RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

NEW-AGE GLOBAL UNIVERSITY
FOR LIBERAL EDUCATION

School of Computer Science and Engineering

Where Ideas Ignite Minds

Master of Technology (M.Tech.)
(2026 Scheme)

I and II Semester
Academic Year 2026-27

Syllabus Book

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS
RV Vidyaniketan, 8th Mile, Mysuru Road, Bengaluru, 560059, India
Ph : +91 80 68199900 | www.rvu.edu.in

School of Computer Science and Engineering (SoCSE)

Master of Technology
M.Tech

I and II Semester
(2026 Scheme)

Academic Year 2026-27

RV University (RVU)

RV Educational Institutions (RVEI), governed by Rashtreeya Sikshana Samithi Trust (RSST), is recognised among the few value–based and quality–oriented educational groups in the country. The Trust endeavours to impart quality education to all strata of society. RV University is a State Private University which has been established in Karnataka State with RSST as the sponsoring body through Act No.11 of 2019, passed by the Karnataka Legislature. The Missions of RSST and RV University is "Excellence in Education with Societal Commitment".

RVU Vision

To be a World–class, tech–driven, global university for liberal education, empowering citizens of tomorrow.

RVU Mission

RVU M1:
Strive for excellence in teaching, research, capacity building, and community engagement, benchmarking against global universities to lead across disciplines.

RVU M2:
Utilize digital and emerging technologies to enhance teaching-learning and research, accessible to all, while fostering a multidisciplinary, inclusive environment that meets evolving learner needs.

RVU M3:
Cultivate a diverse, global academic community through strong national and international collaborations that enrich learning, facilitate mobility, and drive institutional growth.

RVU M4:
Integrate theory with practical application to develop self-driven, empathetic problem - solvers equipped to create meaningful societal impact.

School of Computer Science and Engineering (SoCSE)

The School of Computer Science and Engineering (SoCSE) focuses on problem–solving, critical thinking, innovation, creativity, communication, entrepreneurship data science to deal with the VUCA (Volatility, Uncertainty, Complexity, and Ambiguity) world. All programmes of the school offer an opportunity to the students to work closely with all stakeholders – industries, government policymakers, researchers, think tanks and global organizations. School focuses on imparting 21st century skills through experiential, holistic learning. Curriculum supports interdisciplinary studies with minors from other schools and includes internships and student exchange programs with foreign universities.

SoCSE Vision

To be a pioneering school of Computer Science and Engineering committed to fostering liberal education and empowering the next generation of technologists to make a positive global socio-economic impact.

SoCSE Mission

SoCSE M1:
To be a pioneer in computer science education, fostering multidisciplinary research, innovation and entrepreneurship.

SoCSE M2:
To provide state-of-the-art facilities and dynamic curriculum that enables exemplary pedagogy, integrating advanced technology, theoretical foundations, and hands–on applications to address real-world challenges.

SoCSE M3:
To promote diverse communities of faculty and students through national and international academic, industry collaborations contributing to institution-building.

SoCSE M4:
To cultivate a generation of ethical, self-motivated, and empathetic problem solvers dedicated to achieving sustainable development goals.

About the Program – M.Tech

The M.Tech program provides students with a strong foundation in computer science and its real-world applications, preparing them for dynamic careers in the ever-evolving tech industry.The curriculum focuses on core and elective courses at RVU on par with industry requirements. The curriculum emphasizes practical learning through internships and project-based assignments, fostering hands-on expertise. Students are encouraged to engage in interdisciplinary R and D, consultancy projects, gaining practical experience with modern engineering tools aligned with their coursework. We follow a unique approach to teaching and learning focused on imparting 21st Century Skills to the students. It incorporates cutting-edge teaching methodologies such as flipped learning, mastery learning, and peer learning, all supported by advanced facilities. The program is led by distinguished faculty with extensive industry experience and academic credentials from premier institutions in India and abroad. Students also benefit from a vibrant campus life, enriched by dynamic student clubs and diverse extracurricular activities, making their academic journey truly memorable.

Program Outcomes (POs)

A graduate of the program will demonstrate:

PO1 — Application of Knowledge:
Apply knowledge from the core/interdisciplinary areas and develop skills to solve complex real-world problems.

PO2 — Critical Thinking & Creativity:
Analyze and evaluate the existing situations, practices and evidence to develop innovative multi-perspective solutions.

PO3 — Research & Innovation:
Cultivate a keen sense of observation and enquiry using appropriate methodology to develop new ideas, processes & solutions in response to contemporary challenges.

PO4 — Digital Transformation:
Instill an ability for confident, critical and responsible use of digital technologies for learning, work and participation in society.

PO5 — Communication & Interpersonal Skills:
Develop skills to effectively express thoughts and ideas both orally and in writing to people from all sections of the society.

PO6 — Leadership & Collaborative Teamwork:
Exhibit skills to set goals and guide people to align effectively in building a team to achieve the goals.

PO7 — Project Management & Entrepreneurial Mindset:
Foster an enterprising spirit by working on diverse projects, demonstrating time management, resource allocation and risk assessment capabilities to create value.

PO8 — Lifelong Learning:
Engage in continuous self-directed learning to adapt to evolving technologies, societal changes and emerging global challenges.

PO9 — Global Perspective & Cultural Awareness:
Acquire knowledge about the values and beliefs of diverse cultures, effectively engaging with empathy.

PO10 — Ethical Reasoning & Social Responsibility:
Demonstrate moral and value-based principles while being socially conscious and accountable.

PO11 — Civic Engagement & Community Service:
Participate actively in civic life exhibiting a commitment to community service, social justice and the common good.

PO12 — Environmental Sustainability:
Show responsibility to manage and work towards achieving the Sustainable Development Goals (SDGs).

Program Educational Objectives (PEOs) – M.Tech

PEO1: Graduates will be able to apply their knowledge and critical thinking to solve real- world problems, demonstrate leadership and ethics, and contribute to societal progress through lifelong learning and interdisciplinary collaboration.

PEO2: Graduates will drive technological innovation and entrepreneurship, leveraging multidisciplinary research to develop sustainable solutions that address societal challenges.

PEO3: Graduates will engage in global academic and industry collaborations, utilizing advanced computing technologies to lead transformative initiatives for organizational development.

Program Specific Outcomes (PSOs) – M.Tech

A graduate of the M.Tech will also demonstrate;

PSO1: Professional Readiness for Evolving Tech Ecosystems:
Demonstrate readiness for professional roles, higher studies, research, or entrepreneurship by continuously adapting to emerging technologies.

PSO2: Tech-Driven Community Engagement:
Engage with societal challenges to develop impactful, technology-driven solutions that serve and empower communities.

PSO3: Adaptive Technology Integration and Innovation:
Leverage advanced skills and cutting-edge technologies to develop innovative solutions while effectively adapting to the technological landscape.

Credit Distribution (2-Year PG — 81 Credits)

No.	Category	Credits
1	Core Courses – Major (Core)	34
2	Elective Courses – Major (Elective)	21
3	Internship – (IN)	4
4	Seminar – (SE)	2
5	Project – (PD)	20
	Total	**81**

M.Tech Course Matrix

Sem	Core	Elective	IN	SE	PD	Credits
SEM 1	12	6				**18**
SEM 2	10	9				**19**
SEM 3	12	6	4		2	**24**
SEM 4				2	18	**20**
Total	**34**	**21**	**4**	**2**	**20**	**81**

Semester 1

Type	Code	Subject	Cr.	L–T–P	Internal		Sem. End		Pass
					Max	Min	Max	Min	
Core	CS5800	Mathematics for Computer science	4	3-1-0	70	35	30	15	50
Core	CS5000	Advanced Data Structures and algorithms	4	3-0-2	70	35	30	15	50
Core	CS5200	Advanced Database Management system	4	3-0-2	70	35	30	15	50
Elective	CS5XXX	Elective – I/II	3	2-0-2	70	35	30	15	50
Elective	CS5XXX	Elective – I/II	3	2-0-2	70	35	30	15	50
Total			**18**						

Elective I courses

Type	Code	Subject	Cr.	L–T–P	Internal		Sem. End		Pass
					Max	Min	Max	Min	
Elective	CS5201	Advanced Machine learning	3	2-0-2	70	35	30	15	50
Elective	CS5202	Natural language processing	3	2-0-2	70	35	30	15	50

Elective II courses

Type	Code	Subject	Cr.	L–T–P	Internal		Sem. End		Pass
					Max	Min	Max	Min	
Elective	CS5203	ML for Data Science	3	2-0-2	70	35	30	15	50
Elective	CS5500	Principles of Data Analytics and Visualization	3	2-0-2	70	35	30	15	50

Semester 2

Type	Code	Subject	Cr.	L–T–P	Internal		Sem. End		Pass
					Max	Min	Max	Min	
Core	CS5100	High performance computer architecture	4	3-1-0	70	35	30	15	50
Core	CS5101	Advanced Operating Systems	4	3-0-2	70	35	30	15	50
Core	CS5900	Research Methodology	2	2-0-0	70	35	30	15	50
Elective	CS5XXX	Elective – I/II	3	2-0-2	70	35	30	15	50
Elective	CS5XXX	Elective – I/II	3	2-0-2	70	35	30	15	50
Elective	CS5XXX	Elective – I/II	3	2-0-2	70	35	30	15	50
Total			**19**						

Elective I courses

Type	Code	Subject	Cr.	L–T–P	Internal		Sem. End		Pass
					Max	Min	Max	Min	
Elective	CS5204	Deep learning	3	2-0-2	70	35	30	15	50
Elective	CS5205	Computer Vision	3	2-0-2	70	35	30	15	50
Elective	CS5206	MLOps	3	2-0-2	70	35	30	15	50

Elective II courses

Type	Code	Subject	Cr.	L–T–P	Internal		Sem. End		Pass
					Max	Min	Max	Min	
Elective	CS5207	Statistical Methods for Data Science	3	2-0-2	70	35	30	15	50
Elective	CS5501	Predictive and prescriptive analysis	3	2-0-2	70	35	30	15	50
Elective	CS5400	Data Privacy, Security and Ethics	3	2-0-2	70	35	30	15	50

L – Lecture, T – Tutorial, P – Practical / Project

Semester - **I**			
Course: **MATHEMATICS FOR COMPUTER SCIENCE**			
Program	**M.Tech CSE**	Category	**Major (Core)**
Course Code	**CS5800**	CIE Marks	**70**
Credits (L:T:P)	**4 (3:1:0)**	SEE Marks	**30**
Hours	**45L + 15T + 0P = 60**	SEE Mode	**Theory**

Course Objectives: students will be able to	
1	Discuss linear vector spaces and associated concepts like independence, dimensions, linear transformations and matrix representations.
2	Illustrate various orthogonality concepts and its application in projections and regression problems.
3	Discuss applications of matrix decomposition methods and spectral theory in data and image processing.
4	Describe probability theory and its applications in statistics and hypothesis testing using statistical methods.

Module - 1	**9 hours**
Vector spaces; sub-spaces; Linear independence; Basis vector sets; dimensions; coordinate vectors; Linear transformations, matrix representation of transformations.	

Module - 2	**9 hours**
Inner product, orthogonal sets, orthogonal projections, orthogonal bases; Gram-Schmidt orthogonalization process. QR factorization of matrices, least square problems, applications to linear models (least square lines and least square fitting of other curves).	

Module - 3	**9 hours**
Symmetric and Quadratic Forms; Diagonalization, Constrained Optimization (Lagrange multipliers), Singular Value Decomposition; Applications to image processing and statistics; Principal Component Analysis.	

Module - 4	**9 hours**
Random variable (discrete and continuous), Parametric families of distributions (binomial, Poisson, exponential, normal, Gamma), Probability mass function (pmf), Probability density function (pdf), Joint distribution, Bayes' Theorem, Mathematical expectation, conditional expectation.	

Module - 5	**9 hours**
Sampling theory: Random samples, sampling distributions of estimators, uni-variate and multivariate Central Limit Theorem, Probabilistic inequalities (Markov and Chebyshev's), Markov chains. Methods of Moments and Maximum Likelihood; testing of hypothesis by t-test, Chi square- test, p-value test.	

Course Outcomes: After completing the course, the students will be able to	
1	Analyse vector spaces and linear transformations by examining sub-spaces, basis sets, coordinate vectors, and their matrix representations.
2	Analyse inner product spaces and orthogonal structures to solve least squares problems using projections, orthogonalization, and QR factorization.
3	Analyse symmetric matrices and quadratic forms to perform diagonalization, optimization, and data reduction using SVD and PCA in applied contexts.
4	Evaluate probabilistic models and inferential methods by examining distributions, expectations, sampling behaviour, and hypothesis testing to draw conclusions from data.

Text Books	
1	Sheldon Ross, A First Course on Probability, Pearson Education, 10th ed (2022), ISBN-13: 978-9356064034
2	Gilbert Strang, Introduction to Linear Algebra, Wellesley-Cambridge, 6th ed (2023), ISBN-13: 978-1733146678
3	Erwin Kreyzig, Advanced Engineering Mathematics, Wiley, 10th Ed, (2023), ISBN 978-0-470-45836-5

Reference Books and Resources	
1	Hoel, Port, Stone, Introduction to Probability Theory, Houghtion Mifflin,1st ed(1972), ISBN-13978-0395046364
2	David Poole, Linear Algebra-A Modern Introduction, Cengage Learning India, 4th ed(2014), ISBN-13978-8131530245.
3	John Vince, Foundation Mathematics for Computer Science, Springer, 1st ed(2015), ISBN-13978-3319214368
4	K. Trivedi, Probability and Statistics with Reliability, Queuing, and Computer Science Applications, 2ed (2001), John Wiley & Sons Inc. ISBN-13: 978-0471333418
5	M. Mitzenmacher and E. Upfal, Probability and Computing: Randomized Algorithms and Probabilistic Analysis. 2 ed. (2017) Cambridge University Press.
6	Alan Tucker, Applied Combinatorics, Wiley, 6Ed, (2012), ISBN-13: 978-0470458389

Tutorials	**15 Hours**
1	Vector spaces; subspaces; Linear independence; Basis vector sets; dimensions; coordinate vectors;
2	Linear transformations, matrix representation of transformations.
3	Construct subspace, compute span and compare representations in different spaces.
4	Inner product, orthogonal sets, orthogonal projections, orthogonal bases; Gram-Schmidt orthogonalization process.

Tutorials (continued)	
5	QR factorizations of matrices, least square problems, applications to linear models (least square lines and least square fitting of other curves)
6	Gram-Schmidt orthogonalization and QR factorization to solve an overdetermined linear system, analyse orthogonal projections, and interpret the least squares solution geometrically and algebraically.
7	Diagonalization, Quadratic forms, Constrained Optimization (Lagrange multipliers).
8	Singular Value Decomposition; Applications to image processing and statistics; Principal Component Analysis.
9	Singular Value Decomposition (SVD) on real datasets to extract principal components, analyse quadratic forms, and apply constrained optimization using Lagrange multipliers in practical scenarios like image compression or data clustering.
10	Poisson, exponential, normal, Gamma, Probability mass function (pmf), Probability density function (pdf).
11	Joint distribution, Bayes' Theorem, Mathematical expectation, conditional expectation.
12	Students simulate discrete and continuous random variables in Python or MATLAB to visualize PMFs and PDFs, compute expectations and conditional expectations, and apply Bayes' Theorem to real-world inference problems using parametric distributions (e.g., binomial, normal, Poisson).
13	Central Limit Theorem, Probabilistic inequalities (Markov and Chebyshev's), Markov chains. Methods of Moments and Maximum Likelihood.
14	Exploring Sampling and Estimation through Simulation - Students generate random samples from known distributions (e.g., normal, exponential) to empirically verify the Central Limit Theorem, compare sampling distributions of mean and variance, and evaluate Markov and Chebyshev inequalities. They estimate parameters using Method of Moments and Maximum Likelihood, and assess estimator properties using visualizations and metrics like bias and variance.
15	Hypothesis Testing and Markov Chains in Practice - Students apply t-tests, chi-square tests, and p-values to real or synthetic datasets to test statistical hypotheses. In parallel, they model simple Markov chains (e.g., weather patterns or board games) using transition matrices, simulate the system over time, and analyse steady-state behaviour and transition probabilities.

Semester - **I**			
Course: **ADVANCED DATA STRUCTURES AND ALGORITHMS**			
Program	**M.Tech CSE**	Category	**Major (Core)**
Course Code	**CS5000**	CIE Marks	**70**
Credits (L:T:P)	**4 (3:0:2)**	SEE Marks	**30**
Hours	**45L + 0T + 30P = 75**	SEE Mode	**Practical**

Course Objectives: students will be able to	
1	To provide an in-depth understanding of advanced data structures such as balanced trees, heaps, hash structures, and disjoint sets, and their applications in solving computational problems efficiently.
2	To develop the ability to analyze and design algorithms using techniques like divide- and-conquer, dynamic programming, greedy methods, and backtracking, with an emphasis on both correctness and complexity.
3	To equip students with the knowledge of advanced graph algorithms and string processing techniques, enabling them to model and solve real-world problems involving networks and text data.
4	To introduce computational complexity theory, including NP-completeness, approximation algorithms, and randomized approaches, thereby preparing students for tackling intractable or large-scale computational problems.

Module - 1	**9 hours**
Trees: Red-Black Trees, B-Trees, Splay Trees, Heaps: Binary Heaps, Binomial Heaps, Fibonacci Heaps, Hashing: Collision resolution, Hash functions, Hash tables, Disjoint Sets: Union-Find, Applications in Kruskal's Algorithm, Segment Trees, Fenwick Trees, Persistent Data Structures (intro), Amortized Analysis.	

Module - 2	**9 hours**
Graph representations and traversals (BFS, DFS), Topological Sorting, Strongly Connected Components (Kosaraju's Algorithm), Minimum Spanning Tree: Prim's and Kruskal's Algorithms, Shortest Path: Dijkstra's, Bellman-Ford, Floyd-Warshall, Network Flow: Ford-Fulkerson, Edmonds-Karp, Johnson's Algorithm, A* Search (intro), Dynamic Connectivity using DSU	

Module - 3	**9 hours**
Greedy Algorithms: Activity Selection, Huffman Coding, Closest Pair, Dynamic Programming: Matrix Chain Multiplication, LCS, 0/1 Knapsack, Backtracking: N-Queens, Sudoku Solver, Bitmask Dynamic Programming (intro), Branch and Bound.	

Module - 4	**9 hours**
String Matching: Naive, Rabin-Karp, KMP, Z-algorithm, Tries and Suffix Trees, Geometry: Convex Hull (Graham Scan, Jarvis March), Line Segment Intersection, Suffix Arrays, Rolling Hash techniques.	

Module - 5	**9 hours**
Complexity Classes: P, NP, NP-Hard, NP-Complete, Cook-Levin Theorem (conceptual), Approximation Algorithms: Vertex Cover, TSP (heuristics), Randomized Algorithms: Monte Carlo and Las Vegas algorithms, Probabilistic Analysis, Randomized QuickSort, Skip Lists, Parameterized Complexity (intro), Streaming Algorithms, Bloom Filters.	

Course Outcomes: After completing the course, the students will be able to	
1	Apply advanced data structures like B trees, heaps, and disjoint sets to solve complex problems.
2	Analyze different algorithm design strategies such as divide-and-conquer, greedy, and dynamic programming to understand their problem-solving approaches, efficiency, and applicability in various computational scenarios.
3	Apply advanced graph algorithms and string matching techniques to solve real-world problems effectively.
4	Evaluate the complexity of algorithms and use approximation or randomized techniques to address NP-complete problems.

Text Books	
1	Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms (3rd ed.). MIT Press.ISBN-13: 978-0-262-03384-8
2	Basu, S. K. (2013). Design methods and analysis of algorithms. PHI Learning Pvt. Ltd. ISBN-13: 978-8120347465

Reference Books and Resources	
1	Weiss, M. A. (2014). Data structures and algorithm analysis in C++ (4th ed.). Pearson. ISBN-13: 9780132847377
2	Kleinberg, J., & Tardos, É. (2005). Algorithm design. Pearson Education. ISBN-13: 978-0-321-29535-4
3	Dasgupta, S., Papadimitriou, C. H., & Vazirani, U. V. (2008). Algorithms. McGraw- Hill. ISBN-13: 978-0073523408
4	Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1983). Data structures and algorithms. Addison-Wesley. ISBN-13: 978-0-201-00023-8
5	de Berg, M., Cheong, O., van Kreveld, M., & Overmars, M. (2008). Computational Geometry: Algorithms and Applications (3rd ed.). Springer. ISBN-13: 978- 3540779735
6	Vazirani, V. V. (2001). Approximation Algorithms. Springer. ISBN-13: 978-3540653677

Lab Programs / Practical	**30 Hours**
Part A: Lab Programs	
1	Implement Red-Black Trees to maintain a balanced BST. Analyze rotation operations and height balance.

Lab Programs / Practical (continued)	
2	Implement Binary Heap and compare with Binomial Heap or Fibonacci Heap (analysis-based)
3	Implement hashing using chaining and open addressing. Evaluate performance with different hash functions.
4	Implement with path compression and union by rank. Apply to dynamic connectivity problems.
5	Implement Kruskal's and Prim's algorithms. Compare efficiency on different graph inputs.
6	Build and analyze the Ford-Fulkerson algorithm or Edmonds-Karp algorithm for computing network flows. Apply them to real-world scenarios like job assignment or transportation networks.
7	Implement DP and Greedy solutions. Compare their performance and results to understand where each technique excels or fails.
8	Implement algorithms such as Knuth-Morris-Pratt (KMP) and Rabin-Karp for efficient string searching. Test them on various pattern and text combinations to evaluate performance.
9	Implement algorithms like Graham's scan or Jarvis March to compute the convex hull of a set of points. Use this to understand computational geometry fundamentals.
10	Apply approximation techniques to problems like vertex cover or traveling salesman. Analyze how close the solutions are to the optimal and evaluate their efficiency.
11	Explore randomized algorithms such as QuickSort or Monte Carlo methods. Conduct experiments to study average-case performance and probability of correctness.
Part B:	
Develop a mini project applying advanced data structures/algorithms (e.g., route optimization, text search engine, network analysis, or recommendation system). Include performance analysis and report.	

Semester - **I**			
Course: **ADVANCED DATABASE MANAGEMENT SYSTEM**			
Program	**M.Tech CSE**	Category	**Major (Core)**
Course Code	**CS5200**	CIE Marks	**70**
Credits (L:T:P)	**4 (3:0:2)**	SEE Marks	**30**
Hours	**45L + 0T + 30P = 75**	SEE Mode	**Practical**

Course Objectives: students will be able to	
1	To provide in-depth knowledge of advanced data models, including object-oriented, object-relational, and semi-structured databases.
2	To introduce the design, architecture, and implementation aspects of distributed and parallel databases.
3	To develop competence in data warehousing, OLAP, and data mining techniques for decision support systems.
4	To expose learners to advanced database applications and emerging trends such as active, temporal, spatial, mobile, and multimedia databases.

Module - 1	**9 hours**
Database design – Query processing. Data modelling – ER – EER –Object Oriented Databases – Object Relational Databases, Document oriented Databases – Background of NoSQL –XML document – Structure of XML Data – XML Document Schema – Querying and Transformation – API –Storage of XML Data – XML Applications.	

Module - 2	**9 hours**
Overview of OOP; Complex objects; Identity, structure etc. Object model of ODMG, Object definition Language ODL; Object Query Language OQL; Conceptual design of Object database. Overview of object relational features of SQL; Object-relational features of Oracle; Implementation and related issues for extended type systems; syntax and demo examples, The nested relational model.	

Module - 3	**9 hours**
Data Distribution – Distributed Transactions, Parallel Databases – Performance measure – Parallel operations for relational operations, Information Integration – Federated Database – Data Warehouses – Mediators – Schema matching methods.	

Module - 4	**9 hours**
Introduction to decision support; OLAP, multidimensional model; Window queries in SQL; Finding answers quickly; Implementation techniques for OLAP; Data Warehousing; Views and Decision support, View materialization, Maintaining materialized views. Introduction to Data Mining; Counting co-occurrences; Mining for rules; Tree-structured rules; ROC and CMC Curves; Clustering; Similarity search over sequences; Incremental mining and data streams; Additional data mining tasks.	

Module - 5	**9 hours**

Active database concepts and triggers; Temporal, Spatial, and Deductive Databases – Basic concepts. More Recent Applications: Mobile databases; Multimedia databases; Geographical Information Systems; Genome data management.

Course Outcomes: After completing the course, the students will be able to	
1	Differentiate between traditional, object-oriented, object-relational, and document- oriented databases and explain their features and use-cases.
2	Apply object-oriented and object-relational query languages for advanced data modeling and query operations.
3	Analyze the architecture and operations of distributed and parallel databases for improving performance and integration.
4	Design and implement data warehouse schemas, OLAP queries, and data mining solutions for decision support applications.

Text Books	
1	Silberschatz, Abraham, Henry F. Korth, and S. Sudarshan. Database System Concepts. 7th Edition, McGraw-Hill, 2021. ISBN: 978-93-9072750-6
2	Elmasri, Ramez, and Shamkant B. Navathe. Fundamentals of Database Systems. 7th Edition, Pearson, 2017. ISBN: 978-93-3258270-5
3	Ramakrishnan, Raghu, and Johannes Gehrke. Database Management Systems. 3rd Edition, McGraw-Hill, 2003. ISBN: 978-93-392-1311-4
4	Jiawei Han and MichelineKamber, Data Mining Concepts and Techniques , Morgan, KaufnamPublishers, 3r Edition,2011, ISBN: 9780123814791

Reference Books and Resources	
1	Connolly, Thomas, and Carolyn Begg. Database Systems: A Practical Approach to Design, Implementation, and Management. 6th Edition, Pearson, 2019. ISBN: 978- 9353438913
2	Sadalage, Pramod J., and Martin Fowler. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Addison-Wesley, 2012. ISBN: 978-0-321-82662-6
3	Angles, Renzo, and Claudio Gutierrez. Graph Data Management: Techniques and Applications. IGI Global, 2012. ISBN: 978-1-61350-053-8
4	Kleppmann, Martin. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. O'Reilly Media.

Lab Programs / Practical	**30 Hours**
Part A: Lab Programs	
1	Design an EER diagram and transform it into a relational schema.
2	Implement and manage a document-oriented database using MongoDB.
3	Store, retrieve, and query XML documents using XPath/XQuery.
4	Develop complex objects using PostgreSQL user-defined types.

Lab Programs / Practical (continued)	
5	Demonstrate and analyze object inheritance in object-relational databases.
6	Construct ODL definitions and execute OQL queries for object-oriented database design.
7	Simulate and evaluate a distributed database environment.
8	Implement and test the Two-Phase Commit Protocol using Python.
9	Design, integrate, and query a federated database system.
10	Create and model a star schema for a sales data warehouse.
11	Formulate and analyze OLAP queries using ROLLUP, CUBE, and GROUPING SETS.
12	Apply window functions to analyze decision-support queries.
13	Perform and interpret association rule mining using the Apriori algorithm.
14	Develop and execute database triggers for active database applications.
15	Demonstrate and evaluate spatial or temporal queries using PostgreSQL.
Part B: Integrated Mini-Project	
1	Analyze & Model: Deconstruct a real-world domain (e.g., Healthcare or Finance) to design an EER Diagram and implement it using Object-Relational features like inheritance and user-defined types.
2	Integrate Multi-Model Data: Contrast and implement structured storage with MongoDB document-oriented collections and XML/XPath querying to manage semi-structured data.
3	Execute Distributed Logic: Architect a distributed database simulation and develop a Two-Phase Commit (2PC) protocol to ensure atomic transactions across fragmented nodes.
4	Evaluate Decision Support: Construct a Star Schema to perform OLAP operations (CUBE, ROLLUP) and utilize active database triggers for automated system responses.
5	Discover Patterns: Apply the Apriori mining algorithm to analyze datasets for hidden associations and integrate spatial or temporal queries for specialized data insights.

Semester - **I**			
Course: **ADVANCED MACHINE LEARNING**			
Program	**M.Tech CSE**	Category	**Major (Elective)**
Course Code	**CS5201**	CIE Marks	**70**
Credits (L:T:P)	**3 (2:0:2)**	SEE Marks	**30**
Hours	**30L + 0T + 30P = 60**	SEE Mode	**Theory**

Course Objectives: students will be able to	
1	To provide a comprehensive understanding of machine learning fundamentals, including regression, classification, and ensemble techniques.
2	To develop the ability to build, train, and optimize deep learning architectures using advanced neural network techniques.
3	To explore unsupervised learning and deep reinforcement learning algorithms for solving complex, high-dimensional, and sequential decision-making problems.
4	To introduce trustworthy and responsible machine learning practices, focusing on privacy, security, and ethical considerations.

Module - 1	**6 hours**
Introduction to Machine Learning Fundamentals, Linear Regression and Multiple Linear Regression, Loss Functions: Mean Squared Error (MSE), Cross-Entropy, etc., Logistic Regression for Classification, Decision Tree and Random Forest Algorithms, Ensemble Methods: Bagging, Boosting, AdaBoost, Gradient Boosting Machines (GBM), Stacking, Support Vector Machines (SVM) and Kernel Methods, Hyperparameter Tuning Techniques: Grid Search, Random Search, Cross-Validation.	

Module - 2	**6 hours**
Neural Network Fundamentals: Biological and artificial neurons, Activation Functions (ReLU, Leaky ReLU, GELU, Tanh). Feedforward Neural Networks, Loss Functions (Cross-Entropy, MSE), Gradient Descent and its variants (SGD, Adam, RMSprop), Backpropagation algorithm. Introductions to Convolutional neural networks.	

Module - 3	**6 hours**
Dimensionality Reduction: PCA, t-SNE, Clustering: K-Means, DBSCAN, Hierarchical Clustering.	

Module - 4	**6 hours**
Reinforcement Learning (RL), Key Concepts in Reinforcement Learning (RL), Agent, Environment, Actions and Action Space, Observations and State, Reward and Total Reward (Return), Discounted Total Reward, Q-function, Deep Reinforcement Learning Algorithms, Deep Q Networks (DQN), Policy Gradient Methods.	

Module - 5	**6 hours**

Federated Learning and Distributed Machine Learning, Adversarial Machine Learning and Robustness, Privacy-Preserving Machine Learning: Differential Privacy, Homomorphic Encryption, Ethical Considerations in Machine Learning.

Course Outcomes: After completing the course, the students will be able to	
1	Apply appropriate machine learning algorithms and techniques to solve classification, regression, and decision-making problems across various domains
2	Design, develop, and optimize learning architectures by selecting suitable models, functions, and optimization strategies for effective problem-solving.
3	Analyze complex datasets using unsupervised and sequential learning methods to uncover hidden patterns and support intelligent decision-making.
4	Develop machine learning solutions that emphasize privacy, and ethical responsibility in real-world applications.

Text Books	
1	Ethem Alpaydin, Introduction to Machine Learning, MIT Press, Prentice Hall of India, forth editions, 2020. ISBN13: 978-0262043793
2	Foundations of Machine Learning Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar MIT Press, Second Edition, 2018. ISBN: 9780262018258
3	Introduction to Machine Learning with Python: A Guide for Data Scientists by Andreas C. Müller and Sarah Guido, O'Reilly Media (2016), ISBN 978-1449369415

Reference Books and Resources	
1	Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT Press, 2016, ISBN: 9780262035613.
2	Aurélien Géron. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. Shroff/O'Reilly. Third edition. ISBN-10: 1492032646

Lab Programs / Practical	**30 Hours**
Part A: Lab Programs	
1	Implementation of Linear and Logistic Regression Models This exercise focuses on linear regression and logistic regression models from scratch and using Scikit-learn to solve real-world regression and binary classification problems. They will evaluate the models using appropriate performance metrics such as Mean Squared Error (MSE), Accuracy, and ROC curves.
2	Decision Trees and Random Forests for Classification This exercise focuses on building and visualizing decision tree and random forest classifiers using a dataset with multiple features. Students will learn how to tune hyperparameters like maximum depth and number of estimators to reduce overfitting and improve model generalization.

Lab Programs / Practical (continued)	
3	Ensemble Learning: Bagging, Boosting, and Stacking Techniques Students will experiment with ensemble methods including Bagging, AdaBoost, Gradient Boosting Machines (GBM), and stacking to enhance predictive performance. They will compare these methods in terms of accuracy, robustness, and computational efficiency using benchmark datasets.
4	Support Vector Machines with Kernel Methods In this lab, students will develop SVM models using different kernel functions (linear, polynomial, RBF) to handle complex decision boundaries. They will perform grid search to identify the best kernel and hyperparameters to optimize classification performance.
5	Building and Training Feedforward Neural Networks Students will build basic feedforward neural networks using TensorFlow or PyTorch. They will experiment with different activation functions (ReLU, Leaky ReLU, GELU, Tanh), loss functions, and optimizers to understand their impact on convergence and accuracy.
6	Unsupervised Learning using Dimensionality Reduction and Clustering This exercise involves applying dimensionality reduction techniques such as Principal Component Analysis (PCA), t-SNE, and Autoencoders to high-dimensional datasets. Students will also perform clustering using DBSCAN and hierarchical clustering to discover hidden patterns.
7	Deep Reinforcement Learning using Deep Q-Networks Students will implement a Deep Q-Network (DQN) agent to solve a simple reinforcement learning environment (e.g., CartPole). They will analyze the agent's learning curve, tune the discount factor, and explore the effects of exploration vs. exploitation.
8	Federated Learning and Privacy-Preserving Machine Learning In this lab, students will simulate a federated learning environment using TensorFlow Federated or a similar framework. They will understand the concepts of distributed training and demonstrate privacy-preserving techniques using differential privacy.
9	Adversarial Machine Learning and Model Robustness Students will generate adversarial examples using methods like the Fast Gradient Sign Method (FGSM) to test the robustness of trained models. They will analyse how minor perturbations can fool machine learning models and explore basic adversarial defences.
Part B:	
Design and implement an advanced machine learning solution for predictive analytics using real-world datasets and performance optimization techniques.	

Semester - **I**			
Course: **NATURAL LANGUAGE PROCESSING**			
Program	**M.Tech CSE**	Category	**Major (Elective)**
Course Code	**CS5202**	CIE Marks	**70**
Credits (L:T:P)	**3 (2:0:2)**	SEE Marks	**30**
Hours	**30L + 0T + 30P = 60**	SEE Mode	**Theory**

Course Objectives: students will be able to	
1	Introduce foundational concepts of Natural Language Processing and linguistic structures.
2	Explore classical and deep learning techniques for language modelling and text processing.
3	Develop the ability to implement fine-tune transformer models for real-world NLP tasks.
4	Enable hands-on proficiency in multilingual, speech-enabled, and multimodal NLP applications.

Module - 1	**6 hours**

Introduction to NLP and Linguistic Essentials: syntax, semantics, morphology, and pragmatics. Text Preprocessing Techniques: tokenization, normalization, stop-word removal, stemming, lemmatization. Part-of-Speech (POS) Tagging: rule-based and probabilistic methods. Named Entity Recognition (NER): dictionary-based, CRF-based methods. N-gram Language Models: smoothing, backoff, and perplexity. Classical Text Representations: Bag-of-Words (BoW), TF-IDF. Word Embedding Techniques: Word2Vec (CBOW, Skip-gram), GloVe, FastText. Comparison of classical and distributed word representations.

Module - 2	**6 hours**

Recurrent Neural Networks (RNN): architecture, training issues (vanishing gradients). Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU): cell structure and variants. Convolutional Neural Networks (CNN) for text classification. Sequence-to-sequence models with encoder-decoder structure. Attention Mechanism: additive vs. multiplicative attention, Bahdanau and Luong attention. Evaluation Metrics: Precision, Recall, F1-Score, BLEU, ROUGE. Word alignment and context windows in sequential tasks. Case studies: sentiment classification, spam detection.

Module - 3	**6 hours**

Transformer Architecture: multi-head attention, positional encoding, residual connections. Pretrained Encoder Models: BERT, RoBERTa, ALBERT – architecture and training strategies. Generative Pretrained Transformers: GPT-2, GPT-3, and decoder-only models. Text-to-Text Models: T5 and BART for summarization and translation. Transfer Learning: pre-training vs. fine-tuning, feature extraction, task adaptation. Hugging Face Ecosystem: tokenizers, pretrained model usage, pipelines. Zero-shot and few-shot learning using LLMs. Applications: sentence classification, entity extraction, semantic similarity.

Module - 4	**6 hours**
Question Answering (QA) Systems: Closed-domain QA (SQuAD-style), Open-domain QA with retrieval-based techniques. Summarization: Extractive methods: LexRank, TextRank. Abstractive methods: sequence-to-sequence and transformer-based models. Conversational Agents: Dialogue act classification, Rule-based vs. machine-learned dialogue systems. Chatbot Development Frameworks: Dialog flow, Rasa, Microsoft Bot Framework. Text Generation: Prompt engineering, Controlled and conditional generation techniques. Real-world applications in healthcare, legal tech, customer service.	

Module - 5	**6 hours**
Automatic Speech Recognition (ASR): Acoustic and language models, Wav2Vec 2.0, Whisper. Text-to-Speech (TTS): Tacotron 2, FastSpeech 2 – architecture and synthesis techniques. Multilingual NLP: Challenges in low-resource languages, mBERT, XLM-R, IndicNLP, MuRIL, Code-switching and cross-lingual transfer. Multimodal Learning: Joint modeling of text and vision, CLIP: vision-language alignment, LLaVA: multimodal instruction following. Case studies: Image captioning, visual question answering, multilingual QA.	

Course Outcomes: After completing the course, the students will be able to	
1	Apply text preprocessing techniques and represent text using classical and modern embeddings.
2	Implement deep learning models (RNN, LSTM, CNN) for sequence tasks and text classification.
3	Analyze transformer-based architectures and fine-tune them for advanced NLP tasks.
4	Develop NLP applications integrating multilingual, speech, and vision modalities.

Text Books	
1	Daniel Jurafsky & James H. Martin. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models. Pearson, 3rd Edition, January 2025.
2	Lewis Tunstall, Leandro von Werra, Thomas Wolf. Natural Language Processing with Transformers. O'Reilly Media. Revised Edition, July 2022, ISBN-10:1098136799, ISBN-13:978-1098136795.
3	Philipp Koehn. Neural Machine Translation. Cambridge University Press, First Edition, November 2020, ISBN-10:108497322, ISBN-13:978-1108497329.
4	Jacob Eisenstein. Introduction to Natural Language Processing (Adaptive Computation and Machine Learning series). The MIT Press, Illustrated Edition, October 2019, ISBN- 10:0262042843, ISBN-13:978-0262042840.
5	Yoav Goldberg. Neural Network Methods for Natural Language Processing (Synthesis Lectures on Human Language Technologies). Morgan & Claypool Publishers, First Edition, 2017. ISBN-10: 1627052984, ISBN-13: 978-1627052986.
6	Daniel Bikel & Imed Zitouni. Multilingual Natural Language Processing Applications: From Theory to Practice. IBM Press, First Edition, May 2012, ISBN-10:0137151446, ISBN-13:978-0137151448.

Reference Books and Resources	
1	Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, Harshit Surana. Practical Natural Language Processing. O'Reilly Media, First Edition, July 2020, ISBN-10:1492054054, ISBN-13:978-1492054054.
2	Delip Rao and Brian McMahan. Natural Language Processing with PyTorch. 'Reilly Media, First Edition, December 2019. ISBN-10: 1491978236, ISBN-13: 978- 1491978238.
3	Dipanjan Sarkar. Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from Your Data. Apress, Second Edition, July 2019. ISBN- 10: 1484243533, ISBN-13: 978-1484243531.
4	Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep Learning (Adaptive Computation and Machine Learning series). The MIT Press. First Edition, November 2016, ISBN-10: 0262035618, ISBN-13: 978-0262035613.
5	Steven Bird, Ewan Klein, and Edward Loper. Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit. O'Reilly Media, First Edition, June 2009. ISBN-10: 0596516495, ISBN-13: 978-0596516499.
6	Christopher D. Manning and Hinrich Schütze. Foundations of Statistical Natural Language Processing. MIT Press, First Edition, June 1999. ISBN-10: 0262133601, ISBN-13: 978-0262133609.

Lab Programs / Practical	30 Hours
Part A: Lab Programs	
1	Text Preprocessing: Tokenization, stop-word removal, lemmatization, POS tagging, NER. Tools: spaCy, NLTK
2	Word Representation Techniques: Compute BoW and TF-IDF vectors for a corpus. Compare with GloVe/Word2Vec embeddings. Tools: scikit-learn, genism.
3	Sequence Modelling for Text Classification: Build a sentiment classifier using LSTM or GRU on IMDB or product reviews. Tools: Keras or PyTorch.
4	Attention and CNN for Text: Implement attention mechanism or text classification using CNN. Tools: TensorFlow/Keras, PyTorch.
5	Fine-tuning BERT on Custom Dataset: Text classification or NER using Hugging Face Transformers. Dataset: IMDb, AGNews, or any custom labelled dataset. Tools: Hugging Face Transformers, Datasets, scikit-learn.
6	Language Generation with GPT-2 or T5: Generate text given a prompt or paraphrase sentences. Tools: Transformers, Pipeline, GPT2LMHeadModel.
7	Question Answering using Pretrained Models: Use a SQuAD-style context-question- answer pipeline with BERT. Tools: Hugging Face, pipeline(task="question- answering").
8	Summarization: Implement an abstractive text summarization system using state-of- the-art transformer models like BART or T5. Tools: Transformers, Hugging Face pipeline.
9	Machine Translation: Translate sentences using MarianMT or mBART. Tools: Hugging Face Transformers Library.

Lab Programs / Practical (continued)	
10	Speech-to-Text using Whisper or Wav2Vec 2.0: Transcribe speech into text from .wav files. Tools: OpenAI Whisper, Facebook Wav2Vec2, transformers.
11	Text-to-Speech with Tacotron or FastSpeech: Convert text to speech and play audio output. Tools: TTS (Coqui), ESPNet, PyTorch, Tacotron.
12	Multimodal NLP Demo: Use CLIP or LLaVA to generate captions or match images with text prompts. Tools: OpenCLIP, LLaVA, transformers, datasets.

Part B:

Mini Project involving the development of an AI-enabled NLP application using modern AI tools and frameworks for tasks such as sentiment analysis, chatbot development, text summarization, question answering, multilingual processing, or speech-based NLP. The project includes dataset preparation, linguistic preprocessing, implementation of deep learning or transformer models, performance evaluation using standard NLP metrics, model fine-tuning, final demonstration, viva voce, and submission of a technical report.

Semester - **I**			
Course: **ML FOR DATA SCIENCE**			
Program	**M.Tech CSE**	Category	**Major (Elective)**
Course Code	**CS5203**	CIE Marks	**70**
Credits (L:T:P)	**3 (2:0:2)**	SEE Marks	**30**
Hours	**30L + 0T + 30P = 60**	SEE Mode	**Theory**

Course Objectives: students will be able to	
1	Understand core machine learning concepts and mathematical foundations
2	Apply machine learning algorithms to real-world datasets
3	Design and optimize machine learning workflows
4	Analyse machine learning applications

Module - 1	**7 hours**
Machine Learning Paradigms: - Supervised Learning, Unsupervised Learning, Semi Supervised Learning, Reinforcement Learning, Parameter Estimation Methods and Optimization: -Maximum Likelihood Estimation (MLE), Maximum A Posteriori Estimation (MAP), Bayes Estimation, Least Square Estimation (LS), Minimum Mean Square Estimation (MMSE), The Gradient Descent Method and its variants (Simple Gradient Descent Method, Stochastic Gradient Method, Batch Gradient Method),Data Preprocessing: - Feature scaling, missing data, Encoding, Train/test split, Cross-validation - K-Fold Cross-Validation and Stratified K-Fold Cross-Validation	

Module - 2	**6 hours**
Regression Techniques: - Linear regression, Ridge, Lasso, Classification Algorithms: - Logistic Regression, Naïve Bayes, K-Nearest Neighbours, Decision Trees – ID3, Support Vector Machines, Model Evaluation: - Confusion Matrix, Accuracy, Precision, Recall, F1- Score, ROC-AUC	

Module - 3	**6 hours**
Clustering: - K-Means, Hierarchical, DBSCAN Dimensionality Reduction: - PCA, t-SNE,Association Rule Learning: - Apriori, FP-Growth (concepts only)	

Module - 4	**6 hours**
Ensemble Methods: - Bagging, Boosting (AdaBoost, Gradient Boosting), Introduction to XGBoost, Model Selection and Tuning: - Grid Search, Randomized Search, Bias-Variance trade-off, Interpretability: - SHAP, LIME (introduction)	

Module - 5	**5 hours**
Working with Real-world Data: - Handling unstructured data (text/images overview), Imbalanced data, Feature engineering, Pipeline Design: - Building ML pipelines, Ethics in ML: - Fairness, Bias, Explainability, Privacy, Recent Trends: - AutoML, ML Ops (brief), LLMs in Data Science.	

Course Outcomes: After completing the course, the students will be able to	
1	Explain key concepts and mathematical foundations of machine learning algorithms used in data science.
2	Implement supervised and unsupervised learning models using Python libraries and evaluate their performance using appropriate metrics.
3	Apply dimensionality reduction and model tuning techniques to improve model performance and interpretability.
4	Develop end-to-end machine learning pipelines for real-world data science problems, considering practical and ethical aspects.

Text Books	
1	E. Alpaydın, Introduction to Machine Learning, 4th ed. Cambridge, MA, USA: MIT Press, 2020. ISBN: 978-0-262-04379-3.
2	D. P. Kroese, Z. I. Botev, T. Taimre, and R. Vaisman, Data Science and Machine Learning: Mathematical and Statistical Methods, 1st ed. CRC Press, 2019. ISBN: 978-1-4987-5908-4.
3	M. Mohri, A. Rostamizadeh, and A. Talwalkar, Foundations of Machine Learning, 2nd ed. MIT Press, 2018. ISBN: 978-0-262-03940-6.
4	A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 1st ed. O'Reilly Media, 2017. ISBN: 978-1-4919-6214-3.

Reference Books and Resources	
1	M. Mitchell, Machine Learning. McGraw-Hill Education, 1997. ISBN: 978-0-07-042807-2.
2	C. M. Bishop, Pattern Recognition and Machine Learning. Springer, 2016. ISBN: 978-0-387-31073-2.

Lab Programs / Practical	**30 Hours**
Part A: Lab Programs	
1	Data exploration using Pandas and NumPy (e.g., UCI datasets) Description: - This lab focuses on performing exploratory data analysis (EDA) using Python libraries Pandas and NumPy. Students will load a dataset (e.g., from the UCI Machine Learning Repository), clean and preprocess the data, and compute descriptive statistics to understand the data's structure, patterns, and anomalies. The activity includes operations like handling missing values, summarizing distributions, detecting outliers, computing correlations, and basic group-wise analysis.
2	Implement regression and classification models on scikit-learn Description: - This lab introduces students to building and evaluating regression and classification models using the scikit-learn library. Students will train models like Linear Regression, Logistic Regression, k-NN, and Decision Trees, and assess their performance using standard metrics.

Lab Programs / Practical (continued)	
3	Apply 5-fold cross-validation on Logistic Regression and Decision Tree models, visualize fold-wise metrics, and analyse performance variance. Description: - In this lab, students will apply 5-fold cross-validation on Logistic Regression and Decision Tree models using scikit-learn. They will record and visualize fold-wise metrics such as Accuracy and F1-score, and analyse the variance in performance across folds to understand model stability and generalization.
4	Apply clustering techniques for customer segmentation and visualize high-dimensional data using PCA and t-SNE. Description: - This lab involves applying clustering techniques like K-Means for customer segmentation. Students will use PCA and t-SNE to reduce data dimensionality and visualize clusters in 2D space. The activity helps in understanding grouping patterns in high-dimensional customer data and interpreting unsupervised learning results.
5	Implement hyperparameter tuning using Grid Search or Random Search, and compare the performance of at least two machine learning models on a given dataset. Description: - This lab focuses on using Grid Search or Random Search to tune model hyperparameters and compare the performance of two machine learning models on a given dataset using evaluation metrics.
6	End-to-end mini project on a real dataset (Kaggle/UCI) Description: - In this lab, students will work on a real-world dataset from Kaggle or UCI to build an end-to-end machine learning pipeline. This includes data preprocessing, model selection, training, evaluation, tuning, and result interpretation, simulating a complete data science workflow.

Part B: Project: End-to-End Machine Learning on a Real Dataset
Description:- Students shall undertake a mini-project individually or in groups using a real-world dataset sourced from Kaggle or the UCI Machine Learning Repository. The project will focus on identifying a relevant problem in the domain of Machine Learning for Data Science and developing an appropriate solution through a complete machine learning pipeline. The work shall include data preprocessing, exploratory analysis, feature engineering, model selection, training, evaluation, and hyperparameter tuning. Students are expected to justify their methodology through a brief literature review and ensure that each project addresses a unique or distinct problem statement. The project should demonstrate the practical application of concepts learned during the course and conclude with result interpretation, model comparison, and meaningful insights derived from the data.

Semester - **I**			
Course: **PRINCIPLES OF DATA ANALYTICS AND VISUALIZATION**			
Program	**M.Tech CSE**	Category	**Major (Elective)**
Course Code	**CS5500**	CIE Marks	**70**
Credits (L:T:P)	**3 (2:0:2)**	SEE Marks	**30**
Hours	**30L + 0T + 30P = 60**	SEE Mode	**Theory**

Course Objectives: students will be able to	
1	Introduce learners to the core principles of data visualization, including design, perception, and effective communication of data.
2	Building interactive multi page reports
3	Data enrichment & advanced visual analytics using analytical expressions
4	Advanced Data Modelling for a responsive dashboard

Module - 1	**6 hours**
Introduction to Data Visualisation, Types of Data Visualisation. Exploring the tools' landscape. Building your first interactive dashboard using the report design best practices by understanding the importance of page layout design, conditional formatting, colour template selection etc.	

Module - 2	**6 hours**
Introduction to Exploratory Data Analysis (EDA), Ingesting Data, Data Pre-processing, Dates, Table Calculations and Key Performance Indicators. Using the no-code options for advanced data pre-processing techniques and automating the ETL pipeline.	

Module - 3	**6 hours**
Import data from multiple sources into the staging area and convert the tables into star schema or snowflake schema. Fine tune the models by avoiding many to many connections, using bridge tables. Create date-time tables when necessary. Optimise the model as per the business requirement for a highly responsive dashboard.	

Module - 4	**6 hours**
Exploring DAX (Data Analytics Expressions) which is a programming interface similar to Python. Create additional columns when required (data enrichment) or create measures when necessary. Effectively use the measures in the visuals for effectively communicating the results of analysis.	

Module - 5	**6 hours**
Sharing the reports to stakeholders by publishing it to the cloud service. Applying row level security parameters while publishing the report.	

Course Outcomes: After completing the course, the students will be able to	
1	Apply the foundational data visualisation principles to create engaging reports

Course Outcomes: After completing the course, the students will be able to (continued)	
2	Apply visualisation design techniques to create interactive charts and draw inferences using conditional formatting, slicers and filters
3	Analyse data using analytical expressions and present them visually
4	Build and share multipage the report in cloud with the respective stakeholders with enhanced security features

Text Books	
1	Greg Deckler, Brett Powell, Mastering Microsoft Power BI: Expert techniques to create interactive insights for effective data analytics and business intelligence, 2nd Edition, Packt Publishing,2018, ISBN-13: 978-1801811484

Reference Books and Resources	
1	Brent Dykes, Effective Data Storytelling: How to Drive Change with Data, Narrative and Visuals, 1st Edition, Wiley Publication,2019 Edition ISBN-10 :1119615712, ISBN- 13:978-1119615712

Lab Programs / Practical	**30 Hours**
Part A: Lab Programs	
1	Installing the analytics tool and setting up the environment
2	Implement data pre-processing / data cleaning
3	Advanced data cleaning and automation of data pre-processing
4	Building the first report by analysing sales data
5	Data modelling for visualization
6	Optimizing data modelling for a responsive dashboard
7	Use the Gen AI features in the tool for creating a rapid report template
8	Getting started with the use analytical expressions reports
9	Exploring advanced use cases of analytical expressions
10	Creating Date Tables for time series analysis
11	Formatting the reports with cosmetic detailing for a professional visual appeal
12	Building multi page report with buttons and page navigation
13	Applying row level security and sharing the reports in cloud
Part B:	

Lab Programs / Practical (continued)	
1	An organization generates large amounts of data from multiple sources and requires an intelligent Business Intelligence (BI) solution for effective analysis and decision-making. Using Generative AI tools along with modern BI platforms, develop an interactive dashboard solution by performing data ingestion, exploratory data analysis (EDA), data preprocessing, KPI generation, and ETL automation. Utilize Generative AI tools to assist in data cleaning, schema design, DAX query generation, dashboard design recommendations, visualization selection, and report summarization. Design and optimize the data model using suitable schemas and relationships, and create meaningful visualizations by applying dashboard design best practices such as layout optimization, colour selection, and conditional formatting. Finally, publish the dashboard to a cloud platform with appropriate security mechanisms including Row Level Security (RLS), and demonstrate how Generative AI can enhance reporting, insight generation, and business analytics.

Semester - **II**			
Course: **HIGH PERFORMANCE COMPUTER ARCHITECTURE**			
Program	**M.Tech CSE**	Category	**Major (Core)**
Course Code	**CS5100**	CIE Marks	**70**
Credits (L:T:P)	**4 (3:1:0)**	SEE Marks	**30**
Hours	**45L + 15T + 0P = 60**	SEE Mode	**Theory**

Course Objectives: students will be able to	
1	To understand modern performance metrics, pipeline design, and their trade-offs.
2	To be proficient in RISC-V ISA, pipeline behaviour, and simulation.
3	To analyze and design memory subsystems, interconnects, coherence models, and parallel architectures.
4	To evaluate domain-specific architectures and accelerators for ML/DL, and apply RISC-V vector extensions in real-world problem

Module - 1	**9 hours**
Flynn's taxonomy, an overview of SISD, SIMD, MISD and MIMD architectures. Performance metrics, CPI, IPC, Amdahl's Law, Power-performance trade-offs, Instruction set design principles, Introduction to RISC-V ISA and comparison with legacy ISAs (x86, ARM), RISC-V base ISA, pipeline stages, pipeline structures, pipeline registers. RISC-V pipeline structure.	

Module - 2	**9 hours**
Basic pipeline architecture and hazards (data, control, structural), Forwarding, hazard resolution, branch prediction, Superscalar and dynamic scheduling (Tomasulo's algorithm), MIPS and RISC-V pipeline modelling (5-stage and extended designs).	

Module - 3	**9 hours**
Cache design (associativity, write policies, inclusion), Memory hierarchy performance, virtual memory, Interconnects: buses, crossbars, NoCs, Coherence and consistency: MESI and Directory based protocols.	

Module - 4	**9 hours**
Multicore processors, Chip Multiprocessor (CMP), SMT, Memory consistency models, Introduction to RISC-V-based SoCs and integration into multicore platforms, Synchronization, parallel memory, and shared resources.	

Module - 5	**9 hours**
GPU architecture overview and comparison with CPUs, Tensor Processing Units (TPU) and deep learning accelerators, FPGA-based acceleration. Overview of vector extensions in RISC-V (RVV), and AI accelerators in the RISC-V ecosystem, CUDA programming.	

Course Outcomes: After completing the course, the students will be able to	
1	Analyze domain-specific architectures and accelerators for ML/DL, including RISC-V vector extensions in real-world applications
2	Apply knowledge of RISC-V, pipeline behavior, and simulation techniques to analyze processor performance.
3	Analyse memory subsystem designs, interconnects, cache coherence models, and parallel architectures.
4	Analyze domain-specific architectures and accelerators for ML/DL, including RISC-V vector extensions in real-world applications

Text Books	
1	John L Hennessy and David A Patterson, Computer Architecture a Quantitative Approach, MK, 2019, 6th ed. ISBN: 978-0128119051

Reference Books and Resources	
1	William Stallings., Computer Organization and Architecture Designing for Performance, Pearson, 2019, 11th ed, ISBN: 978-9332518704
2	David A. Patterson and John L. Hennessy, Computer Organizations and Design, The Hardware/Software Interface: RISC-V Edition, MK, 2017, ISBN: 978-0128122754
3	John Cheng, Max Grossman and Ty McKercher, Professional CUDA C Programming, Wiley Publisher, 2014, ISBN: 978-1118739327

Tutorials	**15 Hours**
Part A: Tutorials	
1	a) Implement a Bubble Sort algorithm in RISC-V assembly language using the RISC-V Online Simulator and analyze the program by counting the number of comparisons and swaps while evaluating its instruction-level efficiency. b) Implement a Traffic Signal Controller using RISC-V Online Simulator
2	Analyze how increasing the issue width of a superscalar RISC-V processor affects Instruction-Level Parallelism (ILP) and overall execution performance for arithmetic- intensive workloads using Web-Based Simulator of Superscalar RISC-V Processors https://arxiv.org/html/2411.07721v1
3	Analyze the movement of instructions through the IF, ID, EX, MEM, and WB stages using the University of Michigan pipeline simulator https://vhosts.eecs.umich.edu/370simulators/pipeline/simulator.html
4	Simulate data hazards using the online simulator of University of Michigan https://vhosts.eecs.umich.edu/370simulators/pipeline/simulator.html
5	Analyze the effect of cache size variation on cache hit rate and memory access performance using the CPU-OS Simulator. https://teach-sim.com/downloads/
6	Analyze the execution of a CUDA kernel launched with 2 blocks and 8 threads per block by observing thread and block identifiers using Google Colab with the NVIDIA T4 GPU.

Tutorials (continued)	
7	Compare the execution time of vector addition on CPU and GPU using CUDA programming in Google Colab with the NVIDIA T4 GPU and analyze the performance improvement achieved through parallel execution
8	Implement a CUDA program for Dot Product computation using the parallel reduction method in Google Colab and analyze the performance, thread utilization, and reduction efficiency achieved through parallel execution.
9	Implement a CUDA program to compute Cosine Similarity between two vectors using parallel reduction techniques in Google Colab with the NVIDIA T4 GPU and analyze execution time, parallel efficiency, and GPU resource utilization
10	Analyze how GPU parallelism accelerates machine learning operations such as similarity computation and gradient updates using CUDA programming in Google Colab with the NVIDIA T4 GPU

Part B:
Design and analyze a high-performance computing framework integrating pipelined and superscalar RISC-V architectures, cache optimization, branch prediction, and CUDA-based GPU acceleration for machine learning workloads, and evaluate system performance in terms of execution time, CPI, instruction-level parallelism, memory efficiency, and throughput.

Semester - **II**			
Course: **ADVANCED OPERATING SYSTEMS**			
Program	**M.Tech CSE**	Category	**Major (Core)**
Course Code	**CS5101**	CIE Marks	**70**
Credits (L:T:P)	**4 (3:0:2)**	SEE Marks	**30**
Hours	**45L + 0T + 30P = 75**	SEE Mode	**Practical**

Course Objectives: students will be able to	
1	To understand advanced concepts, structures, and functionalities of modern operating systems.
2	To analyze process management, multithreading, and design considerations for multiprocessor and multicore systems.
3	To explore distributed operating system mechanisms such as deadlock detection, resource sharing, and file management.
4	To gain practical experience through case studies like PintOS, focusing on thread and virtual memory management.

Module - 1	**9 hours**
Modern Operating Systems, OS structures -monolithic, layered, microkernel, modular, system calls and OS services, virtual machines and hypervisors, container concepts, Micro kernel architecture, OS design considerations for multiprocessors and multicore, interrupts and system boot process, Microsoft Windows overview, Linux, Linux Virtual Machine Architecture.	

Module - 2	**9 hours**
Process Description and Control - Process States, description and control, execution of OS, Security issues. Threads –Processes and threads, types of threads, Multicore and Multithreading, Windows Threads and SMP Management, Linux Process and Thread Management.	

Module - 3	**9 hours**
Deadlock handling strategies in distributed systems, issues in deadlock detection and resolution, centralized deadlock detection algorithms, distributed deadlock detection algorithms, hierarchical deadlock detection algorithms.	

Module - 4	**9 hours**
Distributed file systems: Architecture, mechanisms for building distributed file systems, design issues, Log-structured file systems. Distributed shared memory: Architecture and motivation, algorithms for implementing DSM, memory coherence, coherence protocols, design issues.	

Module - 5	**9 hours**
Structures of multiprocessor operating systems, operating system design issues, threads, process synchronization, process scheduling, memory management, reliability/fault tolerance, Case study: PintOS- Threads and Virtual memory.	

Course Outcomes: After completing the course, the students will be able to	
1	Examine the architecture, objectives, and critical functionalities of modern operating systems.
2	Evaluate and compare algorithms for deadlock detection, resource management, and multiprocessor scheduling.
3	Design multi-process and multithreaded mechanisms ensuring memory coherence and deadlock resolution.
4	Demonstrate the working of process concurrency, distributed file systems, and shared memory models for applications across heterogeneous operating systems.

Text Books	
1	Operating Systems: Internals and Design Principles, William Stallings, 9 th Edition, Pearson Education, 2018, ISBN 13: 978-0-13-230998-1.
2	Advanced concepts in operating systems, Mukesh Singhal, Niranjan G Shivarathri, Tata Mcgraw Hill Education Pvt. Ltd, 2017, ISBN: 978-0070472686.

Reference Books and Resources	
1	Operating Systems, Gary Nutt, Nabendu Chaki, Sarmistha Neogy, 3 rd Edition, Pearson Education, 2017, ISBN 0201773449.
2	https://web.stanford.edu/class/cs140/projects/pintos/pintos.pdf

Lab Programs / Practical	**30 Hours**
Part A: Lab Programs	
1	Create child processes using fork(), exec(), and wait() system calls.
2	Demonstrate orphan and zombie process creation.
3	Implement communication between parent and child processes using pipes.
4	Create two processes that exchange messages using System V message queues.
5	Thread Creation using POSIX Threads
6	Implement the Wait-for Graph (WFG) method to detect deadlocks among processes.
7	Implement and analyze a C program for the Banker's Algorithm to evaluate safe and unsafe system states in resource allocation.
8	Develop a C program in which two processes communicate using shared memory synchronized through semaphores to demonstrate inter-process communication and process synchronization concepts.
9	Implement producer–consumer synchronization using threads and semaphores.
10	Demonstrate thread synchronization on shared data using mutex locks.
11	Implement FIFO and LRU page replacement algorithms.
Part B:	

Lab Programs / Practical (continued)
Project Title :AI-Assisted Intelligent System Monitor and Process Analyzer Design and develop an AI-powered Operating System monitoring tool that analyzes system processes, CPU scheduling behavior, memory usage, deadlock situations, and thread execution in a Linux environment. The system should integrate modern AI tools for log analysis, anomaly detection, and automated reporting.

Semester - **II**			
Course: **RESEARCH METHODOLOGY**			
Program	**M.Tech CSE**	Category	**Major (Core)**
Course Code	**CS5900**	CIE Marks	**70**
Credits (L:T:P)	**2 (2:0:0)**	SEE Marks	**30**
Hours	**30L + 0T + 0P = 30**	SEE Mode	**Theory**

Course Objectives: students will be able to	
1	Introduce the fundamental principles, purpose, and process of research in academic and applied domains
2	Develop the ability to identify research problems, formulate hypotheses, and design appropriate research methodologies.
3	Equip students with skills in measurement, sampling, data analysis, and interpretation for making informed research decisions.
4	Familiarize students with research tools, academic databases, and ethical standards for scholarly communication and publication.

Module - 1	**6 hours**
Meaning, Objectives, Motivation, Utility. Concept of theory, empiricism, deductive and inductive theory. Characteristics of scientific method – Understanding the language of research – Concept, Construct, Definition, Variable. Research Process. Problem Identification & Formulation – Research Question – Investigation Question – Measurement Issues – Hypothesis – Qualities of a good Hypothesis –Null Hypothesis & Alternative Hypothesis. Hypothesis Testing – Logic & Importance.	

Module - 2	**6 hours**
Concept and Importance of Research Design, Features of a Good Research Design, Exploratory Research Design – Concept, Types, Uses, Descriptive Research Design – Concept, Types, Uses, Experimental Design – Independent and Dependent Variables, Qualitative Research – Concept and Features, Quantitative Research – Concept and Features, Concept of Measurement, Causality, Generalization, Replication, Merging Qualitative and Quantitative Approaches.	

Module - 3	**6 hours**
Concept of Measurement, What is Measured, Problems in Measurement, Validity and Reliability, Levels of Measurement – Nominal, Ordinal, Interval, Ratio, Statistical Population, Sample, Sampling Frame, Sampling Error, Sample Size, Non-Response, Characteristics of a Good Sample, Probability Sampling – Simple Random Sampling, Systematic Sampling, Stratified Random Sampling, Multi-stage Sampling, Determining Sample Size, Practical Considerations in Sampling.	

Module - 4	**6 hours**

Data Preparation, Univariate Analysis – Frequency Tables, Bar Charts, Pie Charts, Percentages, Bivariate Analysis – Cross Tabulations, Chi-square Test, Testing Hypothesis of Association, Interpretation of Data, Layout of a Research Paper, Systematic literature survey, Journals in Computer Science, Impact Factor of Journals, When and Where to Publish, Ethical Issues in Publishing, Plagiarism, Self-Plagiarism.

Module - 5	**6 hours**
Use of Encyclopedias, Research Guides, Handbooks, Academic Databases in Computer Science, Methods to Search Required Information Effectively, Reference Management Software – Zotero, Mendeley, Software for Paper Formatting – LaTeX, MS Word, Software for Detection of Plagiarism.	

Course Outcomes: After completing the course, the students will be able to	
1	Analyze foundational concepts, research types, and processes, including the formulation and testing of hypotheses.
2	Apply appropriate research designs and distinguish between qualitative and quantitative approaches for conducting effective research.
3	Demonstrate knowledge of measurement techniques, sampling methods, and statistical tools for data collection and analysis.
4	Utilize academic resources, digital tools, and ethical practices for preparing and publishing scholarly research.

Text Books	
1	Cooper, D. R., & Schindler, P. S., Business research methods (9th ed.)., 2011, McGraw-Hill Education. ISBN: 9780073373706
2	Bryman, A., & Bell, E.,Business research methods (3rd ed.)., 2011, Oxford University Press.ISBN: 9780199583409
3	Kothari, C. R.,Research methodology: Methods and techniques (2nd ed.)., 2004, New Age International Publishers. ISBN: 9788122415223

Reference Books and Resources	
1	Kumar, R., Research methodology: A step-by-step guide for beginners (4th ed.). , 2014, SAGE Publications.ISBN-10. 9789351501336
2	Yin, R. K., Case study research and applications: Design and methods (6th ed.). , 2018, SAGE Publications. ISBN-10.1506336167
3	Creswell, J. W., Research design: Qualitative, quantitative, and mixed methods approaches (4th ed.). , 2014, SAGE Publications. ISBN-10. 1452226105
4	Zikmund, W. G., Babin, B. J., Carr, J. C., & Griffin, M., Business research methods (9th ed.)., 2013, Cengage Learning.ISBN-13: 978-9353503260

Semester - **II**			
Course: **DEEP LEARNING**			
Program	**M.Tech CSE**	Category	**Major (Elective)**
Course Code	**CS5204**	CIE Marks	**70**
Credits (L:T:P)	**3 (2:0:2)**	SEE Marks	**30**
Hours	**30L + 0T + 30P = 60**	SEE Mode	**Theory**

Course Objectives: students will be able to	
1	To introduce the foundational concepts of neural networks and their application in classification tasks.
2	To explore advanced deep learning architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for solving computer vision and sequence modeling problems.
3	To provide hands-on experience in building, training, and evaluating deep learning models using real-world datasets.
4	To develop an understanding of explainable AI (XAI) techniques for interpreting and visualizing deep learning model predictions.

Module - 1	**6 hours**
Deep learning-Deep neural network- Building your Deep Neural Network: Step by Step, Deep Neural Network for Image Classification. Hyper parameter tuning, Initialization, Regularization and Optimization.	

Module - 2	**6 hours**
Introduction to Convolutional Neural Networks, Layers in CNN, A typical CNN structure, Types of convolutions, Depthwise Separable, Dilated, and Pointwise convolutions, Standard CNN models, Mobile net and transfer learning, 3D CNNs and Spatiotemporal Learning, CNNs in multimodal systems.	

Module - 3	**6 hours**
Introduction to Recurrent Neural Networks and their structure, Challenges in RNN (Vanishing and Exploding Gradients), Long-short term memory (LSTM), Gated Recurrent Unit (GRU), Attention mechanism in RNNs, and Transformers.	

Module - 4	**6 hours**
Autoencoder (Types: Linear, CNN-based), Training Autoencoders, Variational Autoencoders (VAE), Introduction to GAN, GAN Architecture (Generator, Discriminator), Applications, Variants of GAN, Diffusion models.	

Module - 5	**6 hours**
Model compression techniques: Pruning, Quantization, Knowledge Distillation, ONNX and model conversion formats for deployment, LIME, SHAP, Grad-CAM, Smooth-Grad, Occlusion, Saliency Maps, PCI.	

Course Outcomes: After completing the course, the students will be able to	
1	Apply fundamental deep learning techniques to design and train neural network models for various tasks.
2	Implement convolutional and sequence-based architectures to solve real-world image and sequential data problems.
3	Utilize generative and unsupervised learning approaches for data representation and synthesis.
4	Examine model compression and interpretability methods to enhance deployment efficiency and explain model decisions.

Text Books	
1	Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT Press, 2016, ISBN : 9780262035613.
2	Zhang, Aston, et al. "Dive into deep learning." Cambridge University Press, 2023, ISBN- 13 : 9781009389433.

Reference Books and Resources	
1	CS231n: Deep Learning for Computer Vision, Stanford University.
2	CS6910: Deep Learning, IIT Madras.
3	Practical Deep Learning, Fast.ai (https://course.fast.ai/)

Lab Programs / Practical	**30 Hours**
Part A: Lab Programs	
1	Implementing a Shallow Neural Network for Classification
2	Deep Neural Network for Image Classification
3	Hyperparameter Tuning and Regularization Techniques
4	Implementing a Basic CNN for Image Recognition and transfer learning
5	Implementing CNN for multimodal systems
6	Implementing RNN, LSTM, and GRU for real time problem.
7	Implementing Autoencoders and Variational Autoencoders (VAE)
8	Building a Generative Adversarial Network (GAN)
9	Model pruning and quantization for deployment optimization
10	Explainability techniques using LIME, SHAP, and Grad-CAM on CNN models.
Part B:	
A capstone project on building a deep learning system using modern AI tools to perform classification and object detection tasks, incorporating parameter tuning, regularization, transfer learning, and multimodal data handling. It also involves using AI tools for real-time and sequential data processing, data generation, deployment optimization through pruning and quantization, and applying explainability tools.	

Semester - **II**			
Course: **COMPUTER VISION**			
Program	**M.Tech CSE**	Category	**Major (Elective)**
Course Code	**CS5205**	CIE Marks	**70**
Credits (L:T:P)	**3 (2:0:2)**	SEE Marks	**30**
Hours	**30L + 0T + 30P = 60**	SEE Mode	**Theory**

Course Objectives: students will be able to	
1	To introduce the fundamental principles of digital image formation, geometric camera models, and classical image processing techniques for enhancement, filtering, and segmentation.
2	To provide a comprehensive understanding of foundational deep learning architectures, including Multilayer Perceptrons (MLPs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs), for vision tasks.
3	To explore key techniques for feature detection, description, and matching, and to establish the geometric relationships between multiple views of a scene using epipolar geometry.
4	To equip students with the ability to apply computer vision and deep learning models to solve complex, high-level tasks such as 3D scene reconstruction, semantic segmentation, and object detection.

Module - 1	**6 hours**
Introduction and Goals of Computer Vision and Image Processing, Image Formation Concepts, Radiometry, Geometric Transformations, Geometric Camera Models, Camera Calibration, Image Formation in a Stereo Vision Setup, Image Reconstruction from a Series of Projections,	

Module - 2	**6 hours**
Image Transforms, Image Enhancement, Image Filtering, Colour Image Processing, Image Segmentation, Texture Descriptors, Colour Features	

Module - 3	**6 hours**
Multilayer perceptron (MLP), Gradient descent, Backpropagation in MLP, Regularization and preprocessing, Convolutional neural network (CNN), CNN properties, CNN architectures, Introduction to recurrent neural network (RNN), Encoder-Decoder models in RNN	

Module - 4	**6 hours**
Low-level vision, Spatial and frequency domain filtering, Edge detection, Line detection, Feature detectors, Harris corner detector, Blob detection, SIFT, Feature descriptors, SURF, Single-view geometry, 2D Geometric transformations, Camera intrinsics and extrinsics.	

Module - 5	**6 hours**

Two-view stereo, Algebraic representation of epipolar geometry, Fundamental matrix computation, Structure from motion, Batch processing in SFM, Dense 3D reconstruction. Deepnets for stereo and SFM, Mid-level vision, Image segmentation, Deepnets for segmentation, High-level vision, Deepnets for object detection.

Course Outcomes: After completing the course, the students will be able to	
1	Analyze and apply fundamental image processing techniques such as spatial/frequency domain filtering, color transformations, and segmentation, and explain the principles of image formation and camera geometry.
2	Design and explain the architecture of foundational neural networks, including CNNs and RNNs, by understanding core concepts like backpropagation, convolutional layers, and encoder-decoder models.
3	Implement and compare various low-level feature detection and description algorithms (e.g., Harris corners, SIFT, SURF) to identify and match keypoints across multiple images.
4	Develop solutions for advanced vision tasks, such as Structure from Motion (SfM) and object detection, by integrating principles of multi-view geometry and applying modern deep learning models.

Text Books	
1	Richard Szeliski, "Computer Vision- Algorithms & Applications", Springer. (ISBN # 978-1848829343)
2	M.K. Bhuyan , " Computer Vision and Image Processing: Fundamentals and Applications", CRC Press, USA, (ISBN 9780815370840).
3	Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, 2016 (ISBN: 978-0262035613)

Reference Books and Resources	
1	Forsyth & Ponce, "Computer Vision-A Modern Approach", Pearson Education. (SBN-13: 978-9332550117)
2	Simon Prince, Computer Vision: Models, Learning, and Inference, 2012. (ISBN: 978-1107011793)
3	Bishop, Christopher M. Pattern Recognition and Machine Learning. Springer, 2006. ISBN 978-0-387-31073-2

Lab Programs / Practical	**30 Hours**
Part A: Lab Programs	
1	Reading, Displaying, and Rescaling Images and Videos using OpenCV. Implement fundamental image transformations such as translation, rotation, resizing and flipping (vertically, horizontally, or both).
2	Convert images between different color spaces like BGR, Grayscale, HSV, and LAB. Split an image into its individual color channels (Blue, Green, Red) and merge them back, including reconstructing individual color channels for visualization

Lab Programs / Practical (continued)	
3	Apply various blurring techniques such as averaging, Gaussian blur, median blur, and bilateral filtering to reduce noise in images. Explore edge detection algorithms like Canny, Laplacian, and Sobel to find boundaries and gradients within images
4	Use bitwise operators (AND, OR, XOR) to create and apply masks on binary images. Detect object contours using various modes and approximation methods, then visualize the results by drawing them.
5	Implement face detection in images using OpenCV's pre-trained Haar cascade classifiers.
6	Compute and visualize histograms for both grayscale and color images in OpenCV. Implement different types of thresholding to convert an image into a binary form, where pixels are either black (0) or white (255).
7	Build and train a basic image classification model using TensorFlow. This includes loading a dataset (e.g., MNIST), preparing data (normalization, reshaping, potentially one-hot encoding), building a neural network architecture (Sequential, Functional, or Model subclassing), compiling the model with an optimizer, loss function, and metrics, and finally training and evaluating the model
8	Perform classification and segmentation of real world image using Deep Learning Technique.

Part B:

Students will architect an integrated vision pipeline for concurrent 2D object detection and 3D spatial reconstruction, optimized for edge-cloud collaborative environments. The project requires implementing model compression techniques and MLOps workflows to deploy high-fidelity geometric models on resource-constrained hardware, while analytically balancing reconstruction accuracy against inference latency and power consumption.

Semester - **II**			
Course: **MLOPS**			
Program	**M.Tech CSE**	Category	**Major (Elective)**
Course Code	**CS5206**	CIE Marks	**70**
Credits (L:T:P)	**3 (2:0:2)**	SEE Marks	**30**
Hours	**30L + 0T + 30P = 60**	SEE Mode	**Theory**

Course Objectives: students will be able to	
1	To provide a comprehensive overview of MLOps, highlighting its significance, challenges, workflows, and tools in comparison to traditional software development and DevOps practices.
2	To introduce data engineering tools, ML pipeline automation, and CI/CD practices essential for managing and deploying machine learning models
3	To equip students with practical skills for deploying machine learning models on cloud platforms.
4	To equip students with the skills to monitor, scale, and govern ML and LLM systems in production using explainability, bias tracking, and performance optimization strategies.

Module - 1	**6 hours**
Overview of MLOps and its significance: Key challenges in deploying and managing ML models in production, Classical Software Engineering Techniques, Agile Software Engineering Technique, Comparison of traditional software development, DevOps and MLOps, Key components of MLOps, MLOps workflow, Landscape of MLOps tools and technologies.	

Module - 2	**8 hours**
Overview of data engineering tools and practices, Data management for ML models, ML Flow, ML pipeline automation. DVC overview. CI/CD for ML Models: Use of version control systems like Git for model development, automated testing & validation, model delivery strategies.	

Module - 3	**3 hours**
Deploying in cloud platform: Introduction, Configuration, Deploying a model to ML cloud platform, Make Predictions, Switching Models, Remove a deployed model.	

Module - 4	**7 hours**
Monitoring and Performance Optimization: Techniques for monitoring model performance in production, Logging and error tracking for ML systems, Model Performance Monitoring using Kube Flow, Performance optimization and scaling strategies. ML Governance: Overview of model explainability and ethical considerations in ML deployments, tracking bias.	

Module - 5	**6 hours**
Introduction to LLM, MLOps for LLM's, FMOps/LLMOps: Operationalize generative AI, LLM System Design, High-level view LLM-driven application, LLMOps Pipeline.	

Course Outcomes: After completing the course, the students will be able to	
1	Explain MLOps workflows, tools, and principles, and differentiate them from traditional DevOps practices to understand the key components for effective ML model deployment and monitoring.
2	Apply data versioning, pipeline automation, and CI/CD techniques to build scalable and reproducible ML workflows.
3	Analyze deployment strategies on cloud platforms to examine configurations that optimize ML model lifecycle management and prediction performance in production.
4	Evaluate and implement monitoring, governance, and scalability strategies for ML and LLM systems to ensure transparency and ethical deployment in production.

Text Books	
1	Introducing MLOps By Mark Treveil, Nicolas Omont, Clément Stenac, Kenji Lefevre, Du Phan, Joachim Zentici, Adrien Lavoillotte, Makoto Miyazaki, Lynn Heidmann, O'Reilly Publisher, 2020. ISBN: 978-1098116439
2	Beginning MLOps with MLFlow: Deploy Models in AWS SageMaker, Google Cloud, and Microsoft Azure, By Sridhar Alla, Suman Kalyan Adari, Apress, 2020. ISBN: 1484284348
3	LLMs in Production: From Language Models to Successful Products By Christopher Brousseau, Matt Sharp, Manning Publisher, 2025, ISBN:9781633437203, 1633437205

Reference Books and Resources	
1	Engineering MLOps Rapidly Build, Test, and Manage Production-Ready Machine Learning Life Cycles at Scale by Emmanuel Raj, Packt Publisher, 2020. ISBN-10 : 1803237325

Lab Programs / Practical	**30 Hours**
Part A: Lab Programs	
1	Building a Minimal MLOps Pipeline: Workflow and Tool Exploration
2	Experiment with Data Engineering Tools for Machine Learning
3	Automating ML Pipelines with DVC and CI/CD Workflows
4	Deploy an ML model in cloud platform.
5	Monitoring and Optimizing ML Models in Production Environments
6	ML Governance and Model Explainability: Ensuring Ethical and Transparent AI
7	Model Versioning and CI/CD Strategies for Base and Fine-Tuned ML Models
8	Balancing Performance, Cost, and Governance in ML Deployments
9	Demonstrate LLM applications using open-source transformer models
10	End-to-End DevOps Pipeline using Git, GitHub, GitHub Actions, Docker, Docker Hub and Kubernetes.

Lab Programs / Practical (continued)
Part B:
Design and develop an end-to-end MLOps framework that automates the machine learning lifecycle from data ingestion to model deployment and monitoring. The project should implement a complete ML pipeline including data preprocessing, model training, evaluation, version control, continuous integration/continuous deployment (CI/CD), and cloud-based deployment platforms.

Semester - **II**			
Course: **STATISTICAL METHODS FOR DATA SCIENCE**			
Program	**M.Tech CSE**	Category	**Major (Elective)**
Course Code	**CS5207**	CIE Marks	**70**
Credits (L:T:P)	**3 (2:0:2)**	SEE Marks	**30**
Hours	**30L + 0T + 30P = 60**	SEE Mode	**Theory**

Course Objectives: students will be able to	
1	To introduce foundational statistical concepts used in modern data science.
2	To develop the ability to apply statistical inference and estimation to real-world datasets.
3	To equip students with practical skills in modelling, analysing, and interpreting data using statistical programming.
4	To foster critical thinking for data-driven decision-making under uncertainty.

Module - 1	**6 hours**
Types of data, statistical graphics: histograms, scatterplots, boxplots, Summary statistics: mean, median, variance, skewness, Set theory, conditional probability, Bayes' Theorem	

Module - 2	**6 hours**
Discrete and continuous distributions: Bernoulli, Binomial, Poisson, Normal Expectation, variance, transformations, Sampling distributions and the Central Limit Theorem.	

Module - 3	**6 hours**
Parameter estimation: point and interval Maximum Likelihood Estimation, Confidence intervals for mean, proportions, and differences.	

Module - 4	**6 hours**
z-test, t-test, chi-square, ANOVA Errors, significance level, power of a test, Real-world applications: A/B testing.	

Module - 5	**6 hours**
Linear regression: estimation, assumptions, diagnostics Logistic regression for classification, Model interpretation and variable selection (R2, AIC, BIC).	

Course Outcomes: After completing the course, the students will be able to	
1	Apply statistical methods to fit and interpret linear and logistic regression models, and analyse model diagnostics for decision-making.
2	Utilise statistical programming tools to analyse datasets, generate insights, and evaluate their implications for real-world decisions.
3	Analyse and compare statistical models using metrics such as R2, AIC, and BIC, and select the most suitable model for a given problem.

Course Outcomes: After completing the course, the students will be able to (continued)	
4	Design and conduct hypothesis tests, interpret statistical evidence, and justify conclusions in real-world contexts.

Text Books	
1	Wasserman, L. (2004). All of statistics: A concise course in statistical inference. Springer. ISBN-10: 0387402721
2	James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning: With applications in R. Springer. (ISLR Book Website) ISBN-10: 1461471370

Reference Books and Resources	
1	Casella, G., & Berger, R. L. (2002). Statistical inference (2nd ed.). Duxbury Press. ISBN-10: 0534243126
2	Montgomery, D. C., & Runger, G. C. (2014). Applied statistics and probability for engineers (6th ed.). Wiley. ISBN-10: 1118539710
3	Rice, J. A. (2006). Mathematical statistics and data analysis (3rd ed.). Cengage Learning. ISBN-10: 0534399428

Lab Programs / Practical	**30 Hours**
Part A: Lab Programs	
1	Write a program to load the Diabetes dataset (from sklearn. datasets or a CSV file) into a panda DataFrame. a)Display the shape of the dataset, Data types of each column, Summary statistics using. describe () and Count of missing values (if any). b)Identify and list: Numerical columns and Categorical columns. c)If any missing values are found, write a short code to impute them (for example, using mean or median for numerical data). Identify which columns may need scaling before applying machine learning models.
2	Write a program to compute and visualize central tendency, spread; plot histograms, boxplots, pair plots. Report skewness and kurtosis for bmi and target from the Diabetes dataset. Apply log-transform if needed and show plots before/after.
3	Create a program to simulate Binomial, Normal, Poisson samples and overlay with feature distributions (e.g., target approx Normal?). Hint: Use stats.normaltest() to test normality of target.
4	Write a program to compute point estimates and confidence intervals for mean blood pressure (bp) and for proportion (Pima: proportion with diabetes). Compare normal vs t-based CI when n small (simulate n=10 sampling). Demonstrate numeric CI intervals and interpretation.
5	Write a program to compute effect size (Cohen's d) for the t-test. Test whether mean bmi differs between male/female (or diabetics vs non-diabetics). Perform chi-square on categorical associations (e.g., categorical age bins vs diabetes). Demonstrate t-statistic, p-value, chi2, p-value and interpretation in context.

Lab Programs / Practical (continued)	
6	Create a program to demonstrate CLT by bootstrapping sample means of target and plotting their distribution for increasing sample sizes. Repeat for n=5, n=50 and compare variances. Demonstrate approximately normal sampling distribution; empirical mean ˜ population mean.
7	Create a program to fit a multiple linear regression (e.g., predict target from bmi, bp, s1), check assumptions (linearity, normality of residuals, homoscedasticity), compute R2. Demonstrate summary table, coefficients, R2, QQ-plot, residual vs fitted plot to check heteroscedasticity. Apply Box-Cox transform to improve normality; compute VIF for multicollinearity.
8	Design a program by using Pima dataset to fit logistic regression, compute confusion matrix, accuracy, precision, recall, ROC and plot ROC curve. Demonstrate Confusion matrix, classification report, ROC curve and AUC. Use cross-validation and compute mean AUC; try regularization (C parameter) and observe changes.
9	Design a program to fit multiple regression models (nested sets of predictors) or logistic models, compare with AIC/BIC and test performance via cross-validation or ROC. Demonstrate AIC/BIC values, recommendation of best model balancing fit & parsimony. Exercises: Implement forward selection using AIC as criterion.

Part B:
Design and implement an end-to-end mini project: pipeline. Create a report. Build a complete analysis pipeline on one diabetes dataset: EDA → preprocessing → modelling → evaluation → short report & CSV outputs. Steps / Code sketch (structure) to follow: a. Load dataset and perform EDA. b. Preprocess (impute, scale with StandardScaler, encode if needed). c. Split data and fit chosen model(s) (linear or logistic). d. Evaluate using appropriate metrics (R2/AIC for regression; ROC/AUC/confusion for classification).

Semester - **II**			
Course: **PREDICTIVE AND PRESCRIPTIVE ANALYSIS**			
Program	**M.Tech CSE**	Category	**Major (Elective)**
Course Code	**CS5501**	CIE Marks	**70**
Credits (L:T:P)	**3 (2:0:2)**	SEE Marks	**30**
Hours	**30L + 0T + 30P = 60**	SEE Mode	**Theory**

Course Objectives: students will be able to	
1	Understand the fundamentals of predictive analytics, the CRISP-DM process, data roles, and statistical tools.
2	Learn how to prepare and preprocess data, handle missing values, and select features for predictive modeling.
3	To introduce the role and scope of prescriptive analytics in data-driven decision-making and solve optimization problems relevant to real-world scenarios.
4	To develop analytical skills using mathematical models and algorithms for decision optimization and perform simulation and heuristic-based prescriptive methods.

Module - 1	**6 hours**
Overview of Predictive Analytics - The CRISP-DM Process Model for Predictive Analysis - The role of data in Predictive Analysis - Data Understanding - Data Visualization - The Value of Statistical Significance - Statistical concepts and tools for Predictive Analysis	

Module - 2	**6 hours**
Understanding the importance of data quality for Predictive Analysis - Data Preparation - Data preprocessing - Dealing with missing data and outliers - Feature selection/creation techniques - Exploratory data analysis for predictive modelling.	

Module - 3	**6 hours**
Overview of Business Analytics: Descriptive, Predictive, and Prescriptive approaches. Importance of prescriptive analytics in decision-making systems. Fundamentals of Optimization: objective functions, decision variables, constraints. Linear Programming (LP) – formulation and graphical solution. Introduction to the Simplex Method and sensitivity analysis.	

Module - 4	**6 hours**
Integer and Nonlinear Programming: concepts and applications. Decision Analysis: decision trees, utility theory, and sensitivity under risk. Multi-objective and Goal Programming: balancing conflicting objectives. Network Models: transportation and assignment problems.	

Module - 5	**6 hours**
Introduction to Simulation modeling: Monte Carlo and discrete-event simulation. Heuristic and Meta-heuristic Optimization: Genetic Algorithms, Simulated Annealing (overview). Integration of predictive and prescriptive analytics for decision support. Implementation overview using Python tools ('PuLP', 'Pyomo', 'SciPy.optimize').	

Course Outcomes: After completing the course, the students will be able to	
1	Grasp the fundamental concepts of predictive analytics and its applications and understand the pre-processing of the data and importance of feature selection.
2	Formulate and solve simple linear programming and interpret optimization results.
3	Analyze and optimize real-world business problems involving multiple objectives and constraints.
4	Apply heuristic and simulation-based approaches for solving complex decision problems.

Text Books	
1	Dean Abbott, “Applied Predictive Analytics-Principles and Techniques for the Professional Data Analyst”, Wiley, 2014.
2	Daniel T. Larose, Chantal D. Larose, “Data Mining and Predictive Analytics”, Wiley, 2015.
3	James R. Evans, Business Analytics: Methods, Models, and Decisions, 3rd Edition, Pearson, 2017.
4	Frederick S. Hillier & Gerald J. Lieberman, Introduction to Operations Research, 11th Edition, McGraw-Hill, 2020.
5	NPTEL Link: https://onlinecourses.swayam2.ac.in/imb23_mg42/preview

Reference Books and Resources	
1	Michael J. Fry, Jeffrey D. Camm, James J. Cochran, Business Analytics, Cengage Learning, 2019.
2	Hamdy A. Taha, Operations Research: An Introduction, 10th Edition, Pearson, 2017.

Lab Programs / Practical	**30 Hours**
Part A: Lab Programs	
1	Exploratory Data Analysis and Visualization for Predictive Analytics
2	Data Preparation, Preprocessing, and Feature Engineering
3	Building and Evaluating Predictive Models using Regression Techniques
4	Linear Optimization using Python (PuLP / Pyomo)
5	Decision and Network Models for Optimization
6	Simulation and Heuristic Techniques for Prescriptive Analytics
Part B:	

Lab Programs / Practical (continued)
An organization wants to improve decision-making using Predictive and Prescriptive Analytics techniques with the support of Generative AI tools. Develop a solution that includes data understanding, visualization, preprocessing, feature selection, and predictive modelling to analyze business data and generate insights. Apply optimization, decision-making, simulation, and heuristic techniques to solve business problems and recommend optimal solutions. Use Generative AI tools to assist in data preparation, model development, visualization, optimization, and result interpretation. Finally, explain how predictive and prescriptive analytics can support intelligent business decision-making systems.

Semester - **II**			
Course: **DATA PRIVACY, SECURITY AND ETHICS**			
Program	**M.Tech CSE**	Category	**Major (Elective)**
Course Code	**CS5400**	CIE Marks	**70**
Credits (L:T:P)	**3 (2:0:2)**	SEE Marks	**30**
Hours	**30L + 0T + 30P = 60**	SEE Mode	**Theory**

Course Objectives: students will be able to	
1	To introduce the foundational concepts of information security, privacy principles, and threat modelling applicable to AI and data-driven systems.
2	To equip students with privacy-preserving techniques such as anonymization, differential privacy, federated learning, and privacy-by-design frameworks, while addressing relevant data protection laws and compliance requirements.
3	To analyse adversarial threats and secure AI model deployment techniques, including defense strategies, encrypted inference, secure APIs, and intellectual property protection mechanisms.
4	To promote ethical and responsible AI practices, ensuring fairness, bias mitigation, transparency, accountability, and regulatory compliance through governance and monitoring frameworks.

Module - 1	**6 hours**
Fundamentals of Information Security: CIA Triad (Confidentiality, Integrity, Availability), Data as an Asset: Ownership, Classification, and Valuation, Privacy Principles in the Context of Big Data and AI, Threat Modelling in Data Systems: Adversarial Models, Insider Threats, Attack Vectors in ML Systems: Model Inversion, Membership Inference, Privacy vs. Utility Trade-offs in Data Science, Introduction to Ethical and Responsible AI, Ethics in the Age of AI, Use Cases across Industries and the need for RAI, Responsible AI Lifecycle, RAI framework, Fairness, Data Principles - Data Privacy & Security.	

Module - 2	**6 hours**
Privacy and Security in AI, Data Privacy in AI, Techniques and strategies: anonymization, differential privacy, Regulatory context: GDPR, HIPAA, India DPDP Act (brief), Privacy by Design (PbD), Federated Learning for Privacy Preservation, Architecture and use cases in healthcare & sustainability, Communication efficiency and data locality challenges. Data security using Split Learning.	

Module - 3	**6 hours**
Adversarial Attacks on AI Models, Evasion attacks (FGSM, PGD), data poisoning, model inversion, Threat surfaces in deployed AI systems, Defensive Mechanisms for Robust AI, Adversarial Training, Defensive Distillation, Randomized Smoothing, Secure Model Serving and Inference, Homomorphic Encryption for encrypted inference, Model Watermarking for IP protection, Model Stealing and Ownership Verification, Secure APIs and Authentication in ML Services, Role-based access control, OAuth, API tokens, Securing inference endpoints and CI/CD pipelines.	

Module - 4	**6 hours**
Understanding Bias in AI Learning, Fairness in Data & Model, Bias in AI Learning, Types of Biases in AI, Mitigating Bias in AI, Mitigating Bias in AI learning, Techniques and strategies for Bias Measurement, Biased AI and Their Consequences, AI Fairness 360, TensorFlow Constrained Optimisation, what if I tool.	

Module - 5	**6 hours**
Accountability in AI, Types of Drift, Drift Detection, Ensuring Accountability and Governance, Laws and regulations impacting AI: GDPR and DPDP, AI Model Monitoring Tools, build a monitoring dashboard using MLflow or Evidently AI.	

Course Outcomes: After completing the course, the students will be able to	
1	Apply foundational concepts of data security and privacy to assess potential risks and balance the trade-offs between information protection and usability.
2	Analyse privacy-preserving approaches in data processing and model development to evaluate their effectiveness in meeting legal and ethical compliance requirements.
3	Develop strategies to safeguard systems against various security threats, ensuring secure and reliable deployment of intelligent applications.
4	Demonstrate the fairness, accountability, and transparency in data-driven systems by integrating ethical principles and monitoring mechanisms throughout the system lifecycle.

Text Books	
1	Coeckelbergh, Mark. AI ethics. Mit Press, 2020, ISBN: 9780262538190.
2	Dubber, Markus Dirk, Frank Pasquale, and Sunit Das, eds. The Oxford handbook of ethics of AI. Oxford Handbooks, 2020, ISBN: 9780190067427.

Reference Books and Resources	
1	Voeneky, Silja, et al., eds. The Cambridge handbook of responsible artificial intelligence: Interdisciplinary perspectives. Cambridge University Press, 2022, ISBN: 9781009207898
2	Dan Hendrycks, Introduction to AI Safety, Ethics, and Society, CRC Press,2024, ISBN 9781032869926
3	Christoph, Molnar. "Interpretable machine learning: A guide for making black box models explainable." (2020), ISBN: 0244768528.

Lab Programs / Practical	**30 Hours**
Part A: Lab Programs	
1	Implementing the CIA Triad in a Data Storage System
2	Simulating Model Inversion and Membership Inference Attacks
3	Data Anonymization and De-identification Techniques
4	Implementing Differential Privacy in Model Training

Lab Programs / Practical (continued)	
5	Federated Learning for Health care use case
6	Adversarial Attacks and Defense Techniques
7	Encrypted Inference with Homomorphic Encryption
8	Bias Detection and Fairness Evaluation in AI Models
9	AI Model Monitoring and Governance Dashboard
Part B:	
Develop a secure, privacy-preserving, and responsible data analytics solution using generative AI tools. Apply AI-based techniques for data analysis, visualization, preprocessing, security, fairness, and monitoring. Implement suitable privacy protection, bias mitigation, and AI governance mechanisms for trustworthy analytics.	

Programmes:

1. B.Tech. Honors(CSE)
2. B.Sc. Honors(CSE)
3. B.C.A. Honors(CSE)
4. M.Tech. (CSE)
5. M.Tech. Data Science (in collaboration with upGrad Campus) for working professionals
6. Ph.D.

*Proposed infrastructure under construction

RV University Vision and Mission

Vision

To be a world-class, tech-driven, global university for liberal education, empowering citizens of tomorrow.

Mission

- Strive for excellence in teaching, research, capacity building, and community engagement, benchmarking against global universities to lead across disciplines.
- Utilise digital and emerging technologies to enhance teaching-learning and research, accessible to all, while fostering a multidisciplinary, inclusive environment that meets evolving learner needs.
- Cultivate a diverse, global academic community through strong national and international collaborations that enrich learning, facilitate mobility, and drive institutional growth.
- Integrate theory with practical application to develop self-driven, empathetic problem-solvers equipped to create meaningful societal impact.

RV UNIVERSITY
Go, change the world
प्रज्ञा: ★ धीरा:
an initiative of RV EDUCATIONAL INSTITUTIONS

RV Vidyanikethan Post, 8th Mile, Mysuru Road, Bengaluru – 560 059
For general enquiry: enquiry@rvu.edu.in | Phone: +91 95136 73778

RV University | _RVUniversity_ | rv-university | RV.University1 | rv.university