

CENTRAL UNIVERSITY OF KASHMIR

Tulmulla, Ganderbal, Jammu and Kashmir, 191131



Masters in Technology (M. Tech) Artificial Intelligence & Machine Learning (AI&ML)

Course Syllabus
NEP-2020
[BoS – 16/04/2026]

School of Engineering & Technology
Department of Information Technology

Semester I

S.No.	Course Code	Course Title	Hours per Week			Total Credits	CIA	ESE
			L	T	P			
1.	TAA.61.506	Introduction to Artificial Intelligence	4	0	0	4	50	50
2.	TAA.61.507	Mathematical Foundations for AI & Machine Learning (Algebra, Prob. Stats, Multivariate calculus)	4	0	0	4	50	50
3.	TAA.61.508	Advanced Data Structure and Algorithm	3	0	2	4	50	50
4.	TAA.61.509	Computer Systems	3	0	2	4	50	50
5.	TAA.61.510	Software Lab	0	0	4	2	50	50
6.	TAA.61.511	Research Methodology, Ethics and IPR	2	0	0	2	50	50
		Total				20		

Programme:	M.Tech AI & ML	Semester:	I	
Course Code:	TAA.61.506	Course Title:	Introduction to Artificial Intelligence	
Scheme of Instruction			Scheme of Evaluation	
Total Duration	60 Hrs.		Max. Marks	100
Hours/Week	4 Hrs.		Assignment/IA	45
Credits	4		Attendance	05
Instruction Delivery	L:4, T:0, P:0		End Semester Exam	50
Course Basket	Major		Exam Duration	2 Hr.
Instruction Mode	Regular		Evaluation	Grade
Course Objectives	1. Understand the fundamental concepts of Artificial Intelligence. 2. Learn intelligent agents and problem-solving techniques. 3. Apply knowledge representation and reasoning methods. 4. Explore expert systems and AI applications. 5. Analyze ethical and societal implications of AI.			
Course Outcomes	CO1: Explain AI concepts and applications. CO2: Apply search and problem-solving techniques. CO3: Represent knowledge using logical models. CO4: Analyze AI systems and reasoning methods. CO5: Evaluate ethical issues in AI.			
Content				
Unit -1	Module 1	Basics of Artificial Intelligence with IKS Definition, Scope of Artificial Intelligence, AI vs Human Intelligence, Introduction to Indian Knowledge Systems (IKS), Nyaya Logic, Concept of Pramana (Means of Knowledge), relevance of IKS in Artificial Intelligence.		
	Module 2	Evolution of AI and Indian Contributions History and Evolution of Artificial Intelligence, Contributions from Indian Knowledge Systems: Panini’s Grammar (rule-based		

		system), Pingala's Binary System, symbolic reasoning in Indian philosophy, relevance to modern AI.
	Module 3	Types of Artificial Intelligence Narrow AI, General AI, Reactive Machines, Limited Memory Systems, Learning-based AI systems.
	Module 4	Applications of Artificial Intelligence Applications in Healthcare, Education, Business, Robotics, Natural Language Processing, limitations and challenges of AI systems.
	Module 5	Overview and Future of AI Future scope of AI, societal impact, advantages and disadvantage of AI.
Unit -2	Module 1	Intelligent Agents Definition of agents, types of agents, environment types, rational agents, agent architecture.
	Module 2	Problem Solving in AI Problem-solving agents, state space representation, formulation of problems.
	Module 3	Search Techniques Uninformed search: BFS, DFS, Depth Limited Search, Informed search: Heuristic search, A* Algorithm.
	Module 4	Advanced Search Methods Local search algorithms, hill climbing, adversarial search, game playing.
	Module 5	Constraint Satisfaction Problems (CSP) Definition, characteristics, applications of CSP in AI.
Unit -3	Module 1	Knowledge Representation Introduction to knowledge representation, importance and techniques.
	Module 2	Logical Representation Predicate logic, First-order logic, syntax and semantics.
	Module 3	Knowledge Structures Semantic networks, frames, scripts.
	Module 4	Rule-Based Systems

		Production systems, inference rules, forward and backward chaining.
	Module 5	Reasoning under Uncertainty Probability concepts, Bayesian networks, handling uncertainty in AI.
Unit -4	Module 1	Introduction to Expert Systems Definition, characteristics and applications of expert systems.
	Module 2	Components of Expert Systems Knowledge base, inference engine, user interface.
	Module 3	Knowledge Engineering Knowledge acquisition, representation techniques.
	Module 4	Reasoning Techniques Case-based reasoning, model-based reasoning.
	Module 5	AI Ethics and Social Impact Fairness, accountability, bias, privacy, governance, societal impact.
Note:	Each Unit is Equivalent to 1 Credit – 15 Hrs/ Week of Theory and 1 Credit consist of 5 Modules. Each Module is Equivalent to 3 Hrs. or Multiple of 3 Hrs.	
Examination	Part – A: Multiple Choice / Short Answer Type based on Level of Study Part – B: Short Type Questions. Part – C: Essay Type Questions. Note: Part – A & B Questions Covers all Units and Part – C from last 2 Units	
Text Book	[1] Stuart Russell & Peter Norvig, Artificial Intelligence: A Modern Approach [2] Elaine Rich & Kevin Knight, Artificial Intelligence [3] Dan W. Patterson, Introduction to AI and Expert Systems	
References	[1] Nils J. Nilsson, Artificial Intelligence: A New Synthesis [2] George F. Luger, Artificial Intelligence [3] Ivan Bratko, Prolog Programming for AI	

Programme:	M.Tech AI & ML	Semester:	I	
Course Code:	TAA.61.507	Course Title:	Mathematical Foundations For AI & ML	
Scheme of Instruction			Scheme of Evaluation	
Total Duration	60 Hrs.		Max. Marks	100
Hours/Week	4 Hrs.		Assignment/IA	45
Credits	4		Attendance	05
Instruction Delivery	L:4, T:0, P:0		End Semester Exam	50
Course Basket	Major		Exam Duration	2 Hr.
Instruction Mode	Regular		Evaluation	Grade
Course Objectives	1. Gain foundation in mathematical concepts related to Linear algebra necessary for ML. 2. Understands the concepts related to Orthogonality and rotations 3. Learns different forms of matrix decompositions. 4. Understands the concepts of differentiation for finding gradients and curvatures. 5. Learns various optimization methods.			
Course Outcomes	CO1: Attains the capability to solve linear equations and define mappings. CO2: Extract latent orthogonal features from the set of examples using PCA / SVD. CO3: Estimate the gradients of matrices to apply back propagation and other techniques for parameter update. CO4: Capable of estimating the summary statistics of datasets and select appropriate model to fit the data.			
Content				
Unit -1	Module 1	Ancient Indian Geometry (Śulba tradition – Baudhayana, Apastamba), Pingala’s Meru Prastara (Pascal-like triangle), Sanskrit Prosody (Chandas Shastra)		

	Module 2	Pingala's Binary System (Chandah Shastra), Vedic Geometry in altar construction, Indian Temple Architecture (Vastu Shastra). Āryabhata's astronomical computations, Bhāskara II (Leelavati & Bijaganita).
	Module 3	Norms & Inner Product Spaces Norms, Inner Products, Lengths and Distances , Inner Product of Functions.
	Module 4	Orthogonality Concepts Angles and Orthogonality, orthonormal Basis, orthogonal complement, orthogonal projections
	Module 5	Geometric Transformations Rotations
Unit -2	Module 1	Fundamental Matrix Properties Determinant and Trace
	Module 2	Eigen Concepts Eigenvalues and Eigenvectors
	Module 3	Matrix Decomposition Techniques–I Eigen Decomposition and Diagonalization, Cholesky Decomposition.
	Module 4	Matrix Decomposition Techniques–II Singular Value Decomposition, Matrix Approximation
	Module 5	Advanced Applications Matrix Phylogeny
Unit -3	Module 1	Basics of Differentiation Differentiation of Univariate Functions
	Module 2	Multivariable Differentiation Partial Differentiation and Gradients, Gradients of Vector-Valued Functions
	Module 3	Advanced Gradient Concepts Gradients of Matrices, Useful Identities for Computing Gradients
	Module 4	Computational Techniques Backpropagation and Automatic Differentiation
	Module 5	Higher-Order Analysis Higher-Order Derivatives, Linearization and Multivariate

		Taylor Series
Unit -4	Module 1	Foundations of Probability Construction of a Probability Space, Discrete and Continuous Probabilities
	Module 2	Probability Rules & Statistics Sum Rule, Product Rule, and Bayes' Theorem, Summary Statistics and Independence
	Module 3	Probability Distributions Gaussian Distribution, Conjugacy and the Exponential Family, Change of Variables / Inverse Transform
	Module 4	Optimization Techniques–I Optimization Using Gradient Descent
	Module 5	Optimization Techniques–II Constrained Optimization and Lagrange Multipliers , Convex Optimization
Note:	Each Unit is Equivalent to 1 Credit – 15 Hrs/ Week of Theory and 1 Credit consist of 5 Modules. Each Module is Equivalent to 3 Hrs. or Multiple of 3 Hrs.	
Examination	Part – A: Multiple Choice / Short Answer Type based on Level of Study Part – B: Short Type Questions. Part – C: Essay Type Questions. Note: Part – A & B Questions Covers all Units and Part – C from last 2 Units	
Text Book	[1] “Mathematics for Machine Learning”, Marc Peter Deisenroth, A. Aldo Faisal and Cheng Soon Ong, Cambridge University Press. [2] The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition, Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer 2017	
References	[1] Machine Learning: An Applied Mathematics Introduction, Paul Wilmott, John Wiley & Sons, 2011. [2] Machine Learning: A Probabilistic Perspective, John D. Lafferty, Cambridge University Press, 2012.	

Programme:	M.Tech AI & ML	Semester:	I	
Course Code:	TAA.61.508	Course Title:	Advanced Data Structure and Algorithm	
Scheme of Instruction			Scheme of Evaluation	
Total Duration	60 Hrs.		Max. Marks	100
Hours/Week	5 Hrs.		Theory Internal Assessment	25
			practical	20
Credits	4		Attendance	05
Instruction Delivery	L:3, T:0, P:2		End Semester Exam	50
Course Basket	Major		Exam Duration	2 Hr.
Instruction Mode	Regular		Evaluation	Grade
Course Objectives	1. The aim of this Course is to learn how to develop efficient algorithms for simple computational tasks and reasoning about the correctness of them. 2. To analyse the asymptotic performance of algorithms. 3. To write rigorous correctness proofs for algorithms. 4. Through the complexity measures, different range of behaviors of algorithms and the notion of tractable and intractable problems will be understood 5. Attains the capability to solve linear equations and define mappings.			
Course Outcomes	CO1: To apply knowledge of computing and mathematics to algorithm design. CO2: To analyze a problem and identify the computing requirements appropriate for its solution. CO3: To design, implement, and evaluate an algorithm to meet desired needs. CO4: To apply mathematical foundations, algorithmic principles, and computer science theory to the modeling and design of computer-based systems in a way that demonstrates comprehension of the trade-offs			

		involved in design choices. CO5: To apply design and development principles in the construction of software systems of varying complexity.
Content		
Unit -1	Module 1	Concept of algorithmic thinking in ancient India: Stepwise reasoning in Sulbasutras, combinatorics in Pingala's Chandaḥśāstra. Comparison of traditional Indian problem-solving approaches with modern algorithmic paradigms. Relevance of IKS to computational efficiency, optimization, and structured reasoning
	Module 2	Problem-solving frameworks: Pramana (validation), Tantrayukti (logical structuring of arguments) Case studies: Sulbasutra geometric constructions, Early recursive reasoning patterns
	Module 3	Introduction: Notion of Complexity covering time complexity and space complexity. Worst case complexity, Average case complexity
	Module 4	Big Oh , Big Theta and Big Omega notations with Example and Comparison. Examples of simple algorithms and illustration of their complexity. Amortized Cost. Iteration and Recursion- Problem solving using iteration and recursion.
	Module 5	Substitution Method (Forward and Backward), Tree Method, Master Theorem method.
Unit -2	Module 1	List Abstract Data Type: Operations, Use Cases. Array-Based Lists: Insertion, Deletion, Traversal. Pointer-Based Lists: Linked List Basics. Stack ADT & Applications: Push, Pop, Infix-Postfix.
	Module 2	Queue ADT & Applications: Enqueue, Dequeue, Circular Queue. Implementations (Array vs Linked List): Pros & Cons. Dictionary ADT, Open Hashing (Chaining) and Closed Hashing (Linear Probing) with collision Resolution techniques.
	Module 3	Skip Lists Structure, Binary Tree Basics: Common types and properties of binary trees, Tree Traversals (Pre/In/Post-order),
	Module 4	Counting of binary trees. Huffman coding using binary trees.

		Binary search trees: worst case analysis and average case analysis
	Module 5	AVL trees. Splay trees. Priority Queues. Binary heaps: insert and delete min operations and analysis. Binomial queues
Unit -3	Module 1	Directed Graphs- Data structures for graph representation. Shortest path algorithms: Dijkstra (greedy algorithm) and Bellman-Ford (dynamic programming).
	Module 2	Depth-first search and Breadth-first search, Directed acyclic graphs, Undirected Graphs
	Module 3	Minimal spanning trees and algorithms (Prim's and Kruskal) and implementation. All pair Shortest path (Floyd-Warshall), Travelling salesman problem.
	Module 4	Searching: Linear & Binary search, Hashing and Their Analysis. Sorting- Bubble sort, Selection sort, Insertion sort. Merge sort, Quicksort, Heap sort.
	Module 5	Shell sort, Radix sort, Analysis of the sorting methods. Selecting the top k elements. Lower bound on sorting. Complexity Classes: P class, NP-Complete, NP-Hard, P=NP problem.
Unit -4	Practical Component	1. Comparative study and implementation of algorithmic problem-solving approaches inspired by Sulbasutras and recursive reasoning patterns using flowcharts and pseudocode. 2. Implementation and analysis of iterative and recursive algorithms for factorial, Fibonacci series, GCD, and Tower of Hanoi problems. 3. Implementation of asymptotic analysis and empirical comparison of algorithms using Big O, Big Theta, and Big Omega notations. 4. Implementation of List ADT operations using arrays and linked lists including insertion, deletion, searching, and traversal. 5. Implementation of Stack and Queue ADTs along with applications such as infix-to-postfix conversion, expression evaluation, and circular queue operations. 6. Implementation of hashing techniques including chaining, linear probing, and collision resolution methods. 7. Implementation of binary trees, binary search trees, AVL trees, and heap operations including insertion, deletion, and traversals. 8. Implementation of graph traversal and shortest path

		algorithms including DFS, BFS, Dijkstra's, Bellman-Ford, Prim's, and Kruskal's algorithms. 9. Comparative implementation and performance analysis of searching and sorting algorithms including Binary Search, Merge Sort, Quick Sort, Heap Sort, and Radix Sort. 10. Mini project involving optimization and analysis of real-world computational problems using appropriate data structures and algorithms with complexity evaluation.
Note:	Each Theory Unit is equivalent to 1 Credit – 15 Hours of instruction, and each credit consists of 5 Modules. Each Module is equivalent to 3 Hours or a multiple thereof. The Practical Component (P:2) shall consist of 2 Hours of Laboratory/Hands-on Sessions per week.	
Examination	Part – A: Multiple Choice / Short Answer Type based on Level of Study Part – B: Short Type Questions. Part – C: Essay Type Questions. Note: Part – A and Part – B questions shall cover all Theory Units. Part – C questions shall primarily cover topics from Theory Units 2 and 3. The practical component shall be assessed through laboratory exercises, programming assignments, record work, mini-project implementation, presentation, and viva-voce.	
Text Book	[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, Clifford Stein, "Introduction to Algorithms", 5th Ed., PHI [2] Ellis Horowitz and SartazSahani, "Computer Algorithms", Galgotia Publications.	
References	[1] V. Aho, J. E. Hopcroft, J. D. Ullman, "The Design and Analysis of Computer Algorithms", Addison Wesley. [2] D. E. Knuth, "The Art of Computer Programming", 2nd Ed., Addison Wesley. [3] Jon Kleinberg and ÉvaTardos, "Algorithm Design", 1ST Edition, Pearson. [4] Mark Allen Weiss, "Data Structures and Algorithm Analysis in Java" Addison Wesley. [5] Robert Sedgewick and Kevin Wayne's, "Algorithms", Addison-Wesley Professional.	

Programme:	M.Tech AI & ML	Semester:	I	
Course Code:	TAA.61.509	Course Title:	Computer Systems	
Scheme of Instruction			Scheme of Evaluation	
Total Duration	60 Hrs.		Max. Marks	100
Hours/Week	5 Hrs.		Theory Internal Assessment	25
			Practical	20
			Attendance	05
Instruction Delivery	L:3, T:0, P:2		End Semester Exam	50
Course Basket	Major		Exam Duration	2 Hr.
Instruction Mode	Regular		Evaluation	Grade
Course Objectives	1. To understand fundamental concepts of information representation including number systems, binary arithmetic, and floating-point standards. 2. To develop an understanding of machine-level programming and x86-64 architecture, including control flow, procedures, and stack operations. 3. To analyze program performance and understand how modern processors, caches, and memory hierarchy impact efficiency. 4. To gain knowledge of operating system interfaces including processes, signals, virtual memory, and dynamic memory allocation.			
Course Outcomes	CO1: Explain and apply different number systems, data representations, and IEEE floating-point format for computational tasks. CO2: Analyze and interpret machine-level programs including control flow, procedures, stack operations, and memory access in x86-64 architecture. CO3: Evaluate program performance and optimize code using concepts of processor architecture, pipelining, and cache memory behavior. CO4: Understand and apply operating system concepts such as processes, signals, virtual memory, and dynamic memory allocation in system-level programming.			

		CO5: Design and develop basic client-server applications using Unix system calls, sockets, and concurrent programming techniques.
Content		
Unit -1	Module 1	Information Representation Binary and hexadecimal number systems; Byte ordering; Integer representation using Two's complement; Floating-point representation and IEEE 754 arithmetic.
	Module 2	Machine-Level Programming Introduction to x86-64 architecture; Registers and operands; Data movement instructions; Arithmetic and logical operations in assembly language.
	Module 3	Control Flow and Procedures Condition codes; Jump instructions; Loops and switch statements in assembly; Stack discipline; Procedure call and return mechanisms; Parameter passing; Local variables and recursion.
	Module 4	Arrays, Memory Layout, and Security Array allocation and access; Nested arrays; Buffer overflow vulnerabilities; Stack canaries; Address Space Layout Randomization (ASLR); Security defenses at machine level.
	Module 5	Program Optimization and Processor Architecture Compiler optimizations; Loop unrolling; Eliminating loop inefficiencies; Procedure call optimization; Superscalar processors; Pipeline hazards; Instruction-level parallelism; Branch prediction.
Unit -2	Module 1	Memory Hierarchy and Cache Organization Storage technologies – RAM, ROM, SSD, HDD; Temporal and spatial locality; Cache memory organization; Direct-mapped, set-associative, and fully associative caches.
	Module 2	Cache-Aware Programming Impact of cache misses on performance; Memory Mountain concept; Rearranging loops for locality; Matrix multiplication optimization techniques.
	Module 3	Exceptions and Process Management Hardware exceptions – interrupts, traps, faults, and aborts; Process context switching; System calls vs. function calls; Process creation and termination using fork, exec, and wait; Zombie processes and reaping.

	Module 4	Signals and Virtual Memory Signal handling terminology; Sending and handling signals; Safe signal handlers; Non-local jumps; Virtual memory concepts; Address translation; Page tables and Translation Lookaside Buffer (TLB).
	Module 5	Dynamic Memory Allocation and I/O Systems Implicit and explicit free lists; Allocation strategies – first fit and best fit; Garbage collection basics; Memory-related bugs; Unix I/O model; File descriptors; Standard I/O and Robust I/O (RIO).
Unit -3	Module 1	Network Programming Fundamentals Client-server communication model; IP addressing; Socket interface; TCP-based connection-oriented communication.
	Module 2	Concurrent Programming Models Concurrency vs. parallelism; Process-based concurrency; Event-driven programming using select and epoll mechanisms.
	Module 3	Thread-Based Concurrency Thread execution model; POSIX threads (Pthreads); Shared variables; Thread creation and management.
	Module 4	Synchronization Mechanisms Synchronization concepts; Mutexes; Semaphores; Critical sections and synchronization strategies in multithreaded programming.
	Module 5	Concurrency Pitfalls and Performance Race conditions; Deadlocks; Thread safety and reentrancy; Performance scaling on multi-core processors.
Unit -4	Practical Component	Practical Component (30 Hours) 1. Implementation of number system conversions and representation of signed integers and floating-point numbers using IEEE 754 format. 2. Writing and execution of basic x86-64 assembly language programs involving arithmetic, logical, and data movement instructions. 3. Implementation of control flow constructs in assembly language including loops, conditional branching, procedures, recursion, and stack operations. 4. Study and demonstration of memory organization, array handling, and buffer overflow vulnerabilities along with basic security defense mechanisms.

	5. Performance analysis and optimization of programs using compiler optimization flags, loop unrolling, and profiling techniques. 6. Simulation and analysis of cache memory organization including direct-mapped, set-associative, and fully associative cache models. 7. Implementation of process management system calls including fork(), exec(), wait(), signal handling, and inter-process communication in Unix/Linux environment. 8. Implementation of dynamic memory allocation techniques and analysis of memory-related bugs such as dangling pointers and memory leaks. 9. Development of client-server applications using socket programming and TCP communication. 10. Implementation of concurrent programming using POSIX threads (Pthreads), mutexes, semaphores, and synchronization mechanisms along with demonstration of race conditions and deadlocks.
Note:	Each Theory Unit is equivalent to 1 Credit – 15 Hours of instruction, and each credit consists of 5 Modules. Each Module is equivalent to 3 Hours or a multiple thereof. The Practical Component (P:2) shall consist of 2 Hours of Laboratory/Hands-on Sessions per week.
Examination	Part – A: Multiple Choice / Short Answer Type based on Level of Study Part – B: Short Type Questions. Part – C: Essay Type Questions. Note: Part – A and Part – B questions shall cover all Theory Units. Part – C questions shall primarily cover topics from Theory Units 2 and 3. The practical component shall be assessed through laboratory exercises, programming assignments, record work, mini-project implementation, presentation, and viva-voce.
Text Book	[1] R. E. Bryant and D. R. O'Hallaron, Computer Systems: A Programmer's Perspective, 3rd ed. Boston, MA, USA: Pearson, 2016.
References	[1] B. W. Kernighan and D. M. Ritchie, The C Programming Language, 2nd ed. Englewood Cliffs, NJ, USA: Prentice Hall, 1988 [2] W. R. Stevens and S. A. Rago, Advanced Programming in the UNIX Environment, 3rd ed. Upper Saddle River, NJ, USA: Addison-Wesley, 2013. [3] J. L. Hennessy and D. A. Patterson, Computer Architecture: A Quantitative

	Approach, 6th ed. Cambridge, MA, USA: Morgan Kaufmann, 2017. [4] W. Stallings, Computer Organization and Architecture: Designing for Performance, 11th ed. Upper Saddle River, NJ, USA: Pearson, 2022.
Online and Technical Resources	[1] Intel Corporation, Intel 64 and IA-32 Architectures Software Developer's Manual Volume 1: Basic Architecture, Order No. 253665, 2024. [2] The Open Group Base Specifications Issue 7 (IEEE Std 1003.1, 2017 Edition), POSIX.1-2017, 2017. [3] CS:APP3e Student Site, "Laboratory assignments." [Online]. Available: http://csapp.cs.cmu.edu/3e/labs.html. Accessed on: Apr. 5, 2026.

Programme:	M.Tech. AI & ML	Semester:	I	
Course Code:	TAA.61.510	Course Title:	Software Lab	
Scheme of Instruction			Scheme of Evaluation	
Total Duration	30 Hrs.		Max. Marks	100
Hours/Week	2 Hrs.		Assignment/IA	45
Credits	2		Attendance	05
Instruction Delivery	L:0, T:0, P:4		End Semester Exam	50
Course Basket	Major		Exam Duration	2 Hr.
Instruction Mode	Regular		Evaluation	Grade
Course Objectives	1. To introduce students to the fundamentals of data collection, cleaning, and pre-processing using tools such as Excel, Power Query, APIs, and web scraping techniques. 2. To develop practical skills in data visualization and dashboard creation using Power BI and Tableau for effective business intelligence reporting. 3. To provide hands-on experience in applying machine learning techniques such as classification, regression, and clustering using Orange and Python. 4. To enable students to analyse datasets, interpret results, and generate meaningful insights through interactive visualizations and predictive models.			
Course Outcomes	CO1: Collect, clean, pre-process, and manage datasets using Excel, Power Query, APIs, and web scraping tools. CO2: Create interactive dashboards and analytical reports using Microsoft Power BI and Tableau for effective data visualization. CO3: Apply machine learning workflows including classification, regression, clustering, and prediction using Orange and Python. CO4: Evaluate and compare machine learning model performance using suitable metrics and visualization techniques.			
Content				

PRACTICAL EXPERIMENTS	1. Data Collection and Cleaning using Excel and Power Query Mini Project Mobile Sales Data Cleaning Import dataset from publically available datasets. Remove duplicate records, Handle missing values, Apply filtering and sorting and Prepare cleaned dataset for analytics. 2. Creation of new data sets using API and Google Scrapping Import dataset, Remove duplicate records, Handle missing values, Apply filtering and sorting and Prepare cleaned dataset for analytics. 3. Dashboard Development using Power BI Mini Project Student Marks Dashboard Import dataset into Power BI and Create: Bar charts, Pie charts, KPI cards , Line graphs , Apply slicers and filters and Create interactive dashboard then Publish dashboard report. 4. Advanced Visualization using Tableau Mini Project Weather Data Visualization Connect Tableau to dataset, Create: Heat maps ,Geographic maps, Scatter plots, Storyboards .Design dashboard layout and Add calculated fields then Publish Tableau visualization. 5. Machine Learning Workflow using Orange/ Python Mini Project: Student Pass/Fail Prediction Import dataset into Orange/ Python, Perform data preprocessing, Apply classification algorithms/ Regression then Compare model performance and Visualize classification results. 6. Clustering and Prediction using Orange / Python Mini Project: Basic Customer Grouping Perform clustering using Orange, Apply K-Means clustering, Generate confusion matrix, Visualize clusters and Create prediction workflow. 7. Software Documentation and Collaboration Mini Project: Class Attendance Report Create project wiki, Maintain meeting notes, Prepare software documentation, Create project timeline and Generate software progress report. 8. Final Analytics and Project Presentation Mini Project: Simple Data Analytics Presentation Combine diagrams, dashboards, and reports, Prepare final project presentation, Demonstrate Power BI dashboard, Demonstrate Tableau visualization, Present Orange ML workflow.
Note:	The Practical Component (P:4) shall consist of 4 Hours of Laboratory/Hands-on Sessions per week
Examination	50 Marks of Internal Assessment (including attendance) and 50 Marks of ESE
Text Book	[1] An Introduction to Statistical Learning: with Applications in R by Gareth James, Daniela Witten Trevor Hastie and Robert Tibshirani [2] Data Analytics by Maheshwari, McGraw. [3] Data Mining Concepts and Techniques by Jiawei Han, Micheline Kamber

	and Jian Pei. Morgan Kaufmann
References	[1] S. H. Rowe and M. L. Schuh, Computer Networking. [2] Keizer, Local Area Networks, McGraw-Hill. [3] L. Peterson and B. Davie, Computer Networks: A Systems Approach, 5th ed., Elsevier, 2011. [4] A.S.Tanenbaum, Computer Networks, 4th ed., Pearson Education/PHI. [5] Cisco, "Introduction to Computer Networking," Cisco Networking Academy

Programme:	M.Tech. AI & ML	Semester:	I	
Course Code:	TAA.61.511	Course Title:	Research Methodology, Ethics & IPR	
Scheme of Instruction			Scheme of Evaluation	
Total Duration	30 Hrs.		Max. Marks	100
Hours/Week	2 Hrs.		Assignment/IA	45
Credits	2		Attendance	05
Instruction Delivery	L:2, T:0, P:0		End Semester Exam	50
Course Basket	Major		Exam Duration	2 Hr.
Instruction Mode	Regular		Evaluation	Grade
Course Objectives	5. To develop a comprehensive understanding of the fundamental concepts, objectives, and types of research across various disciplines. 6. To equip students with the skills necessary to identify, define, and design a feasible research problem using structured methodologies. 7. To introduce tools and techniques for conducting effective literature reviews and applying advanced research methods such as the Research Matrix 8. To foster ethical research practices and encourage collaboration through teamwork and multidisciplinary approaches. 9. To enhance scientific communication skills through effective report writing, proper referencing, research paper development, and oral presentations.			
Course Outcomes	CO1: Students will be able to explain the core principles and methodologies of research and differentiate among various research types. CO2: Students will be capable of formulating research problems and designing appropriate research strategies based on defined goals. CO3: Students will demonstrate the ability to conduct a thorough literature review and apply structured techniques for solid research execution. CO4: Students will show awareness of research ethics and the value of interdisciplinary collaboration in research environments.			

		CO5: Students will produce well-structured research reports and papers, properly cite sources, and confidently present their findings orally.
Content		
Unit -1	Module – 1	Introduction to Research Meaning, objectives, and significance of research, Types of research and their methodologies
	Module – 2	The Research Process Steps in the research process, Criteria of good research
	Module – 3	Defining the Research Problem Research problem and its components, Techniques for defining and refining a problem
	Module – 4	Designing the Research Introduction to research design, Goals of research
	Module – 5	Literature Review Literature review and solid research practices, The Research Matrix Method and Jim Gray Criteria, Conducting research, teamwork, and multidisciplinary research
Unit -2	Module – 1	Introduction to research ethics Ethics, Intellectual honesty & Research Integrity, Publication Misconduct, Plagiarism.
	Module – 2	Fundamentals of Scientific Writing Importance of report writing, Structure and components of a research report, Precautions in report writing, Use of pictures, graphs, and visual aids, Citation styles, Footnotes and bibliography
	Module – 3	Research Paper & Journals Characteristics of a great paper, Misconceptions in paper writing, The 10/30 test, Practical writing tips. Journal Indexes & Metrics
	Module – 4	Effective Oral Communication Preparation and delivery of oral presentations, Presenting research findings with impact
	Module – 5	Intellectual Property Rights Meaning, scope and importance. Types, Patents: definition,

	patentability criteria, filing and grant procedure.
Note:	Each Unit is Equivalent to 1 Credit – 15 Hrs/ Week of Theory and 1 Credit consist of 5 Modules. Each Module is Equivalent to 3 Hrs or Multiple of 3 Hrs
Examination	Part – A: Multiple Choice / Short Answer Type based on Level of Study Part – B: Short Questions of 100 – 200 Words Answer Type Part – C: Essay Type Question of 500 – 1000 Words Note: Part – A & B Questions Covers all Units and Part – C from last 2 Units
Text Book	[1] C. X. Ling and Q. Yang, <i>A Guide to Successful Master's and Ph.D. Degrees in Science & Engineering</i> . Morgan & Claypool, 2012. [2] C. R. Kothari, <i>Research Methodology: Methods & Techniques</i> , 2nd ed. New Delhi, India: Wishwa Prakashan, 2004.
References	[1] B. L. Garg, R. Karadia, and F. Aggarwal, <i>An Introduction to Research Methodology</i> . Jaipur, India: RBSA Publishers, 2002. [2] D. H. McBurney, <i>Research Methods</i> . Belmont, CA, USA: Thomson Learning, 2001. [3] D. R. Cooper and P. S. Schindler, <i>Business Research Methods</i> , 8th ed. New Delhi, India: Tata McGraw-Hill Co. Ltd., 2006. [4] R. P. Misra, <i>Research Methodology – A Handbook</i> . New Delhi, India: Concept Publishing Company, 1988. [5] K. Murlidhar, A. Ghosh, and A. K. Singhvi, <i>Ethics in Science Education, Research and Governance</i> . New Delhi, India: Indian National Science Academy, 2019.