VIDYASAGAR UNIVERSITY

Revised Proposed Syllabus

M.Tech. in Computer Science with Artificial Intelligence

Under the Department of Computer Science
From Session 2025-2026

MIDNAPORE-721102

M. Tech. in Computer Science with Artificial Intelligence								
Semester I								
S No.	**Paper Code**	**Paper Name**	**Marks**		**Class Hours**			**Credit**
			Int.	**Ext.**	**L**	**T**	**P**	**Cr. Pt.**
1	MTCST101	Mathematical Foundations	30	70	3	1	0	4
2	MTCST102	Data Science using Python	30	70	4	0	0	4
3	MTCST103	Artificial Intelligence	30	70	4	0	0	4
4	MTCST104	Advanced Algorithms	30	70	4	0	0	4
5	MTCST105 (Elective - I)	(a) Applied Software Engineering [MERN] / Devops (b) Edge Computing. (c) VLSI System Design (d) MOOCs/ NPTEL/ SWYAM/ Open Course	30	70	4	0	0	4
6	MTCST106	Artificial Intelligence Lab	00	25	0	0	3	1
7	MTCST107	Data Science and Algorithms Lab	00	25+25	0	0	6	2
8	MTCST108	Term Paper I	00	25	0	0	0	1
		Total	**150**	**450**	**19**	**1**	**6**	**24**
Semester II								
S No.	**Paper Code**	**Paper Name**	**Marks**		**Class Hours**			**Credit**
			Int.	**Ext.**	**L**	**T**	**P**	**Cr. Pt.**
1	MTCST201	High Performance Computing Architecture	30	70	4	0	0	4
2	MTCST202	Applied Soft Computing	30	70	4	0	0	4
3	MTCST203	Applied Machine Learning	30	70	3	1	0	4
4	MTCST204 (Elective - II)	(a) Green Computing for AI (b) Big Data Analysis (c) Project Management & Entrepreneurship (d) MOOCs/ NPTEL/ SWYAM/ Open Courses	30	70	4	0	0	4
5	MTCST205 (Elective - III)	(a) Image Processing and Pattern Recognition (b) Natural Language Processing (c) Recommender Systems (d) MOOCs/ NPTEL/ SWYAM/ Open Courses	30	70	4	0	0	4
6	MTCST291	HPC and Soft Computing Lab	00	25	0	0	3	2
7	MTCST292	Applied Machine Learning Lab	00	50	0	0	3	2
8	MTCST293	Term Paper II	00	25	0	0	0	1
		Total	**150**	**450**	**19**	**1**	**6**	**25**
Semester III								

S No.	Paper Code	Paper Name	Marks		Class Hours			Credit
			Int.	**Ext.**	**L**	**T**	**P**	**Cr. Pt.**
1	MTCST301	Cyber security	30	70	3	1	0	4
2	MTCST302	Deep Learning and Generative AI	30	70	4	0	0	4
3	MTCST303 (Elective - IV)	(a) Robotics and Path Planning (b) IoT Foundation (c) Quantum Computing (d) MOOCs/ NPTEL/ SWYAM/ Open Courses	30	70	4	0	0	4
4	MTCST391	Deep learning and Gen AI lab	00	50	0	0	3	4
5	MTCST392	Project Work (Part-I)	00	150	0	0	9	6
		Total	**90**	**410**	**11**	**01**	**12**	**22**
		Semester IV						
S No.	**Paper Code**	**Paper Name**	**Marks**		**Class Hours**			**Credit**
			Int.	**Ext.**	**L**	**T**	**P**	**Cr. Pt.**
1	MTCST491	Project Work (Part-II)	00	300	0	0	20	10
2	MTCST492	Viva-Voce with Term Paper -III	00	100	0	0	0	5
		Total	**00**	**400**	**0**	**0**	**20**	**15**
			390	**1710**				**86**

Program Outcomes (POs)

PO1: Advanced Knowledge: Apply advanced knowledge of computer science, mathematics, and artificial intelligence to solve complex problems in real-world environments.

PO2: Problem Solving: Identify, analyze, and formulate problems in AI-related domains and develop appropriate computational solutions.

PO3: Research Skills: Conduct independent research and contribute to the advancement of knowledge in artificial intelligence and related fields.

PO4: Tool Proficiency: Use state-of-the-art tools, technologies, and frameworks for AI and machine learning to develop intelligent systems.

PO5: Data Analysis: Collect, analyze, and interpret large-scale data using AI techniques such as deep learning, data mining, and natural language processing.

PO6: Design and Development: Design and implement AI models and systems that are scalable, efficient, and meet ethical and societal requirements.

PO7: Ethics and Responsibility: Understand the ethical, legal, and societal implications of AI technologies and adhere to professional responsibilities.

PO8: Communication Skills: Communicate complex AI concepts and solutions effectively through oral, written, and visual formats.

PO9: Teamwork and Leadership: Work effectively as an individual and as a member or leader of diverse teams in multidisciplinary settings.

PO10: Lifelong Learning: Recognize the need for lifelong learning and engage in continuous professional development in the fast-evolving field of AI.

Semester - I

MTCST101: Mathematical Foundations **[40L]**

Course Outcome (CO):

CO1: Apply discrete mathematical structures (sets, relations, functions, Boolean algebra) to model and solve computational problems.

CO2: Analyze propositional and predicate logic to evaluate arguments for validity, soundness, and equivalence in formal reasoning.

CO3: Examine formal languages, automata theory, and computability, including finite automata, Turing machines, and grammar hierarchies (Chomsky classification).

CO4: Classify computational problems based on complexity theory (P, NP, NP-complete, PSPACE) and apply reduction techniques to assess problem tractability.

Detailed Syllabus:

Module I: Discrete Structures

Sets, Relations and Functions; Proof Techniques, Algebraic Structures, Morphisms, Posets, Lattices and Boolean algebra.

Module II: Logic

Propositional calculus and Predicate Calculus, Satisfiability and validity, Notions of soundness and completeness

Module III: Languages & Automata Theory

Chomsky Hierarchy of Grammars and the corresponding acceptors, Turing Machines, Recursive and Recursively Enumerable Languages; Operations on Languages, closures with respect to the operations.

Module IV: Computability

Church-Turing Thesis, Decision Problems, Decidability and Undesirability, Halting Problem of Turing Machines; Problem reduction (Turing and mapping reduction).

Module V: Computational Complexity

Time Complexity -- Measuring Complexity, The class P, The class NP, NP-Completeness, Reduction, co-NP, Polynomial Hierarchy. Space Complexity -- Savich's Theorem, The class PSPACE.

REFERENCES BOOKS

1. J.P. Trembley and R. Manohar -- Discrete Mathematical Structures with Applications to Computer Science, McGraw Hill Book Co.,

2. Michael Sipser -- Introduction to The Theory of Computation, Thomson Course Technology.

3. John E. Hopcroft and J.D.Ullman -- Inrtroduction to Automata Theory, Languages and Computation, Narosa Pub. House, N. Delhi.

4. H.R. Lewis and C.H.Papadimitrou -- Elements of the Theory of Computation, Prentice Hall, International, Inc.

MTCST102: Data Science using Python [40L]

Course Outcome (CO):

CO1: Understand the fundamental concepts of Data Science, including data types, data structures, and the data science lifecycle, using Python libraries such as NumPy, Pandas, and Matplotlib.

CO2: Apply data preprocessing techniques to clean, transform, and normalize raw datasets for effective analysis using Python's Pandas, Scikit-learn, and SciPy.

CO3: Implement machine learning algorithms (supervised and unsupervised learning) using Python's Scikit-learn to build predictive models and evaluate their performance.

CO4: Visualize and interpret data insights using Python libraries like Matplotlib, Seaborn, and Plotly to create meaningful charts, graphs, and dashboards for decision-making.

Details Syllabus:

Module I: Mathematical Foundations for Data Science

Linear algebra, probability, and statistics.

Module II: Distribution

Continuous and Discrete Random Variables, Distribution Function of a Random Variable, Probability Mass Functions and Probability Density Functions, Characteristic Functions, Central Limit Theorems, Probability Distributions, Continuous Probability Distributions, Binomial, Poisson, Normal distribution

Module III: Data Visualization

Techniques and tools for visualizing data.

Module IV: Big Data

Hadoop, MapReduce, and big data frameworks.

Module V: Business Analytics

Data-driven decision-making processes

Recommended Reference Books

1. "Data Science from Scratch: First Principles with Python" by Joel Grus.
2. "Python for Data Analysis" by Wes McKinney.
3. "Data Engineering: A Novel Approach to Data Design" by Brian Shive.
4. "Introducing Data Science" by Davy Cielen, Arno D. B. Meysman, and Mohamed Ali.

MTCST103: Artificial Intelligence [40L]

Credit: 4

Course Outcome:

CO1: Explain core AI concepts including intelligent agents, problem-solving approaches, and ethical considerations, Differentiate between various AI paradigms (symbolic, statistical, connectionist), Analyze the capabilities and limitations of AI systems

CO2: Implement fundamental search algorithms (BFS, DFS, A*) for AI problem solving, Apply adversarial search techniques (Minimax, Alpha-Beta Pruning) in game playing, Evaluate search strategies based on completeness, optimality and complexity.

CO3: Represent knowledge using formal logic (propositional and predicate), Implement rule-based systems using forward/backward chaining, Develop simple expert systems for domain-specific problem solving.

CO4: Apply basic machine learning algorithms (supervised and unsupervised), Implement simple neural network models for pattern recognition, Evaluate AI solutions based on performance metrics and real-world constraints.

Details Syllabus:

Module I

Introduction: AI problems, foundation of AI and history of AI intelligent agents: Agents and Environments, the concept of rationality, the nature of environments, structure of agents, problem solving agents, problem formulation.

Module II

Searching: Searching for solutions, uniformed search strategies – Breadth first search, depth first Search. Search with partial information (Heuristic search) Greedy best first search, A* search Game Playing: Adversial search, Games, minimax, algorithm, optimal decisions in multiplayer games, Alpha-Beta pruning, Evaluation functions, cutting of search.

Module III

Knowledge Representation**:** Using Predicate logic, representing facts in logic, functions and predicates, Conversion to clause form, Resolution in propositional logic, Resolution in predicate logic, Unification.

Representing Knowledge Using Rules**:** Procedural Versus Declarative knowledge, Logic Programming, Forward versus Backward Reasoning

Module IV

Learning: What is learning, Rote learning, Learning by Taking Advice, Learning in Problem-solving, Learning from example: induction, Explanation-based learning.

Connectionist Models:Hopfield Networks, Learning in Neural Networks, Applications of Neural Networks, Recurrent Networks. Connectionist AI and Symbolic AI.

Module V

Expert System:Representing and using Domain Knowledge, Reasoning with knowledge, Expert System Shells, Support for explanation examples, Knowledge acquisition-examples.

Reference Books

1. Artificial Intelligence – A Modern Approach.Second Edition, Stuart Russel, Peter Norvig, PHI/ Pearson Education.
2. Artificial Intelligence, Kevin Knight, Elaine Rich, B. Shivashankar Nair, 3rd Edition,2008
3. Artificial Neural Networks B. Yagna Narayana, PHI.
4. Artificial Intelligence, 2nd Edition, E.Rich and K.Knight (TMH).
5. Artificial Intelligence and Expert Systems – Patterson PHI.
6. Expert Systems: Principles and Programming- Fourth Edn, Giarrantana/ Riley, Thomson.
7. PROLOG Programming for Artificial Intelligence.Ivan Bratka- Third Edition – Pearson Education.
8. Neural Networks Simon Haykin PHI.
9. Artificial Intelligence, 3rd Edition, Patrick Henry Winston., Pearson Edition.

MTCST104: Advanced Algorithms **[40L]**

Credit: 4

Course Outcome:

CO1: Evaluate time/space complexity of algorithms using asymptotic analysis (Big-O, Ω, Θ), Apply advanced design paradigms (divide-and-conquer, dynamic programming, greedy methods)

CO2: Classify problems into complexity classes (P, NP, NP-Complete) using reductions, Design approximation algorithms (e.g., vertex cover, TSP) with provable bounds

CO3: Utilize augmented data structures (B-trees, Fibonacci heaps, skip lists), Solve graph problems (max-flow, min-cut, bipartite matching) with optimal algorithms.

CO4: Apply streaming algorithms for big data processing, Design online algorithms with competitive ratio analysis, Explore quantum-inspired classical algorithms (Grover-inspired search)

Details Syllabus:

Module I

Time and Space Complexity, Recurrence for divide and conquer and its solution, Methods for solving recurrences, Merge sort, Heap sort, Quick sort and Complexity analysis.(6L)

Module II

Dynamic Programming: Matrix-chain multiplication, All pair shortest paths, Single source shortest path, Travelling salesman problem, 0-1 knapsack problem.(6L)

Module III

Greedy Method: Knapsack problem, Job sequencing with deadlines, Activity – selection, Huffman codes, Minimum spanning tree by Prim's and Kruskal's algorithms.(6L)

Module IV

N-queen‟s Problem: Constraint Satisfaction, Backtracking, Forward Checking, Look-ahead, Conflict directed backjumping

Module V

Set and String Problem: Set cover, String matching, Approximate string matching, longest common subsequence.(4L)

Module VI

Matching Problem: Stable Marriage, Hospital Resident Problem (2L) Amortized Analysis (2L)

Module VII

Network Flow (2L), Complexity Classes: P, NP, NP-Hard, NP-Completeness, SAT, 3-SAT, Graph Coloring, Hamiltonian Cycle, TSP(4L),Approximation Algorithms(2L),Randomized Algorithms(2L)

Reference Books:

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, “Introduction to Algorithms”.
2. Aho, J.Hopcroft and J.Ullman “The Design and Analysis of Algorithms”
3. D.E.Knuth “The Art of Computer Programming”, Vol. 3
4. Jon Kleiberg and Eva Tardos, "Algorithm Design"
5. E.Horowitz and Shani “Fundamentals of Computer Algorithms”
6. Rajeev Motwani and P. Raghavan, “Randomized Algorithms”. Cambridge University Press, New York.
7. Vazirani, Vijay V, “Approximation Algorithms”. Berlin: Springer.

Elective – I

MTCST105(A): Applied Software Engineering [40L]

Course Outcome:

CO1: Explain and evaluate modern software engineering principles, lifecycle models, and agile practices for full-stack application development.
CO2: Apply advanced concepts of MERN stack (MongoDB, Express.js, React, Node.js) for designing and building scalable web applications.
CO3: Integrate software engineering practices with DevOps methodologies including CI/CD, containerization, and cloud deployment.
CO4: Critically analyze and optimize the performance, security, and scalability of deployed software systems in real-world scenarios.

Details Syllabus:

Module I: Foundations of Applied Software Engineering [6L]

Software engineering principles in modern application development, Software development life cycle (SDLC) and Agile methodologies, Requirements engineering and system design modeling (UML, ER diagrams, API design), Version control systems (Git, GitHub/GitLab workflows)

Module II: Full-Stack Development with MERN [15L]

Introduction to JavaScript ES6+ and TypeScript in modern development, MongoDB: schema design, indexing, aggregation, transactions, Express.js: middleware, routing, REST API design, authentication/authorization, React.js: components, hooks, state management (Redux, Context API), Node.js: asynchronous programming, event-driven architecture, microservices, Integration of MERN stack components into scalable applications

Module III: Software Engineering Practices in DevOps [12L]

Principles of DevOps and Agile integration, Continuous Integration (CI) and Continuous Deployment (CD) pipelines, Containerization and orchestration: Docker, Kubernetes basics, Configuration management: Ansible, Terraform overview, Cloud deployment: AWS / Azure / GCP services for MERN applications, Monitoring and logging tools (Prometheus, Grafana, ELK Stack)

Module IV: Advanced Topics and Case Studies [7L]

Secure coding practices and application security (OWASP Top 10, JWT, OAuth2.0), Performance optimization and scaling strategies (caching, load balancing, clustering), Testing strategies: unit testing, integration testing, system testing (Jest, Mocha, Cypress), Software quality metrics and evaluation frameworks, Case studies: Full-stack + DevOps integration in industry applications

Reference Books:

1. *Software Engineering: A Practitioner's Approach*, Roger S. Pressman and Bruce R. Maxim, McGraw-Hill, 9th Edition, 2019.
2. *Designing Data-Intensive Applications*, Martin Kleppmann, O'Reilly Media, 2017.
3. *Learning React: Modern Patterns for Developing React Apps*, Alex Banks and Eve Porcello, O'Reilly Media, 2nd Edition, 2020.
4. *Node.js Design Patterns*, Mario Casciaro and Luciano Mammino, Packt Publishing, 3rd Edition, 2020.
5. *MongoDB: The Definitive Guide*, Shannon Bradshaw, Eoin Brazil, and Kristina Chodorow, O'Reilly Media, 3rd Edition, 2019.
6. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*, Gene Kim, Jez Humble, Patrick Debois, and John Willis, IT Revolution Press, 2nd Edition, 2021.
7. *Kubernetes Up & Running: Dive into the Future of Infrastructure*, Brendan Burns, Joe Beda, and Kelsey Hightower, O'Reilly Media, 3rd Edition, 2022.
8. *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*, Jez Humble and David Farley, Addison-Wesley, 2010.

MTCST105(B): Edge Computing **[40L]**

Course Outcome:

CO1: Design distributed edge architectures (cloud-edge-device hierarchy) for latency-sensitive applications, Evaluate trade-offs (computation offloading, energy efficiency, QoS) in edge network deployments

CO2: Optimize ML model inference (TinyML, model pruning) for resource-constrained edge devices, Implement stream processing frameworks (Apache Kafka, Flink) for edge analytics

CO3: Deploy zero-trust security models for edge device authentication and intrusion detection, Design blockchain-based solutions for secure edge data provenance and integrity

CO4: Investigate 5G/6G integration with MEC (Multi-access Edge Computing) for ultra-low-latency use cases, Prototype digital twin applications leveraging edge-to-cloud synchronization

Details Syllabus:

Module I:

Introduction to Edge Computing, Cloud, Fog, and Edge Computing, Limitations of centralized cloud, Use cases, Edge vs Fog vs Cloud

Module II:

Edge Architecture & Ecosystems**,** Components: Edge nodes, edge gateways, data aggregators, Ecosystem players: AWS Greengrass, Azure IoT Edge, Google Edge TPU, Edge-native computing requirements

Module III:

Protocols and Communication, Edge device communication: MQTT, CoAP, DDS. Data serialization, Local and edge-to-cloud communication, QoS, reliability, and energy-aware transmission

Module IV:

Virtualization at the Edge, Containers vs VMs in edge environments, Docker and Kubernetes (K3s, MicroK8s) for orchestration, Container registries and multi-device deployment, Resource constraints and optimization

Module V:

Edge Storage and Caching, Edge caching strategies: LRU, LFU, Distributed data storage at the edge, Edge file systems and data offloading, Edge-to-cloud synchronization and versioning

Module VI:

Security & Privacy at the Edge, Threat model in edge environments, Secure communication and data encryption, Trusted Execution Environments (TEE), Secure Boot, TPM, Privacy-preserving analytics: Federated Learning, Differential Privacy

Reference Books

1. Fog and Edge Computing: Principles and Paradigms – Rajkumar Buyya, Satish Narayana Srirama
2. Architecting the Cloud – Michael J. Kavis
3. Edge AI: Machine Learning for Embedded Systems – Xiaofei Wang
4. Research papers from **IEEE Transactions on Edge Computing**, **ACM SenSys**, and **NPTEL** lectures on Edge Computing

MTCST105(C): VLSI System Design **[40L]**

Course Outcome:

CO1: Understand the fundamentals of VLSI design flow, including CMOS fabrication, MOSFET operation, and design rules for layout optimization.

CO2: Design and analyze combinational and sequential digital circuits using CMOS logic families (Static, Dynamic, and Pass Transistor Logic).

CO3: Apply Verilog/VHDL for RTL design and verification, including simulation, synthesis, and testing of digital systems.

CO4: Evaluate performance metrics (power, delay, area) of VLSI circuits and apply optimization techniques for low-power and high-speed designs.

Details Syllabus:

Module I:

Introduction to VLSI design flow (Fabrication, CMOS, nMOS, Design Rules). Popular VLSI technologies (CMOS vs. nMOS vs. BiCMOS). Design styles (Full-Custom, Semi-Custom, ASIC, FPGA).

Module II:

Logic families: Static/Dynamic CMOS, DCVS (Differential Cascode Voltage Switch), PLAs (Programmable Logic Arrays). Pass Transistor Logic vs. Conventional Logic. Clocking strategies, Transit time effects. PLA minimization (Folding, SIMPLIFY, ESPRESSO algorithms). Scaling (Moore’s Law, Short-Channel Effects).

Module III:

Partitioning (Kernighan-Lin, Fiduccia-Mattheyses). Floorplanning & Placement (Simulated Annealing, Force-Directed). Routing (Global, Detailed, Channel Routing). Compaction techniques. Reconfigurable architectures (Gate Arrays, FPGAs, MCMs).

Module IV:

Data structures for layout design (MAGIC tool). Design Rule Checking (DRC), LVS (Layout vs. Schematic). Testability issues (Fault Models, ATPG, BIST). Expert systems in VLSI, Symbolic Layout. Complexity analysis of layout algorithms.

Reference Books:

1. C.Mead & L.Conway: Introduction to VLSI Systems, Addison Wesley.
2. A.Mukherjee: Introduction to CMOS VLSI, Prentice Hall.
3. Fabricius: Introduction to VLSI Design, TMH.
4. T.Ohtsuki: Layout Design and Verification, North Holland.
5. N.Sherwani: Algorithms for VLSI Physical Design Automation, Kluwer Academic.
6. M.Sarrafzadeh & C.K.Wong:An Introduction to VLSI Physical Design, MH.

MTCST105 (D): MOOCs/ NPTEL/ SWYAM/ Open Course

Students may allow taking any course from MOOCs/ NPTEL/ SWYAM which is not mentioned in this syllabus.

Practical

MTCST106: Artificial Intelligence Lab **[30L]**

Course Outcome:

CO1: Design and implement foundational AI algorithms (e.g., search, optimization, machine learning) using Python/R, and evaluate their performance on real-world datasets.

CO2: Develop intelligent agents for decision-making (e.g., game-playing bots, recommendation systems) by applying techniques such as reinforcement learning, knowledge representation, and probabilistic reasoning.

Core Lab Exercises

Search Algorithms

Implement uninformed search strategies: Breadth-First Search (BFS), Depth-First Search (DFS).

Implement informed search strategies: A*, Greedy Best-First Search.

Knowledge Representation and Reasoning

Develop Prolog programs for logical reasoning tasks.

Model knowledge using semantic networks and frames.

Natural Language Processing (NLP)

Perform text preprocessing: tokenization, stemming, lemmatization.

Develop simple chatbots using NLP techniques.

Expert Systems

Design rule-based expert systems using forward and backward chaining.

Implement inference engines for decision-making.

Game Playing

Implement the Minimax algorithm with alpha-beta pruning for two-player games.

Develop simple AI agents for games like Tic-Tac-Toe or Chess.

Neural Networks

Implement single-layer and multi-layer perceptron’s.

Train neural networks using backpropagation algorithm.

Reference Books

1. **Artificial Intelligence: A Modern Approach** by Stuart Russell and Peter Norvig – This book provides comprehensive coverage of AI concepts and is widely used in academia.
2. **Logic and Prolog Programming** by Saroj Kaushik – This book offers insights into logic

programming and Prolog, essential for knowledge representation and reasoning.

3. **Prolog Programming for Artificial Intelligence** by Ivan Bratko – A detailed guide to Prolog programming with AI applications.
4. **Pattern Recognition and Machine Learning** by Christopher M. Bishop – This book delves into machine learning techniques, useful for implementing algorithms in the lab.
5. **Deep Learning** by Ian Goodfellow, YoshuaBengio, and Aaron Courville – An in-depth resource on deep learning methodologies.

MTCST107: Advanced Data Science and Algorithms Lab [60L]

Credit: 2

Course Outcome:

CO1: Implement and evaluate advanced data structures (e.g., graphs, heaps, tries) and algorithms (e.g., Dijkstra's, Kruskal's) to solve computational problems, analyzing their time/space complexity empirically.

CO2: Design applications using algorithmic paradigms (greedy, divide-and-conquer, dynamic programming, backtracking) and compare their efficiency for real-world scenarios like scheduling, optimization, or resource allocation.

Details Lab Syllabus

Module I: Data Science Lab **[30L]**

Foundational Data Handling

Python Data Science Stack

NumPy: Array operations, broadcasting, statistical functions.

Pandas: DataFrames, handling missing data, groupby operations.

Data Preprocessing

Feature scaling (MinMax, StandardScaler), outlier detection (IQR, Z-score).

Text preprocessing (TF-IDF, tokenization) using Scikit-learn.

Machine Learning Implementation)

Supervised Learning

Linear regression, decision trees, SVM (using Sklearn).

Model evaluation (cross-validation, ROC curves).

Lab Task: Predict housing prices with regression + hyperparameter tuning.

Unsupervised Learning

Clustering (k-means, DBSCAN), dimensionality reduction (PCA).

Data Visualization & Insights

Exploratory Data Analysis (EDA)

Matplotlib/Seaborn: Histograms, boxplots, correlation matrices.

Plotly: Interactive dashboards.

Algorithmic Optimization for Data Science

Hashing for fast data retrieval (e.g., dictionary ADT).

Parallel sorting (QuickSort) for large datasets.

Module II: Algorithm Lab **[30L]**

Binary Search Tree (BST) Operations:

Insertion, Deletion, Search, and Display.

Binary Search Implementation:

Perform binary search on a set of integer values.

Splay Trees:

Implement splay tree operations.

AVL Trees:

Implement AVL tree operations.

Graph Traversal Algorithms:

Implement Breadth-First Search (BFS) and Depth-First Search (DFS).

Shortest Path Algorithms:

Implement Dijkstra's algorithm for single-source shortest path.

Minimum Spanning Tree Algorithms:

Implement Kruskal's and Prim's algorithms.

Sorting Algorithms:

Implement various sorting algorithms like Quick Sort, Merge Sort, Heap Sort, etc.

Pattern Matching Algorithms:

Implement Brute Force and Boyer-Moore pattern matching algorithms.

B-Trees:

Implement insertion and search operations in B-Trees.

Hashing Techniques:

Implement Dictionary Abstract Data Type (ADT) using hashing.

Dynamic Programming Applications:

Develop applications using dynamic programming techniques.

Backtracking Algorithms:

Implement applications for backtracking algorithms.

Reference Books

1. "Python for Data Analysis" – Wes McKinney (Pandas/NumPy)

2. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" – Aurélien Géron (ML implementation)
3. "Data Science from Scratch" – Joel Grus (algorithms + Python)
4. "Interactive Data Visualization with Plotly and Dash" – Elias Dabbas (Plotly/Seaborn)
5. "Data Structures and Algorithm Analysis in C++" by Mark Allen Weiss, 3rd Edition, Pearson Education, 2007.
6. "Data Structures, Algorithms and Applications in C++" by SartajSahni, 2nd Edition, Universities Press, 2007.
7. "Fundamentals of Algorithms" by Ellis Horowitz, SartajSahni, and Rajasekharan, 2nd Edition, Universities Press, 2009.
8. "Data Structures and Algorithms in Java" by Adam Drozdek, 3rd Edition, Cengage Learning, 2008.
9. "Fundamentals of Data Structures in C++" by Ellis Horowitz, SartajSahni, and Mehta, 2nd Edition, Universities Press, 2007.
10. "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, 4th Edition, MIT Press, 2022

MTCST108: Term Paper I

Seminar topic will be assigned to individual student by the Head of the department at the beginning of the semester.

Semester - II

MTCST201: High Performance Computing Architecture [40L]

Course Outcome:

CO1: Explain computer architecture fundamentals (von Neumann model, ISA design). Evaluate system performance using quantitative techniques (Amdahl's Law, CPI, benchmarking). Compare CISC and RISC processor designs with trade-offs (instruction complexity, power, pipelining).

CO2: Analyze instruction/arithmetic pipelining stages and their throughput. Identify pipeline hazards (data, control, structural) and apply mitigation techniques (forwarding, branch prediction, dynamic scheduling). Optimize pipelines using techniques like loop unrolling and speculative execution.

CO3: Design hierarchical memory systems (cache/virtual memory) and apply optimization techniques (prefetching, cache blocking). Explain instruction-level parallelism (ILP) via superscalar, VLIW, and vector processors. Evaluate advanced architectures (multicore, SIMD/MIMD) for parallel processing.

CO4: Compare shared-memory architectures (UMA/NUMA) and synchronization mechanisms (locks, barriers). Analyze consistency models (sequential, weak) and interconnection networks (crossbar, hypercube). Explore non–von Neumann architectures (dataflow, systolic arrays) and distributed systems (clusters, DSM).

Details Syllabus:

Module I:

Introduction: review of basic computer architecture, quantitative techniques in computer design, measuring and reporting performance. CISC and RISC processors.

Module II:

Pipelining: Basic concepts, instruction and arithmetic pipeline, data hazards, control hazards, and structural hazards, techniques for handling hazards. Exception handling. Pipeline optimization techniques. Compiler techniques for improving performance.

Module III:

Hierarchical memory technology: Inclusion, Coherence and locality properties; Cache memory organizations, Techniques for reducing cache misses; Virtual memory organization, mapping and management techniques, memory replacement policies. Instruction-level parallelism: basic concepts, techniques for increasing ILP, superscalar, superpipelined and VLIW processor architectures. Array and vector processors.

Module IV:

Multiprocessor architecture: taxonomy of parallel architectures. Centralized shared-memory architecture: synchronization, memory consistency, interconnection networks.

Module V:

Distributed shared-memory architecture. Cluster computers. Non von Neumann architectures: data flow computers, reduction computer architectures, systolic architectures.

References:

1. John L. Hennessy and David A. Patterson
   Title: Computer Architecture: A Quantitative Approach, Morgan Kaufmann
2. A foundational and widely used book covering advanced architecture, performance analysis, ILP, memory systems, and multiprocessors.
3. Kai Hwang
   Title: Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill

4. A comprehensive text focused on parallel processing, memory hierarchies, vector and multiprocessor systems.
5. David A. Patterson and John L. Hennessy
   Title: Computer Organization and Design: The Hardware/Software Interface, Morgan Kaufmann

MTCST 202: Applied Soft Computing

Credit: 4

Course Outcome:

CO1: Explain the fundamentals of soft computing, its components (neural networks, fuzzy logic, evolutionary algorithms), and how it differs from conventional AI and machine learning. Analyse the evolution of soft computing techniques and their applications in real-world problem-solving.

CO2: Design and implement artificial neural network models (Perceptron, Backpropagation, Self-Organizing Maps, Adaptive Resonance Theory) for classification, prediction, and clustering tasks. Evaluate the performance of neural networks on real-world datasets and compare their strengths/weaknesses.

CO3: Apply heuristic and meta-heuristic algorithms (Genetic Algorithm, Simulated Annealing, Tabu Search, Particle Swarm Optimization) to solve optimization and search problems. Compare the efficiency and convergence of these algorithms on benchmark problems (e.g., TSP, function optimization).

CO4: Perform fuzzy set operations and apply fuzzy reasoning for decision-making under uncertainty (e.g., control systems, risk assessment). Develop hybrid intelligent systems that integrate neural networks, fuzzy logic, and evolutionary algorithms for adaptive learning in complex applications (e.g., robotics, healthcare).

Details Syllabus:

Module I

Introduction to Soft Computing, Evolution of Computing, Soft Computing Constituents, From Conventional Artificial Intelligence to Computational Intelligence - Machine Learning Basics.

Module II

Neural Networks, Biological Neuron, Artificial Neuron, Artificial Neural Network, basic models, Hebb's learning, Adaline, Perceptron, Multilayer feed forward network, Back propagation, Different issues regarding convergence of Multilayer Perceptron, Competitive learning, Self-Organizing Feature Maps, Adaptive Resonance Theory, Associative Memories, Applications.

Module III

Heuristic and Meta-heuristic Search, Genetic Algorithm (GA), different operators of Genetic

Algorithm, Analysis of selection operations, Hypothesis of building Blocks, Schema theorem and convergence of Genetic Algorithm, Simulated annealing and Stochastic models, Boltzmann Machine, Tabu Search, Swarm Intelligence, Particle Swarm Optimization, Applications.

Module IV

Fuzzy sets and Fuzzy logic, Introduction, Fuzzy sets versus crisp sets, operations on fuzzy sets, Extension principle, Fuzzy relations and relation equations, Fuzzy numbers, Linguistic variables, Fuzzy logic, Linguistic hedges, Applications, Fuzzy Decision Making, Applications.

Module V

Hybrid Systems, Neural-Network-Based Fuzzy Systems, Fuzzy Logic-Based Neural Networks, Genetic Algorithm for Neural Network Design and Learning, Fuzzy Logic and Genetic Algorithm for Optimization, Applications.

Reference Books:

1. Mitchell Melanie, "An Introduction to Genetic Algorithm", Prentice Hall, 1998.
2. David E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", Addison Wesley, 1997.
3. S. Haykin, "Neural Networks", Pearson Education, 2ed, 2001.
4. S. Rajasekaran& G. A. V. Pai, Neural Networks, Fuzzy logic, and Genetic Algorithms, PHI.
5. Fuzzy Sets and Fuzzy Logic, Klir& Yuan, PHI, 1997.
6. Neural Networks, Fuzzy logic, and Genetic Algorithms, S. Rajasekaran and G. A. V. Pai, PHI.
7. Intelligent Hybrid Systems, D. Ruan, Kluwer Academic Publisher, 1997.

MTCST203: Applied Machine Learning **[40L]**

Course Outcome:

CO1: Explain the fundamentals and scope of machine learning, differentiating between supervised, unsupervised, and reinforcement learning paradigms. Apply data preprocessing techniques (normalization, feature engineering, handling missing data) to prepare datasets for analysis.

CO2: Implement supervised learning algorithms (Linear/Logistic Regression, Decision Trees, SVM, KNN) for classification and regression tasks. Evaluate model performance using metrics (accuracy, precision-recall, ROC curves, MSE) and techniques (cross-validation, hyperparameter tuning).

CO3: Design unsupervised learning solutions using clustering (K-Means, DBSCAN), dimensionality reduction (PCA, t-SNE), and association rule mining. Develop and train neural networks (MLPs, CNNs, RNNs/LSTMs) for pattern recognition, and explore generative models

(Autoencoders, GANs).

CO4: Explore advanced ML topics like reinforcement learning (Q-learning), NLP (Transformers), and ethical considerations (bias, fairness). Understand model deployment strategies (ML pipelines, APIs, edge computing) and real-world challenges (scalability, interpretability).

Details Syllabus:

Module 1:

Definition and scope of Machine Learning, Applications in various domains, Types of learning: Supervised, Unsupervised, Semi-supervised, Reinforcement, Basic concepts: Features, Labels, Training and Testing sets

Module II:

Linear Regression, Logistic Regression, Decision Trees, Random Forests, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Model evaluation metrics: Accuracy, Precision, Recall, F1-Score, ROC-AUC

Module III:

Clustering: K-Means, Hierarchical Clustering, DBSCAN, Dimensionality Reduction: Principal Component Analysis (PCA), t-SNE, Association Rule Learning: Apriori, Eclat

Module IV:

Perceptron, Multi-Layer Perceptron (MLP), Backpropagation algorithm, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), Autoencoders, Generative Adversarial Networks (GANs)

Module V:

Reinforcement Learning: Markov Decision Processes, Q-Learning, Natural Language Processing: Text preprocessing, Word Embeddings, Transformers, Model Deployment: Saving and loading models, REST APIs, Cloud deployment, Ethical considerations in Machine Learning

Recommended Reference Books

1. Machine Learning by Tom M. Mitchell
2. Pattern Recognition and Machine Learning by Christopher M. Bishop
3. Deep Learning by Ian Goodfellow, YoshuaBengio, and Aaron Courville
4. Introduction to Machine Learning by EthemAlpaydin
5. Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig
6. Data Mining: Concepts and Techniques by Jiawei Han, Micheline Kamber, and Jian Pei

Elective - II

MTCST204(A): Green Computing **[40L]**

Course Outcome:

CO1: Explain principles of green computing and evaluate environmental impacts of ICT systems.

CO2: Analyze energy-efficient hardware/software techniques and power management strategies.

CO3: Design sustainable IT infrastructure (data centers, cloud) and e-waste solutions.

CO4: Examine green IT policies, ethics, and emerging trends.

Details Syllabus:

Module I:

Definition and importance of Green Computing, Environmental impacts of Information and Communication Technology (ICT), Goals and principles of sustainable computing

Overview of energy consumption in computing systems

Module II:

Low-power hardware components and architectures, Energy-efficient algorithms and software practices, Power management techniques and tools, Role of virtualization in energy conservation

Module III:

Design and operation of energy-efficient data centers, Cooling technologies and thermal management, Green cloud computing models and practices, Metrics for assessing data center energy efficiency

Module IV:

Lifecycle assessment of computing devices, E-waste management and recycling strategies

Green procurement policies and standards, Case studies on sustainable IT implementations

Module V:

Government and industry initiatives for Green IT, International standards and certifications (e.g., Energy Star, EPEAT), Emerging trends in sustainable computing technologies, Ethical considerations and social responsibilities in Green Computing

Reference Books

1. Handbook of Energy-Aware and Green Computing – *Ishfaq Ahmad & Sanjay Ranka*
2. The Green Computing Book: Tackling Energy Efficiency at Large Scale – *Wu-chun Feng*
3. Green IT: Reduce Your Information System's Environmental Impact While Adding to the Bottom Line – *B. Dixon*
4. Green Computing: Tools and Techniques for Saving Energy, Money, and Resources – *Bud E. Smith*

MTCST 204(B): Big Data Analysis **[40L]**

Course Outcome:

CO1: Apply mathematical/statistical concepts (linear algebra, probability, optimization) and programming tools (Python/R, Pandas, NumPy) for data analysis and manipulation.

CO2: Design and implement machine learning/data mining techniques (clustering, classification, regression) and utilize big data technologies (Hadoop, Spark) for scalable processing.

CO3: Develop interactive visualizations (Tableau, Power BI) and deploy cloud-based big data solutions (AWS/Azure/GCP) using IaaS/PaaS models.

CO4: Solve domain-specific problems (NLP, IoT, time series) with advanced analytics (graph algorithms, deep learning) while addressing security/privacy challenges.

Details Syllabus:

Module I:

Linear Algebra: Matrices, Eigenvalues, Eigenvectors, Probability and Statistics: Distributions, Hypothesis Testing, Optimization Techniques: Gradient Descent, Convex Optimization

Module II:

Programming Languages: Python, R, Data Structures and Algorithms Data Manipulation and Analysis Libraries (e.g., Pandas, NumPy)

Module III:

Relational Databases and SQL, NoSQL Databases: MongoDB, Cassandra, Data Warehousing Concepts and ETL Processes

Module IV:

Supervised and Unsupervised Learning, Clustering, Classification, Regression Techniques
Association Rule Mining

Module V:

Hadoop Ecosystem: HDFS, MapReduce, Apache Spark: RDDs, DataFrames, Spark SQL, Real-time Data Processing with Spark Streaming

Module VI:

Visualization Tools: Tableau, Power BI, Designing Effective Dashboards, Storytelling with Data

Module VII:

Cloud Service Models: IaaS, PaaS, SaaS, Big Data Solutions on AWS, Azure, Google Cloud
Scalability and Resource Management

Reference Books

1. Hadoop: The Definitive Guide by Tom White
2. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data by EMC Education Services

3. Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph by David Loshin
4. Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses by Michael Minelli, Michele Chambers, and AmbigaDhiraj
5. Big Data Analytics with Spark: A Practitioner's Guide to Using Spark for Large Scale Data Analysis by Mohammed Gulle.

MTCST 204(C): Project Management and Entrepreneurship [40L]

Course Outcome:

CO1: Analyze modern project management frameworks (Agile, Waterfall, Hybrid) and design effective project plans with risk-mitigated schedules.

CO2: Develop advanced control systems to minimize scope creep using tools like Earned Value Management (EVM) and Monte Carlo simulations.

CO3: Evaluate entrepreneurial ventures through case-based learning, financial modeling, and business plan development for tech startups.

CO4: Synthesize strategies for sustainable innovation, intrapreneurship, and social entrepreneurship in engineering contexts.

Details Syllabus:

Module I:

Modern Frameworks: Agile (Scrum, Kanban), Waterfall, Hybrid models.Critical Chain Project Management (CCPM).

Project Planning & Control: Work Breakdown Structure (WBS), PERT/CPM, Gantt charts. Scope creep mitigation: Change control boards, baseline management.

Tools & Techniques: EVM, Monte Carlo simulations (using @Risk/Python). Lab: MS Project/Primavera for schedule optimization.

Module II:

Academic Foundations: Literature review: Journals (e.g., Journal of Business Venturing). Case studies: Tesla (scaling challenges), Zomato (pivot strategies).

Venture Creation: Lean Canvas vs. traditional business plans. Assignment: Pitch a deep-tech startup idea (AI/IoT/Blockchain).

Module III:

Funding & Growth: VC term sheets, crowdfunding (Kickstarter), ICOs for tech ventures. Case Study: Ola Electric's funding journey.

Specialized Domains: Family businesses (e.g., Tata Group succession planning). Social entrepreneurship (e.g., SELCO's solar solutions).

Corporate Innovation: Intrapreneurship: Google's "20% time" policy. Workshop: Design thinking for sustainable innovation**.**

Reference Books:

1. M. Y. Yoshino And U. S. Rangan, Strategic Alliances: An Entrepreneurial Approach To Globalization, Hbs Press, 1995.
2. Foster, Richard N., Innovation: The Attacker's Advantage, London, Macmillan, 1986.
3. Howard H. Stevenson, Michael J. Roberts, Amar Bhide, William A. Sahlman (Editor), The Entrepreneurial Venture (The Practice Of Management Series).
4. Udayan Gupta (Editor), Done Deals: Venture Capitalists Tell Their Stories.
5. Steve Kemper, Code Name Ginger: The Story Behind Segway And Dean Kamen's Quest To Invent A New World.
6. Paul A. Gompers And Josh Lerner, The Money Of Invention: How Venture Capital Creates NewWealth.
7. Larry Bossidy, Ram Charan And Charles Burck, Execution: The Discipline Of Getting Things Done.
8. JeffryTimmons And Stephen Spinelli, New Venture Creation: Entrepreneurship For The 21st Century With Powerweb And New Business Mentor Cd.
9. The Entrepreneur"s Guide To Business Law, Constance E. Bagley And Craig E. Dauchy, West Educational Publishing, 1998.
10. Mary Coulter, Entrepreneurship In Action, Prentice-Hall, 2001.

Elective – III

Code: MTCST205(A): Image Processing and Pattern Recognition [40L]

4L Credit: 4

Course Outcome:

CO1: Demonstrate image formation principles and perform spatial/frequency domain operations

CO2: Implement image restoration and segmentation techniques for object analysis

CO3: Apply feature extraction and statistical classification methods

CO4: Design clustering systems and advanced pattern recognition models

Details Syllabus:

Image Processing: [20L]

Module I:

Introduction, Background, Digital Image Representation, Fundamental steps in Image Processing, Elements of Digital Image Processing - Image Acquisition, Storage, Processing, Communication, Display.

Module II:

Digital Image Formation, A Simple Image Model, Geometric Model- Basic Transformation (Translation, Scaling, Rotation), Perspective Projection, Sampling & Quantization - Uniform & Non uniform.

Module III:

Mathematical Preliminaries, Neighbour of pixels, Connectivity, Relations, Equivalence & Transitive Closure; Distance Measures, Arithmetic/Logic Operations, Fourier Transformation, Properties of The Two Dimensional Fourier Transform, Discrete Fourier Transform, Discrete Cosine & Sine Transform.

Module IV:

Image Enhancement, Spatial Domain Method, Frequency Domain Method, Contrast Enhancement -Linear & Nonlinear Stretching, Histogram Processing; Smoothing - Image Averaging, Mean Filter, Low-pass Filtering; Image Sharpening. High-pass Filtering, High-boost Filtering, Derivative Filtering, Homomorphic Filtering; Enhancement in the frequency domain - Low pass filtering, High pass filtering.

Module V:

Image Restoration, Degradation Model, Discrete Formulation, Algebraic Approach to Restoration - Unconstrained & Constrained; Constrained Least Square Restoration, Restoration by Homomorphic Filtering, Geometric Transformation – Spatial Transformation, Gray Level Interpolation.

Module VI:

Image Segmentation, Point Detection, Line Detection, Edge detection, Combined detection, Edge Linking & Boundary Detection – Local Processing, Global Processing via The Hough Transform; Thresholding - Foundation, Simple Global Thresholding, Optimal Thresholding; Region Oriented Segmentation - Basic Formulation, Region Growing by Pixel Aggregation, Region Splitting & Merging.

Pattern Recognition: **[20L]**

Module I:

Basic concepts- Definitions, data sets for Pattern Recognition, Structure of a typical pattern recognition system.Different Paradigms of Pattern Recognition.Representations of Patterns and Classes. Metric and non- metric proximity measures

Module II:

Feature vectors - Feature spaces - Different approaches to Feature Selection-Branch and Bound

Schemes. Sequential Feature Selection.

Module III:

Principal Component Analysis (PCA), Kernel PCA

Module IV:

Pattern classification using Statistical classifiers - Bayes‟ classifier - Classification performance measures – Risk and error probabilities. Linear Discriminant Function, Mahalanobis Distance, K-NN Classifier, Fisher‟s LDA, Single Layer Perceptron, Multi-layer Perceptron, Training set, test set; standardization and normalization

Module V:

Basics of Clustering; similarity / dissimilarity measures; clustering criteria.Different distance functions and similarity measures. K-means algorithm, K-medoids, DBSCAN

Module VI:

Structural PR, SVMs, FCM, Soft-computing and Neuro-fuzzy techniques, and real life examples.

Reference Books:

1. Digital Image Processing, Gonzalves,Pearson
2. Digital Image Processing, Jahne, Springer India
3. Digital Image Processing &Analysis,Chanda&Majumder,PHI
4. Fundamentals of Digital Image Processing, Jain, PHI
5. Image Processing, Analysis & Machine Vision, Sonka, VIKAS
6. Devi V.S.; Murty, M.N. (2011) Pattern Recognition: An Introduction, Universities Press, Hyderabad.
7. R.O.Duda, P.E.Hart and D.G.Stork, Pattern Classification, John Wiley, 2001.
8. Statistical pattern Recognition; K. Fukunaga; Academic Press, 2000.
9. S.Theodoridis and K.Koutroumbas, Pattern Recognition, 4th Ed., Academic Press, 2009.

MTCST205(B): Natural Language Processing **[40L]**

Course Outcome:

CO1: Explain the linguistic foundations, scope, and challenges of NLP, including key applications (e.g., machine translation, sentiment analysis).

CO2: Implement text preprocessing pipelines (tokenization, POS tagging, parsing) and probabilistic language models (N-grams, HMMs) to analyze syntax and ambiguity.

CO3: Design semantic analysis systems (word sense disambiguation, coreference resolution) and evaluate NLP technologies (e.g., QA systems) for real-world use cases.

CO4: Develop deep learning-based NLP models (RNNs, Transformers) for contextual understanding and advanced tasks (e.g., text generation, named entity recognition).

Details Syllabus:

Module I:

Definition and scope of NLP, Applications in various domains, Challenges in NLP, Overview of linguistic essentials: syntax, semantics, pragmatics

Module II:

Tokenization, stemming, lemmatization, Part-of-speech tagging, Named Entity Recognition (NER), Parsing techniques: dependency and constituency parsing

Module III:

N-gram models, Hidden Markov Models (HMM), Probabilistic Context-Free Grammars (PCFG), Syntactic ambiguity and disambiguation techniques

Module IV:

Word sense disambiguation, Semantic role labeling, Discourse analysis, Coreference resolution

Module V:

Machine Translation, Text Summarization, Sentiment Analysis, Question Answering Systems

Module VI:

Word embeddings: Word2Vec, GloVe, Recurrent Neural Networks (RNNs), LSTMs, Transformers and BERT, Applications of deep learning in NLP tasks

Reference Books

1. Speech and Language Processing by Daniel Jurafsky and James H. Martin
2. Foundations of Statistical Natural Language Processing by Christopher D. Manning and HinrichSchütze
3. Natural Language Processing with Python by Steven Bird, Ewan Klein, and Edward Loper
4. Neural Network Methods in Natural Language Processing by Yoav Goldberg
5. Natural Language Processing in Action by Hobson Lane, Cole Howard, and Hannes Hapke

MTCST205(C): Recommender Systems [40]

Course Outcome:

CO1: Explain the foundational concepts, architectures, and real-world applications of recommender systems (e.g., e-commerce, streaming platforms).

CO2: Implement content-based (TF-IDF, cosine similarity) and collaborative filtering (SVD, ALS) techniques to generate recommendations.

CO3: Design advanced recommender systems using deep learning (Neural CF, GNNs), graph-based methods, and context-aware modeling.

CO4: Evaluate recommender systems using offline/online metrics (RMSE, NDCG), address challenges (cold-start, bias), and analyze ethical concerns (fairness, privacy).

Details Syllabus:

Module I:

Definitions, use-cases and industry applications (e.g., Netflix, Amazon, Spotify), Types of recommenders: Personalized vs Non-personalized, Implicit vs Explicit feedback, Overview of system architectures

Module II:

User profiles and item attributes, Similarity metrics: cosine, Euclidean, TF-IDF, Feature engineering for content-based recommenders

Module III:

Memory-based approaches: user-user and item-item collaborative filtering, Similarity measures and neighborhood selection, Model-based approaches: Matrix Factorization (SVD, ALS), Singular Value Decomposition

Module IV:

Deep Learning for Recommendations: Neural Collaborative Filtering (NCF), Autoencoders, RNNs for session-based recommendations, Graph-based Recommendations, Context-aware and Temporal Recommendations, Factorization Machines

Module V:

Combining content and collaborative methods, Linear blending, stacking, switching hybrid systems, Ensemble learning for recommenders

Module VI:

Offline metrics: Precision, Recall, F1, MAP, NDCG, AUC, Online evaluation: A/B testing, Cold-start problem and handling sparsity, Scalability using distributed systems (e.g., Apache Spark, Hadoop)

Module VII:

Filter bubbles, echo chambers, fairness, Differential privacy in recommendations, User data anonymization

Reference Books

1. Recommender Systems Handbook *by Francesco Ricci et al.*
2. Hands-On Recommendation Systems with Python *by Rounak Banik*
3. Deep Learning-Based Recommender System *by Waymond Rodgers (Springer)*

Practical

MTCST291: HPC and Soft Computing Lab [30]

Lab Objectives:

CO1: Design and implement fuzzy logic systems (membership functions, defuzzification) and evolutionary algorithms (GA, PSO) to solve optimization/control problems.

CO2: Develop neural networks (MLPs, CNNs) and hybrid models (ANFIS) using TensorFlow/ PyTorch for classification/regression, and apply multi-objective optimization (NSGA-II).

CO3: Analyze HPC architectures (parallelism, clusters) and implement parallel programs using MPI/OpenMP for scalable scientific computing.

CO4: Optimize HPC workflows (job scheduling, GPU/CUDA) and benchmark performance using profiling tools (e.g., gprof, NVIDIA Nsight).

Detailed lab exercises:

HPC Lab: **[15L]**

Module I:

Introduction to HPC concepts, architectures, and basic programming (C/Python).

Module II:

Introduction to parallel programming with MPI and OpenMP.

Module III:

Implementing parallel algorithms and optimizing code for performance.

Module IV:

Working with HPC clusters, job scheduling, and command-line tools.

Module V:

Benchmarking and performance analysis using various tools and techniques.

Module VI:

Advanced topics like GPU programming (CUDA) and case studies on real-world HPC applications.

Soft Computing Lab: **[15L]**

Module I:

Setup Python/MATLAB with libraries: NumPy, Scikit-fuzzy, TensorFlow, DEAP, PyGAD, Basic programming in Python for mathematical operations.

Module II:

Implementation of a Fuzzy Inference System (FIS), Case Study: Washing machine fuzzy controller / Temperature controller, Defuzzification techniques and membership function design

Module III:

Implement binary-coded GA for function optimization, Visualize selection, crossover, mutation operations, Solve Travelling Salesman Problem (TSP) using GA

Module IV:

Implement PSO from scratch, Solve benchmark optimization problems (e.g., Sphere, Rosenbrock functions)

Module V:

Path optimization in a graph using ACO, Case Study: Vehicle routing or shortest path problem

Module VI:

Implement Simulated Annealing and compare performance with GA, Optimization of nonlinear functions using DE

Module VII:

Implement feedforward neural network with backpropagation, Solve regression/classification problems using NN

Module VIII:

Combine Fuzzy Logic + Neural Networks (Adaptive Neuro-Fuzzy Inference System - ANFIS)

Implement using MATLAB Fuzzy Toolbox or equivalent Python library

MTCST292: Applied Machine Learning Lab **[30L]**

Course Outcome:

CO1: Implement and evaluate supervised learning algorithms (Linear/Logistic Regression, SVM, Decision Trees) and ensemble methods (Bagging, Boosting) for classification and regression tasks using performance metrics (MSE, F1-score, ROC curves).

CO2: Design and apply unsupervised learning techniques (k-Means, PCA, Hierarchical Clustering) for dimensionality reduction and pattern discovery, and validate models using cross-validation.

CO3: Develop neural networks (ANNs, CNNs, RNNs) using TensorFlow/PyTorch for structured and sequential data (images, time-series), and optimize them through feature engineering (normalization, RFE).

CO4: Build real-world ML systems (spam filters, recommender systems) by integrating preprocessing, model selection, and evaluation pipelines, and justify design choices.

Detailed lab exercises:

Module I:

Implementation of Linear Regression and evaluation using Mean Squared Error. Implementation of Logistic Regression for binary classification tasks. Construction and evaluation of Decision Trees and Random Forests. Implementation of Support Vector Machines (SVM) for classification problems. Application of k-Nearest Neighbors (k-NN) algorithm for classification tasks.

Module II:

Implementation of k-Means Clustering and analysis of cluster quality. Application of Hierarchical Clustering techniques. Implementation of Principal Component Analysis (PCA) for dimensionality reduction.

Module III:

Implementation of Bagging and Boosting techniques. Application of AdaBoost and Gradient Boosting algorithms.

Module IV:

Use of cross-validation techniques to assess model performance. Computation of evaluation metrics such as accuracy, precision, recall, and F1-score. Plotting and interpretation of ROC and Precision-Recall curves.

Module V:

Techniques for handling missing data and data normalization.

Implementation of feature selection methods like Recursive Feature Elimination.

Module VI:

Construction of simple Artificial Neural Networks using frameworks like TensorFlow or PyTorch.

Implementation of Convolutional Neural Networks (CNNs) for image classification tasks.

Application of Recurrent Neural Networks (RNNs) for sequence prediction problems.

Reference Books

1. Pattern Recognition and Machine Learning by Christopher M. Bishop
2. Machine Learning: A Probabilistic Perspective by Kevin P. Murphy
3. Deep Learning by Ian Goodfellow, YoshuaBengio, and Aaron Courville
4. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by AurélienGéron
5. Introduction to Machine Learning with Python by Andreas C. Müller and Sarah Guido

Semester 3

MTCST301: Cyber Security **[40L]**

Credit: 4

Course Outcome:

CO1: Explain the societal, economic, and ethical significance of cybersecurity, and analyze its impact on national infrastructure and career opportunities.

CO2: Design and evaluate internetworking solutions using IP (IPv4/IPv6), subnetting, routing protocols (RIP, OSPF, BGP), and mobile IP, addressing scalability and mobility challenges.

CO3: Implement cryptographic protocols (symmetric/public-key ciphers, PKI, TLS) and authentication systems (Diffie-Hellman, PGP, SSH) to secure network layers.

CO4: Assess security mechanisms (firewalls, IPsec, 802.11i) and vulnerabilities in real-world systems (DNS, HTTPS, QUIC), proposing mitigation strategies.

Detail Syllabus:

Module I:

Introduction: Significance and Scope of the Cyber security, Importance of Cyber security in societal, political and economic growth of the nation, Impact of the course on societal and ethical issues and career perspective.

Module II:

Internetworking I: Basic Internetworking (IP), What is an Internetwork?, Service Model, Global Addresses, Datagram Forwarding in IP, subnetting and classless addressing, Address Translation (ARP), Host Configuration (DHCP), Error Reporting (ICMP), Virtual Networks and Tunnels. Internetworking- II: Network as a Graph, Distance Vector (RIP), Link State (OSPF), Metrics, The Global Internet, Routing Areas, Routing among Autonomous systems (BGP), IP Version 6 (IPv6), Mobility and Mobile IP.

Module III:

Security Essentials:

Network Security: Internet Architecture, Network Protocols and Vulnerability, Application-Layer Security- Public Key Infrastructure, DNS Security Extensions, Hyper Text Transfer Protocol Secure (HTTPS), Network Time Protocol (NTP) Security, Transport-Layer Security-Handshake, Key-Derivation, Data-Transfer, Quick UDP Internet Connections (QUIC), Network Layer Security - IP Masquerading, IPv6 Security- Routing Protocol Security, Border Gateway Protocol (BGP) Security.

Module IV:

Cryptographic Building Blocks, Principles of Ciphers, Symmetric-Key Ciphers, Public-Key Ciphers, Authenticators, key Pre-distribution, Pre-distribution of Public Keys, Pre-distribution of Symmetric Keys, Authentication Protocols, Originality and Timeliness Techniques,

Module V:

Authentication and others: Public-Key Authentication Protocols, Symmetric-Key Authentication Protocols, Diffie-Hellman Key Agreement, Example Systems, Pretty Good Privacy (PGP),Secure Shell (SSH), Transport Layer Security (TLS, SSL, HTTPS),IP Security (IPsec), Wireless Security (802.11i),Firewalls, Strengths and Weaknesses of Firewalls.

MTCST302: Deep Learning & Generative AI [40L]

Course Outcome:

CO1: Explain the fundamentals of deep learning, including neural network architectures (MLPs, CNNs, RNNs), activation/loss functions, and training algorithms (backpropagation, gradient descent).

CO2: Design and implement convolutional neural networks (CNNs) for image/video tasks (e.g., classification, object detection) using architectures like ResNet and VGG.

CO3: Develop sequence models (RNNs, LSTMs, GRUs) for temporal data (e.g., text, time-series) and generative models (GANs, autoencoders) for data synthesis.

CO4: Apply advanced techniques (transfer learning, reinforcement learning) to real-world problems, and evaluate ethical challenges (bias, deployment risks) in deep learning systems.

Details Syllabus:

Module I:

Overview of machine learning and deep learning. Historical context and evolution of neural networks. Applications of deep learning in various fields.

Module II:

Perceptron and multilayer perceptron (MLP). Activation functions: sigmoid, tanh, ReLU, etc.

Loss functions and optimization techniques. Backpropagation algorithm and gradient descent variants.

Module III:

Convolution operations and pooling layers. Architectures like LeNet, AlexNet, VGG, and ResNet.

Applications in image and video processing.

Module IV:

Understanding sequences and temporal data. RNN architectures and challenges like vanishing gradients. Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs). Applications in language modeling and time series prediction.

Module V:

Autoencoders and their variants. Restricted Boltzmann Machines (RBMs).

Generative Adversarial Networks (GANs). Applications in data generation and representation learning.

Module VI:

Transfer learning and fine-tuning pre-trained models. Deep reinforcement learning basics.

Deployment of deep learning models in production. Ethical considerations and challenges in deep learning

Reference Books

1. Deep Learning by Ian Goodfellow, YoshuaBengio, and Aaron Courville
2. Pattern Recognition and Machine Learning by Christopher M. Bishop
3. Deep Learning with Python by François Chollet
4. Understanding Deep Learning by Simon J.D. Prince
5. Dive into Deep Learning by Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola

Elective - IV

MTCST303(A): Robotics and Path Planning **[40L]**

Course Outcome:

CO1: Analyze robot kinematics/dynamics (DH parameters, Jacobian, Lagrange) and compute motion trajectories for manipulators/mobile robots.

CO2: Design path planning algorithms (A*, RRT, potential fields) and implement collision avoidance in simulated environments.

CO3: Develop control strategies (PID, LQR, MPC) and trajectory optimization techniques (splines, time-parametrization).

CO4: Implement SLAM (EKF, graph-based) and AI-driven planning (RL, neural networks) for multi-robot systems.

Details Syllabus:

Module I:

History, applications, and classifications (humanoid, mobile). Robotic systems: Sensors (LiDAR, IMU), actuators, ROS basics.

Module II:

Forward/inverse kinematics (DH parameters). Jacobian, velocity control, Lagrange dynamics.

Module III:

C-space, grid-based (Dijkstra), sampling-based (RRT).

Module IV:

Path Planning A, D, potential fields, cost maps.

Module V:

PID, LQR, MPC, spline trajectories.

Module VI:

EKF-SLAM, particle filters, loop closure.

Module VII:

RL for planning, swarm robotics.

Reference Books

1. "Robotics: Modelling, Planning and Control" – Bruno Siciliano et al.
2. "Introduction to Robotics: Mechanics and Control" – John J. Craig
3. "Principles of Robot Motion: Theory, Algorithms, and Implementations" – Howie Choset et al.
4. "Simultaneous Localization and Mapping for Mobile Robots" – Juan-Antonio Fernández-Madrigal

MTCST303(B): IOT Foundation **[40L]**

Course Outcome:

CO1: Explain the architecture, components, and applications of IoT systems, including hardware platforms (e.g., Arduino, Raspberry Pi) and layered design (sensing, networking, cloud).

CO2: Interface sensors/actuators with IoT devices, analyze signal characteristics (noise, resolution), and process data for real-world sensing tasks.

CO3: Implement IoT communication protocols (MQTT, CoAP, LPWAN) and integrate cloud platforms (AWS IoT, ThingsBoard) for data aggregation and remote control.

CO4: Develop end-to-end IoT solutions by applying data preprocessing, ML techniques, and visualization tools to derive insights from sensor data.

Details Syllabus

Module I :

Definitions, applications, layers, and components. IoT hardware and computing platforms.

Module II :

Sensor types, parameters, interfacing, and basic signal processing.

Module III:

Communication networks, IoT protocols (WiFi, 5G, MQTT, LPWAN, etc.), and cloud computing.

Module IV:

Data preprocessing, statistics, machine learning techniques, and visualization.

Reference Books :

1. Arshdeep Bahga and Vijay Madisetti – *Internet of Things: A Hands-On Approach*
2. Adrian McEwen, Hakim Cassimally – *Designing the Internet of Things*
3. Rajkumar Buyya – *Fog and Edge Computing*
4. Jan Holler – *From Machine-to-Machine to the Internet of Things*

MTCST303(C): Quantum Computing [40L]

Course Outcome:

CO1: Explain quantum mechanics foundations (qubits, superposition, entanglement) and contrast classical vs. quantum computational models.

CO2: Design and implement quantum circuits using gates (Hadamard, CNOT, Toffoli) and simulate algorithms (Deutsch-Jozsa, Grover's) on quantum SDKs (Qiskit/Cirq).

CO3: Analyze quantum algorithms (Shor's, QFT, VQE) and evaluate their speedup over classical counterparts for cryptography, optimization, and chemistry.

CO4: Assess hardware challenges (decoherence, error correction) and emerging technologies (superconducting, photonic, trapped-ion qubits).

Details Syllabus:

Module I:

Quantum Foundations: Qubits, Bloch sphere, Dirac notation, Postulates of quantum mechanics Entanglement and Bell inequalities

Module 2:

Quantum Circuits & Gates: Universal gate sets (Pauli, Hadamard, CNOT), Quantum teleportation and superdense coding

Module 3:

Quantum Algorithms: Deutsch-Jozsa, Bernstein-Vazirani, Grover's search, Shor's factoring

Module 4:

Variational Quantum Eigensolver (VQE), Quantum Machine Learning (QML) basics

Lab: VQE for molecular ground states

Module 5:

Decoherence, T1/T2 times, Surface codes, fault-tolerant QC

Reference Books

1. "Quantum Computation and Quantum Information" – Michael Nielsen & Isaac Chuang
2. "Quantum Computing for Computer Scientists" – *Noson S. Yanofsky & Mirco A. Mannucci*

Practical

MTCST391: Deep learning and Gen AI lab **[30L]**

Course Outcome:

CO1: Design and implement neural network architectures such as MLPs, CNNs, and RNNs to solve classification, prediction, and sequence modeling problems using real-world datasets.

CO2: Evaluate deep learning models using appropriate metrics and optimize performance through techniques like regularization, hyperparameter tuning, and transfer learning.

CO3: Build and apply generative models such as GANs, VAEs, and transformer-based architectures for tasks involving image and text generation.

CO4: Demonstrate ethical awareness and practical understanding of deploying generative AI systems responsibly, including the use of prompt engineering and model explainability tools.

Core Lab Exercises:

Deep Learning Lab: **[15L]**

Module I:

Introduction to Deep Learning frameworks: PyTorch, TensorFlow, Tensor operations and model building basics, Implementing Perceptron and Multi-layer Perceptron (MLP), Activation functions (ReLU, Sigmoid, Tanh), Loss functions and optimizers (SGD, Adam, RMSprop)

Module II:

Basics of image processing and feature extraction, Building and training CNNs on MNIST and CIFAR-10, Concepts of pooling, padding, strides, and dropout, Visualizing filters and learned features, Transfer learning using pre-trained models (VGG, ResNet), Module III: Recurrent Neural Networks (RNNs), Sequential data and the concept of memory, Implementing RNN, GRU, and LSTM networks, Applications in time-series forecasting and text generation, Vanishing gradient problem and solutions

Module IV:

Simple and stacked autoencoders, Denoising autoencoders for image restoration, Variational Autoencoders (VAEs) – concept and implementation, Use cases in anomaly detection and dimensionality reduction

Module V:

Model evaluation metrics: Accuracy, Precision, Recall, F1-score, AUC, Cross-validation and train-validation-test splitting, Hyperparameter tuning: Grid search, Random search, Bayesian optimization, Use of TensorBoard and other visualization tools

Generative AI Lab: **[15L]**

Module I:

Overview of generative vs discriminative models, Introduction to Generative AI landscape: GANs, AEs, Diffusion Models, LLMs, Dataset preparation for generation tasks, Basic image/text generation pipeline setup

Module II:

Architecture and training of GANs, Implementing basic GAN for image generation, Deep Convolutional GANs (DCGANs), Common issues: mode collapse, convergence difficulties, Visualizing GAN outputs and latent space manipulation

Module III:

RNN-based vs Transformer-based generation, Attention mechanism and transformer block, Implementing text generation using GPT-like models (Hugging Face Transformers), Prompt design and prompt tuning

Module IV:

Understanding diffusion-based models (e.g., Stable Diffusion), Text-to-image generation using DALL·E or SDXL, Exploring multimodal generation: text + image inputs, Guided generation and conditioning

Reference Books

1. "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville
2. "Neural Networks and Deep Learning" by Michael Nielsen
3. "Deep Generative Models" by Jakub M. Tomczak and Max Welling
4. "GANs in Action: Deep learning with Generative Adversarial Networks" by Jakub Langr and Vladimir Bok
5. "Generative Deep Learning" by David Foster