CENTERED DESIGN AND INTELLIGENT SYSTEMS 9

Science and innovation goals; intelligent agents; tele-bots; assured autonomy and control centers; social robots and active appliances; enhancing human capabilities; human-centered design metaphors.
Activities:
1. Survey and analyze recent research papers emphasizing Human-Centered AI.

UNIT V – HCAI GOVERNANCE, ETHICS AND RESPONSIBLE DEPLOYMENT9

Ethical implications of emerging AI technologies; software engineering best practices; establishing safety cultures; independent oversight and certification; government regulation; national policies; ethical guidelines and legal frameworks for safe and responsible AI.
Activities:
1. Explore national policies, ethical guidelines and legal frameworks for responsible AI.

TOTAL: 45 PERIODS

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

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

MRK Institute of Technology ★ Kattumannarkoil. ★ 608 301.

References:

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

E-Resources:

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

CO-PO-PSO Mapping:

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, principles, and design philosophies of human-centered artificial intelligence for developing responsible and user-focused AI systems.	--	--
CO2	Analyze human-AI interaction models and ethical considerations to understand usability, transparency, and trust in intelligent systems.	PO1 (3)	PSO1 (3)
CO3	Evaluate human-centered AI approaches, frameworks, and applications to assess their effectiveness in enhancing user experience and societal impact.	PO3 (2)	PSO2 (2)
CO4	Design human-centered AI solutions by selecting appropriate interaction techniques, ethical guidelines, and evaluation strategies for real-world applications.	PO2 (1)	PSO1 (3)
CO5	Design trustworthy HCAI solutions by integrating human values, interaction techniques, fairness, privacy, explainability and governance requirements.	PO4 (2)	PSO1 (3), PSO2 (3)

CP25C24	Vibe Coding	L	T	P	C
		3	0	0	3

Course Objective:

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.

UNIT I – VIBE CODING AND HUMAN-AI COLLABORATION 9

Introduction to Vibe Coding; basic prompting techniques for coding tasks; roles and responsibilities in human-AI collaboration; AI assistant, prompt engineer and developer roles; simple AI-augmented workflows; interaction between AI-generated code and developer feedback.

Activities:

1. Role-play AI, prompt engineer and developer to simulate collaborative coding.
2. Learn prompting basics and participate in a class quiz and discussion.

UNIT II – AI-ASSISTED CODING TOOLS AND PROMPT DESIGN 9

Overview of AI-assisted coding tools such as Cursor, v0 and Bolt; features and use cases; setup and navigation of AI-coding platforms; building simple web applications with AI integration; workflow prototyping; prompts, context and flow; prompt optimization techniques.

Activities:

1. Explore AI tools and build a basic application.
2. Design a development workflow using AI tools and explain tool selection.
3. Present a complete AI workflow including prompt design and context handling.

UNIT III – OVERSIGHT, DEVELOPER ROLES AND ETHICAL CODING 9

Human oversight in AI coding systems; ethical and responsible AI usage; challenges and best practices in AI-augmented development; evolving developer roles; shift from coding to orchestration and strategy; AI integration across the software development lifecycle; human-AI collaboration in debugging, design and deployment.

Activities:

1. Discuss accountability, code quality and risks of over-reliance on AI.
2. Showcase how AI transforms developer roles across the SDLC.
3. Simulate a modern development team using AI tools.

UNIT IV – CODE QUALITY, RELIABILITY AND APPLIED AI PROJECTS 9

Long-term maintainability of AI-generated code; refactoring for clarity and standards; documentation strategies for mixed human-AI codebases; security and reliability issues; ethical coding practices; accountability; AI-powered task manager, resume builder, note-taking app, recipe generator, visual memory game and non-diagnostic health symptom checker.

Activities:

1. Refactor AI-generated code and evaluate quality and security.
2. Study ethical case studies.
3. Build and present a working AI-assisted prototype.

UNIT V – AI-DRIVEN WEB APPLICATION DEPLOYMENT 9

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:

1. Design and deploy a full AI-assisted web application as a team.
2. Demonstrate the solution and reflect on human-AI collaboration challenges.

TOTAL: 45 PERIODS

Weightage:	**Continuous Assessment: 40%** **(i) Activity: 10%** **(ii) Internal Theory Examination:**	**End Semester Theory Examinations: 60%**
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		

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	P	PS
CO1	Describe the fundamental concepts, philosophy, and practices of vibe coding and its role in modern,	--	--
CO2	**Analyze** vibe coding workflows, tools, and collaborative practices to understand productivity, creativity, and	P O1 (3)	PS O1 (3)
CO3	**Evaluate** vibe coding approaches, platforms, and community-driven practices to assess their	P O3 (2)	PS O2 (2)
CO4	**Design** software development solutions using vibe coding principles by selecting appropriate tools,	P O2 (1)	PS O1 (3)
CO5	Design and deploy AI-assisted software solutions by integrating suitable prompts, tools, workflows, testing, human oversight and accountability practices.	P O4 (2)	PS O1 (3),

CP25C25	Federated Learning	L	T	P	C
		3	0	0	3

Course Objective:

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

UNIT I – INTRODUCTION TO FEDERATED LEARNING 9

Definition and scope; motivation through data privacy, decentralization and compliance; centralized vs federated vs distributed learning; horizontal, vertical and transfer FL; applications in mobile keyboard, healthcare and IoT; FL system architecture; workflow and lifecycle; communication protocols.

Activities:

1. Quiz on Federated Learning basics.

UNIT II – FEDERATED LEARNING ALGORITHMS 9

Federated Averaging (FedAvg); Federated SGD (FedSGD); FedProx for system heterogeneity; personalized FL including pFedMe, FedPer and FedBN; non-IID data; aggregation and weight-update strategies; convergence issues; performance metrics.

Activities:

1. Flipped classroom on federated learning algorithms.

UNIT III – PRIVACY AND SECURITY IN FEDERATED LEARNING 9

Security threats; differential privacy and its application in FL; secure multiparty computation; overview of homomorphic encryption; secure aggregation; Byzantine robustness; model inversion attacks; data poisoning; mitigation strategies.

Activities:

1. Assignment on privacy and security in Federated Learning.

UNIT IV – SYSTEM AND COMMUNICATION CHALLENGES 9

Device heterogeneity; intermittent availability; memory, battery and network constraints; client selection; communication efficiency using quantization, compression and sparsification; dropout and partial client participation; FL over edge devices; energy-aware FL.

Activities:

1. Flipped classroom on communication challenges.

UNIT V – APPLICATIONS, FRAMEWORKS AND FUTURE DIRECTIONS 9

FL applications in healthcare, finance, smart cities and recommender systems; case studies; TensorFlow Federated, PySyft and Flower; FL simulators; project work; model deployment and evaluation; future directions including split learning, hybrid FL, federated analytics and FL + AI/ML.

Activities:

1. Assignment on applications of FL.
2. Build and evaluate a simple Federated Learning model.
3. Simple FL project.

TOTAL: 45 PERIODS

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.

Weightage:	**Continuous Assessment: 40%** **(i) Activity: 10%** **(ii) Internal Theory Examination:**	**End Semester Theory Examinations: 60%**
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)

MRK Institute of Technology ★ Kattumannarkoil. ★ 608 301.

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	**Analyze** federated learning algorithms and communication strategies to understand model convergence, privacy, and system performance.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** federated learning frameworks, security mechanisms, and deployment approaches to assess their effectiveness in distributed and sensitive data environments.	PO3 (2)	PSO2 (2)
CO4	**Design** federated learning solutions by selecting appropriate aggregation methods, communication protocols, and privacy techniques for real-world	PO2 (1)	PSO1 (3)
CO5	Design privacy-preserving federated learning solutions by integrating appropriate aggregation, communication, security and deployment strategies.	PO4 (2)	PSO1 (3), PSO2

CP25002	Deep Learning for Computer Vision	L	T	P	C
		3	0	0	3

Course Objective:

This course provides the automatic analysis and understanding of images and videos and Computer Vision. It occupies significant importance in applications including security, healthcare, entertainment, mobility, etc. The recent success of deep learning methods has revolutionized the field of computer vision, making new developments increasingly closer to deployment that benefits end users.

UNIT I – IMAGE FUNDAMENTALS AND VISUAL FEATURES 9

Image formation, capture and representation; linear filtering; correlation and convolution; edge, blob and corner detection; scale space and scale selection; SIFT, SURF, HoG and LBP.

Activities:

1. Seminar on image features.

UNIT II – VISUAL MATCHING AND DEEP LEARNING FOUNDATIONS 9

Bag-of-Words; VLAD; RANSAC; Hough transform; pyramid matching; optical flow; review of deep learning; multi-layer perceptron; backpropagation.

Activities:

1. Quiz on the basics of Deep Learning.

UNIT III – CONVOLUTIONAL NEURAL NETWORKS AND VISION APPLICATIONS 9

CNN fundamentals and evolution; AlexNet, ZFNet, VGG, InceptionNets, ResNets and DenseNets; kernel visualization; deconvolution; Deep Dream; hallucination; neural style transfer; CAM, Grad-CAM, Grad-CAM++ and recent methods including IG, Segment-IG and SmoothGrad; recognition and verification using Siamese, triplet, contrastive and ranking losses; object detection using R-CNN, Fast R-CNN, Faster R-CNN, YOLO, SSD and RetinaNet; segmentation using FCN, SegNet, U-Net and Mask-RCNN.

Activities:

1. Assignment on the principle behind hallucination.
2. Discussion differentiating R-CNN, Fast R-CNN and Faster R-CNN.

UNIT IV – VIDEO UNDERSTANDING, ATTENTION AND GENERATIVE MODELS 9

RNNs; CNN and RNN models for video understanding; spatio-temporal models; action/activity recognition; attention models in vision; image captioning; visual question answering; visual dialog; spatial transformers; Transformer networks; GANs, VAEs, PixelRNNs, NADE and normalizing flows; image editing, inpainting, superresolution, 3D object generation and security; CycleGANs, Progressive GANs, StackGANs and Pix2Pix.

Activities:

1. Presentation on Transformers.
2. Flipped classroom on applications of generative models.

UNIT V – RECENT TRENDS AND REAL-WORLD COMPUTER VISION 9

Zero-shot, one-shot and few-shot learning; self-supervised learning; reinforcement learning in vision; recent applications and deployment-oriented computer vision using deep learning.

Activities:

1. Design a simple project on a real-world computer 2. 2.vision application using a deep learning algorithm.

TOTAL: 45 PERIODS

Weightage :	**Continuous Assessment: 60%** **(i) Activity: 15%** **(ii) Internal Theory Examination: 35%** **(iii) Internal**	**End Semester Theory Examinations: 40%**
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		

References:

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, 2016, MIT Press
2. Michael Nielsen, Neural Networks and Deep Learning, 2016
3. Yoshua Bengio, Learning Deep Architectures for AI, 2009, Now Publishers
4. Richard Szeliski, Computer Vision: Algorithms and Applications, 2022, Springer Publisher.
5. Simon Prince, Computer Vision: Models, Learning, and Inference, 2012, Cambridge University Press
6. David Forsyth, Jean Ponce, Computer Vision: A Modern Approach, 2002, Pearson Education.

E-resources:

1. https://opencv.org/blog/deep-learning-with-computer-vision/
2. https://onlinecourses.nptel.ac.in/noc25_cs93/preview
3. https://cs231n.stanford.edu/

CO-PO Mapping

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and techniques of deep learning used for computer vision and image understanding tasks.	--	--
CO2	**Analyze** deep learning–based computer vision models to understand feature learning, performance metrics, and optimization challenges.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** computer vision algorithms and deep learning architectures to assess their effectiveness for various visual recognition and analysis applications.	PO3 (2)	PSO2 (2)
CO4	**Design** computer vision solutions by selecting and integrating suitable deep learning models, datasets, and training strategies for real-world vision problems.	PO2 (1)	PSO1 (3)
CO5	Design deep learning-based computer vision solutions by selecting suitable features, architectures, datasets, training strategies and evaluation methods for real-world problems.	PO4 (2)	PSO1 (3), PSO2 (3)

MRK Institute of Technology ★ Kattumannarkoil ★ 608 301

Dr. K. ANANDAVELU, B.E., M.E., Ph.D., D.I.S., I.G.C.O.H.S., M.I.S.T.E., M.S.E.A (T).,
PRINCIPAL
MRK Institute of Technology
Kattumannarkoil - 608 301.