Scheme of Examinations and Syllabi for M.Sc. in Computer Science (Data Science & Machine Learning) Program

(Based on Curriculum and Credit Framework for PG Programs under NEP-2020)

With Effect from the Session 2024-25

MAHARSHI DAYANAND UNIVERSITY ROHTAK (HARYANA)

Structure for 2 year M.Sc. in Computer Science (Data Science & Machine Learning) Program w.e.f. the Session 2024-25

	Semester	Discipline-Specific Courses (DSC)	Skill Enhancement Courses (SEC) / Vocational Courses (VOC)/ Internship	Research thesis/project	Total Credits
	First year of 2 Year PG program (NHEQF Level 6)				
	I	Statistics for Data Science @ 4 credits Introduction to Data Science @ 4 credits Information Retrieval and Data Mining @ 4 credits Machine Learning and Python Programming @ 4 credits Artificial and Computational Intelligence @ 4 credits	Data Structures using Java / MERN Stack Development/Internship 1 @ 4 credits	---	24
	II	Big Data Analytics and R Programming @ 4 credits Computer Vision and Image Processing @ 4 credits Cloud Computing &IoT @ 4 credits Deep Learning @ 4 credits Graph Algorithms and Mining @ 4 credits	Advanced Machine Learning and Generative AI /Full Stack Development /Internship 2 @ 4 credits	---	24
Students who exit after first year on completion of 48 credits will be awarded PG Diploma in concerned discipline					
Second year of two year PG program (NHEQF Level 6.5) (STUDENT SHOULD SELECT ANY ONE OPTION FOR THE SECOND YEAR OF 2 YEAR PG PROGRAM)					
Only Course Work					
Option 1	III	Predictive Analytics for IoT@ 4 credits Data Visualization and Interpretation @ 4 credits Natural Language Processing @ 4 credits Reinforcement Learning @ 4 credits Information Security and Privacy @ 4 credits	Internship 3/ Project Work 1 @ 4 credits	---	24
	IV	Blockchain Technology @ 4	Internship 4/ Project Work 2 @ 4 credits	---	24

		credits			
		AR/VR Systems and Wearable Computing @ 4 credits			
		Social Networking analysis @ 4 credits			
		Decision Sciences @ 4 credits			
		High Dimensional Data @ 4 credits			
Course work and Research					
Option 2	**III**	**Predictive Analytics for IoT @ 4 credits**	**Internship 3/ Project Work 1 @ 4 credits**	**---**	**24**
		Deep Visualization and Interpretation @ 4 credits			
		Natural Language Processing @ 4 credits			
		Reinforcement Learning @ 4 credits			
		Information Security and Privacy @ 4 credits			
	IV	**--**	**Research Methodology/ Internship 4 @ 4 credits**	**Research thesis @20 credits**	**24**

Only Research (only for the students who have completed 3 Years Bachelor's Program)					
	Semester	**Discipline-Specific Courses (DSC)**	**Skill Enhancement Courses (SEC) / Vocational Courses (VOC)/ Internship**	**Research thesis/project**	**Total Credits**
Option 3	**III**	--	**Research Methodology /Internship 3 @ 4 credits**	**20 credits***	**24**
	IV	--	**Relevance of Research in Data Science /Internship 4 @ 4 credits**	**20 credits****	**24**

Note:

*****The students who opted Option 3 should submit a project report/synopsis of at least 50 pages comprising of Literature survey, identification of Research Problem, Plan of work, methodology as well as practical work (if any) at the end of 3rd semester and the same will be evaluated by internal and external examiners.

**The students should continue the research work in 4^{th}semester based on the project work/synopsis submitted at the end of 3^{rd}semester. The final thesis/project report will be evaluated by the internal and external examiners.

Structure for 1 year Post Graduate Programme (2nd year of M.Sc. in Computer Science (Data Science & Machine Learning) Program)

	Semester	Discipline-Specific Courses (DSC)	Skill Enhancement Courses (SEC) / Vocational Courses (VOC)/Internship	Dissertation/ Project work	Total Credits
(STUDENT SHOULD SELECT ANY ONE OPTION)					
Only Course Work					
Option 1	I (Semester III of 2 year PG Program)	Predictive Analytics for IoT @ 4 credits	Internship 3 @ 4 credits	---	24
		Data Visualization and Interpretation @ 4 credits			
		Natural Language Processing @ 4 credits			
		Reinforcement Learning@ 4 credits			
		Information Security and Privacy @ 4 credits			
	II (Semester III of 2 year PG Program)	Blockchain Technology @ 4 credits	Internship 4 @ 4 credits	---	24
		AR/VR Systems and Wearable Computing @ 4 credits			
		Social Networking Analysis @ 4 credits			
		Decision Sciences @ 4 credits			
		High Dimensional Data @ 4 credits			
Course work and Research					
Option 2	I (Semester III of 2 year PG Program)	Predictive Analytics for IoT @ 4 credits	Internship 3 @ 4 credits	---	24
		Data Visualization and Interpretation @ 4 credits			
		Natural Language Processing @ 4 credits			
		Reinforcement Learning @ 4 credits			
		Information Security and Privacy @ 4 credits			
	II	--	Relevance of Research	Research	24

	(Semester III of 2 year PG Program)		**in Data Science /Internship 4 @ 4 credits**	**Thesis @ 20 credits**	

Type of Course			Credits Distribution			Total Credits	Workload			Total Workload	Marks			
	Nomenclature of Course	Course Code	L	T	P		L	T	P		Theory		Practical	Total Marks
											Internal	External		
Semester I (Session 2024-25)														
DSC1 @ 4 credits	Statistics for Data Science	24CSM201DS01	4	0	0	4	4	0	0	4	30	70	-	100
DSC2 @ 4 credits	Introduction to Data Science	24CSM201DS02	4	0	0	4	4	0	0	4	30	70	-	100
DSC3 @ 4 credits	Information Retrieval and Data Mining	24CSM201DS03	3	0	1	4	3	0	2	5	25	50	25	100
DSC4 @ 4 credits	Machine Learning and Python Programming	24CSM201DS04	3	0	1	4	3	0	2	5	25	50	25	100
DSC5 @ 4 credits	Artificial and Computational Intelligence	24CSM201DS05	3	0	1	4	3	0	2	5	25	50	25	100
SEC1 / VOC1 / Internship 1 @ 4 credits	Data Structures using Java	24CSM201SE01	2	0	2	4	2	0	4	6	---	---	---	100
	OR	OR												
	Full Stack Development-I	24CSM201MV01	2	0	2	4	2	0	4	6	15	35	50	100
	OR	OR												
	Internship- I	24CSM201IN01	0	0	4	4	0	0	8	8	30	70	-	100
Semester II (Session 2024-25)														
DSC 6 @ 4 credits	Big Data Analytics and R Programming	24CSM202DS01	3	0	1	4	3	0	2	5	25	50	25	100
DSC 7 @ 4 credits	Computer Vision and Image Processing	24CSM202DS02	3	0	1	4	3	0	2	5	25	50	25	100
DSC 8 @ 4 credits	Cloud Computing &IoT	24CSM202DS03	3	0	1	4	3	0	2	5	25	50	25	100
DSC 9 @ 4 credits	Deep Learning	24CSM202DS04	3	0	1	4	3	0	2	5	25	50	25	100
DSC 10 @ 4 credits	Graph Algorithms & Mining	24CSM202DS05	3	0	1	4	3	0	2	5	25	50	25	100
SEC2 / VOC2 / Internship 2@ 4 credits	Advanced Machine Learning and Generative AI	24CSM202SE01	2	0	2	4	2	0	4	6	---	---	---	100
	OR	OR												
	Full Stack Development-II	24CSM202MV01	2	0	2	4	2	0	4	6	15	35	50	100
	OR	OR												
	Internship- II	24CSM202IN01	0	0	4	4	0	0	8	8	30	70	-	100
Type of Course			**Credits**			**Total**	**Workload**			**Total**	**Marks**			

			Distribution			Credits				Workload				
	Nomenclature of Course	**Course Code**	**L**	**T**	**P**		**L**	**T**	**P**		**Theory**		**Practical**	**Total Marks**
											Internal	**External**		
OPTION – I (ONLY COURSE WORK)														
Semester III (Session 2025-26)														
DSC 11 @ 4 credits	Predictive Analytics for IoT	25CSM203DS01	4	0	0	4	4	0	0	4	30	70	-	100
DSC 12 @ 4 credits	Data Visualization and Interpretation	25CSM203DS02	2	0	2	4	2	0	4	6	15	35	50	100
DSC 13 @ 4 credits	Natural Language Processing	25CSM203DS03	2	0	2	4	2	0	4	6	15	35	50	100
DSC 14 @ 4 credits	Reinforcement Learning	25CSM203DS04	4	0	0	4	4	0	0	4	30	70	-	100
DSC 15 @ 4 credits	Information Security and Privacy	25CSM203DS05	2	0	2	4	2	0	4	6	15	35	50	100
SEC3/ Internship 3 / Project Work 1@ 4 credits	Internship –III	25CSM203IN01	0	0	4	4	0	0	8	8	30	70	-	100
	OR	OR												
	Project Work-I	25CSM203PD01	0	0	4	4	0	0	8	8	30	70	-	100
Semester IV (Session 2025-26)														
DSC 16 @ 4 credits	Blockchain Technology	25CSM204DS01	2	0	2	4	2	0	4	6	15	35	50	100
DSC 17 @ 4 credits	AR/VR Systems and Wearable Computing	25CSM204DS02	2	0	2	4	2	0	4	6	15	35	50	100
DSC18 @ 4 credits	Social Network Analysis	25CSM204DS03	4	0	0	4	4	0	0	4	30	70	-	100
DSC19 @ 4 credits	Decision Sciences	25CSM204DS04	3	0	1	4	3	0	2	5	25	50	25	100
DSC20 @ 4 credits	High Dimensional Data	25CSM204DS05	4	0	0	4	4	0	0	4	30	70	-	100
SEC4/Internship 4/ Project Work 2 @ 4 credits	Internship-IV	25CSM204IN01	0	0	4	4	0	0	8	8	30	70	-	100
	OR	OR												
	Project Work-II	25CSM204PD01	0	0	4	4	0	0	8	8	30	70	-	100
OPTION – II (COURSE WORK AND RESEARCH)														
Semester III (Session 2025-26)														
DSC 11 @ 4 credits	Predictive Analytics for IoT	25CSM203DS01	4	0	0	4	4	0	0	4	30	70	-	100
DSC 12 @ 4 credits	Data Visualization and Interpretation	25CSM203DS02	2	0	2	4	2	0	4	6	15	35	50	100
DSC 13 @ 4 credits	Natural Language Processing	25CSM203DS03	2	0	2	4	2	0	4	6	15	35	50	100
DSC 14 @ 4 credits	Reinforcement Learning	25CSM203DS04	4	0	0	4	4	0	0	4	30	70	-	100
DSC 15 @ 4 credits	Information Security and Privacy	25CSM203DS05	2	0	2	4	2	0	4	6	15	35	50	100

SEC3/ Internship 3 / Project Work 1@ 4 credits	Internship –III	25CSM203IN01	0	0	4	4	0	0	8	8	30	70	-	100
	OR	OR												
	Project Work 1	25CSM203PD01	0	0	4	4	0	0	8	8	30	70	-	100
Semester IV (Session 2025-26)														
SEC4/ Internship 4 @ 4 credits	Research Methodology	25CSM204SE01	4	0	0	4	4	0	0	4	---	---	---	100
	OR	OR												
	Internship –IV	25CSM204IN01	0	0	4	4	0	0	8	8	30	70	-	100
Research thesis/ project @20 credits	Research Thesis	25CSM204PD01	-	-	-	20	-	-	-	-	-	-	500	500
OPTION -III(ONLY RESEARCH) (ONLY FOR THE STUDENTS WHO HAVE COMPLETED 3 YEARS BACHELOR'S PROGRAM)														
Semester III (Session 2025-26)														
Skill Enhancement Courses (SEC) / Vocational Courses (VOC)/ Internship @ 4 Credits	Research Methodology	25CSM203SE01	4	0	0	4	4	0	0	4	---	---	---	100
	OR	OR												
	Internship III	25CSM203IN01	0	0	4	4	0	0	8	8	30	70	-	100
Research thesis/ Project @20 credits	Research Thesis-I / Project-I	25CSM203PD01	-	-	-	20	-	-	-	-	-	-	500	500
Semester IV (Session 2025-26)														
Skill Enhancement Courses (SEC) / Vocational Courses (VOC)/ Internship	Relevance of Research in Data Science	25CSM204SE01	4	0	0	4	4	0	0	4	---	---	---	100
	OR	OR												
	Internship IV	25CSM204IN01	0	0	4	4	0	0	8	8	30	70	-	100
Research thesis/ project @20 credits	Research Thesis-II/ Project-II	25CSM204PD01	-	-	-	20	-	-	-	-	-	-	500	500

L: Lecture; T: Tutorial; P: Practical

Internship Evaluation

After completion of internship, students need to prepare a comprehensive report highlighting their learning and takeaways during the internship period as per MDUR Internship Regulations 2025. The report shall be signed by the Internship Supervisor from respective UTD/Centre/College and Mentor from internship providing organizations. Evaluation of internship report and viva-voce will be jointly conducted by Internship Supervisor and Mentor on the time and date notified by the concerned HoDs/Directors/Principals. The mentor from host organization may participate in the evaluation through online/offline mode. In case of non-availability of respective mentor, the available relevant mentor as decided by the concerned HOD/Director/Principal may be utilized for the purpose of evaluation.

Suggested distribution of marks will be as below:

<table>
<tr><th>S. No.</th><th>Components</th><th>Employability-Oriented Internship</th><th>Research-Oriented Internship</th></tr>
<tr><td>1</td><td>Assessment by Mentor
<table>
<tr><td>S.No.</td><td>Details</td><td>Marks</td></tr>
<tr><td>1</td><td>Skills learned</td><td>15</td></tr>
<tr><td>2</td><td>Regularity</td><td>10</td></tr>
<tr><td>3</td><td>Conduct</td><td>5</td></tr>
<tr><td></td><td>Total (30)</td><td></td></tr>
</table></td><td>30</td><td>30</td></tr>
<tr><td>2</td><td>Internship Report</td><td>40</td><td>40</td></tr>
<tr><td>3</td><td>Viva-Voce</td><td>30</td><td>30</td></tr>
</table>

EVALUATION OF PROJECT REPORT / DISSERTATION

In case of the Project reports/Dissertation/Research Project, the assessment shall be jointly carried out by the internal and external examiners. There shall be no Internal assessment component for Dissertation / Project Report. External examiners shall be invited from amongst the panel of examiners (ordinarily not below the rank of Associate Professor) recommended by the concerned Board of Studies.

Note : -

1. **The skill enhancement courses should contain maximum of component and w.e.f. the Academic Session 2026 – 2027, the pattern of Skill Enhancement Courses be revised as follows by the UTDS/Centres/Institutes in Scheme and Syllabi of UG/PG programs under NEP - 2020:**

For Four Credit SEC	For Three Credit SEC	For Two Credit SEC
L – T – P	L – T – P	L – T – P
0 – 0 – 4	0 – 0 – 3	0 – 0 – 2
1 – 0 – 3	1 – 0 – 2	----
2 – 0 – 2	---	----

2. The details of Formative Assessment in hard copy to submitted to Controller of Examinations within a week of uploading the marks on the portal and the relevant records alongwith the documents on the basis of which marks has been awarded and shall be maintained for a minimum period of two years in the Department/Centre/Institute/ affiliated colleges. The records may be verified/summoned by the Controller of Examinations.

ProgramLearningOutcomes of M.Sc. in Computer Science (Data Science & Machine Learning) Program

ThegraduateoncompletionofM.Sc. in Computer Science (Data Science & Machine Learning) programwillbeableto:

PLO1	Acquire advanced knowledge about Data Science and Machine Learning concepts with latestdevelopment and issues relating to the domain area.
PLO2	Master the principles, methods, and techniques applicable to the chosen field of study with anin-depth understanding.
PLO3	Utilize the acquired knowledge and skills to extrapolate and apply to solve real-life situationsthat may be known, new and even unfamiliar ones.
PLO4	Integrate and Implement conceptual, operational, or technical knowledge with a range ofcognitive and practical skills required to solve higher order problems in the chosen field oflearning.
PLO5	Conduct and synthesize research to carry out investigations as individuals or in team thatrequire higher order creative and cognitive skills to formulate evidence-based solutions in thedomain of Data Science and Machine Learning.
PLO6	Communicate and present research findings in a well-structured manner to showcase theresults and inferences, specifically in the fields of Computer Science, Data Science, andMachine Learning.
PLO7	Pursue self-paced and self-directed learning to continuously upgrade knowledge and skills,including research-related skills, while adhering to the ethical best practices for developing a life-long learning.

Syllabi for M.Sc. in Computer Science (Data Science & Machine Learning) Program

Semester:First
Session: 2024-2025

Name of Program	**M.Sc in Computer Science (Data Science & Machine Learning)**	**Program Code**	**-----------**
Name of the Course	**Statistics for Data Science**	**Course Code**	**24CSM201DS01**
Hours per Week	**4+0+0**	**Credits**	**4:0:0**
Maximum Marks	**Theory: 100(70+30)**	**Time of Examinations**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objective:
The objective of this course is to provide the learners with the skills necessary to analyze and interpret categorical and numerical data, understand the relationships between variables, and apply probabilistic concepts to discrete and continuous random variables.

Course Outcomes:
By the end of the course the students will be able to:

CO1: Identify and classify different types of data, scales of measurement, and the distinction between descriptive and inferential statistics.
CO2: Construct and interpret frequency distributions for categorical and numerical data and effectively use various measures of central tendency and dispersion to summarize data.
CO3: Analyze the association between two variables and learn about the implications of these associations in data analysis.
CO4: Apply the basic principles of counting, including permutations and combinationsto solve problems involving the addition and multiplication rules of counting.
CO5: Understand and apply probability concepts to real-world scenarios and data sets.

Unit-I

Introduction and type of data, Types of data, Descriptive and Inferential statistics, Scales of measurement, Describing categorical data.

Frequency distribution of categorical data, Best practices for graphing categorical data, Mode and median for categorical variable, Describing numerical data Frequency tables for numerical data.

Unit-II

Measures of central tendency - Mean, median and mode, Quartiles and percentiles, Measures of dispersion - Range, variance, standard deviation and IQR, Five number summary.

Association between two variables - Association between two categorical variables - Using relative frequencies in contingency tables, Association between two numerical variables - Scatterplot, covariance, Pearson correlation coefficient, Point bi-serial correlation coefficient.

Unit-III

Basic principles of counting and factorial concepts - Addition rule of counting, Multiplication rule of counting, Factorials, Permutations and combinations, Probability Basic definitions of probability, Events.

Properties of probability, Conditional probability - Multiplication rule, Independence, Law of total probability, Bayes' theorem, Random Variables - Random experiment, sample space and random variable, Discrete and continuous random variables.

Unit-IV

Probability mass function, Cumulative density function, Expectation and Variance - Expectation of a discrete random variable, Variance and standard deviation of a discrete random variable, Binomial and Poisson random variables - Bernoulli trials, Independent and identically distributed random variable, Binomial random variable, Expectation and variance of abinomial random variable, Poisson distribution, Introduction to continuous random variables - Area under the curve, Properties of pdf, Uniform distribution, Exponential distribution.

Suggested Readings:

1. Bruce, Peter, Andrew Bruce, and Peter Gedeck: Statistics for Data Scientists: 50+ Essential Concepts Using R and Python, O'Reilly Media.
2. Liu, Chung Laung: Elements of Discrete Mathematics, Tata McGraw-Hill Education.
3. Heumann, Christian, Schomaker, Michael, Shalabh: Introduction to Statistics and Data Analysis With Exercises, Solutions and Applications in R, Springer.
4. Douglas C. Montgomery, George C. Runger: Applied Statistics and Probability for Engineers, Wiley (Low price edition)
5. Robert V. Hogg. Allen T. Craig: Introduction to Mathematics and Statistics, Pearson Education.
6. Richard A. Johnson, Irwin Miller, John Freund: Probability and Statistics for Engineers.
7. Irwin Miller, Marylees Miller: Mathematical Statistics with Applications, Pearson Education.
8. Pierre Lafaye de Micheaux, RémyDrouilhet, Benoit Liquet: The R Software-Fundamentals of Programming and Statistical Analysis, Springer.
9. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

Name of the Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	**-----**
Name of the Course	**Introduction to Data Science**	**Course Code**	**24CSM201DS02**
Hours/Week	**4+0+0**	**Credits (L:T:P)**	**4:0:0**
Max. Marks.	**Theory: 100 (70+30)**	**Time of Examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objective:
The objective of this course is to provide Learner with comprehensive knowledge to grasp the fundamentals of data science and explore the role of a data scientist to understand statistical inference like population vs. samples, data preparation, exploratory data analysis techniques, and machine learning algorithms to explore data visualization principles.

Course Outcomes:
By the end of the course the students will be able to:

CO1: Apply data science processes to an e-commerce data and demonstrate the use of estimation methods for analyzing this data.
CO2: Compare and apply appropriate machine learning algorithms for classification.
CO3: Compare and choose one data visualization method for effective visualization of data.
CO4: Design a model of recommendation system based on the content of the data.
CO5: Apply standard clustering methods to analyze social network graph.

Unit-I

Introduction to Data Science: What is Data Science, importance of data science, Big data and data Science, The current Scenario, Industry Perspective Types of Data: Structured vs. Unstructured Data, Quantitative vs. Categorical Data, Big Data vs. Little Data, Data science process, Role of Data Scientist.

Case Studies (if any): Ecommerce Marketplace.

Unit-II

Statistical Interference and Exploratory Data Analysis: Introduction-Population and samples, Data Preparation, Exploratory Data Analysis-Summarizing Data, Data Distribution, Outlier Treatment, Measuring Symmetry, Continuous Distribution, Kernel Density, Estimation: Sample and Estimated Mean, Variance and Standard Scores, Covariance, and Pearson’s and Spearman’s Rank Correlation.

Machine Learning Algorithms: Linear Regression, K-nearest Neighbors (K-NN), K-mean, Spam Filters, Naïve Bayes, and Wrangling: Naive Bayes, Comparing Naive Bayes to k-NN, Scraping the Web: APIs and Other Tools.

Unit-III

Data Visualisation: Introduction, Types of data visualisation, Data for visualisation: Data types, Data encodings, Retinal variables, Mapping variables to encodings, Visual encodings.

Recommendation Systems: A Model for Recommendation Systems - The Utility Matrix, The Long Tail, Applications of Recommendation Systems, Populating the Utility Matrix; Content-Based Recommendations: Item Profiles, Discovering Features of Documents, Obtaining Item Features From Tags, Representing Item Profiles, User Profiles, Recommending Items to Users Based on Content; Collaborative Filtering: Measuring Similarity, The Duality of Similarity, Clustering Users and Items, Evaluation of Recommendation System.

Case Studies (if any): Movie Lens Case Study.

Unit-IV

Social Network Analysis: Social Networks as Graphs, Varieties of Social Networks, Graphs With Several Node Types, Clustering of Social-Network Graphs: Distance Measures for Social-Network Graphs, Applying Standard Clustering Methods, Betweenness, The Girvan-Newman Algorithm, Using Betweenness to Find Communities.

Case Studies (if any): Community detection in social network.

Suggested Readings:

1. Russell S. and Norvig P.: Artificial Intelligence: A Modern Approach, Prentice-Hall.
2. Cathy O'Neil and Rachel Schutt: Doing Data Science, Straight Talk From The Frontline, O'Reilly.
3. Jure Leskovek, AnandRajaraman and Jeffrey Ullman: Mining of Massive Datasets, Cambridge University Press.
4. Laura Igual and SantiSegui: Introduction to Data Science: A Python Approach to Concepts, Techniques and Applications, Springer.
5. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

Name of the Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	**-----**
Name of the Course	**Information Retrieval and Data Mining**	**Course Code**	**24CSM201DS03**
Hours/Week	**3+0+2**	**Credits (L:T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25**	**Time of Examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objective:
The objective of this course is to provide the learner with comprehensive knowledge of information retrieval and data mining techniques to grasp and explore system capabilities and use various data structures for searching and exploring large amount of data.

Course Outcomes:
By the end of the course the students will be able to:

CO1: Understand the basic concepts of the information retrieval.
CO2: Analyse the involvement of the information retrieval in modern life style & social media.
CO3: Apply data pre-processing, indexing, retrieval methods and concepts.
CO4: Evaluate the effectiveness and efficiency of different information retrieval systems
CO5: Understand the basic concepts of Data Warehouse and analysis and mining of data.

Unit-I

Introduction to Information Retrieval Systems: Definition and Objectives of Information Retrieval Systems, Functional Overview, Relationship to Database Management Systems, Digital Libraries and Data Warehouses. **Information Retrieval System Capabilities:** Search Capabilities, Browse Capabilities, Miscellaneous Capabilities Cataloguing and Indexing: History and Objectives of Indexing, Indexing Process, Automatic Indexing, Information Extraction.

Data Structure: Introduction to Data Structure, Stemming Algorithms, Inverted File Structure, N-Gram Data Structures, PAT Data Structure, Signature File Structure, Hypertext and XML Data Structures, Hidden Markov Models.

Unit-II

Automatic Indexing: Classes of Automatic Indexing, Statistical Indexing, Natural Language, Concept Indexing, Hypertext Linkages.Document and Term Clustering: Introduction to Clustering, Thesaurus Generation, Item Clustering, Hierarchy of Clusters.

User Search Techniques: Search Statements and Binding, Similarity Measures and Ranking, Relevance Feedback, Selective Dissemination of Information Search, Weighted Searches of Boolean Systems, Searching the Internet and Hypertext.

Unit-III

Information Visualization: Introduction to Information Visualization, Cognition and Perception, Information Visualization Technologies.Text Search Algorithms: Introduction to Text Search Techniques, Software Text Search Algorithms, Hardware Text Search Systems.

MultimediaInformation Retrieval: Spoken Language Audio Retrieval, Non-Speech Audio Retrieval, Graph Retrieval, Imagery Retrieval, Video Retrieval.**Data Warehousing:** OLAP, Dimensional Modelling (facts, dimensions), cube, Schema, defining schema’s star schema, snow-flakes schema and fact constellation, ETL process.

Unit-IV

Overview of Data Mining: Concept of Data Mining, Association rules, Knowledge Discovery from Databases, Classification, and Clustering.**Classification methods:** Decision tree(ID3,C4.5,CART), Bayesian Classification, Rule based, Neural Network, Lazy and Eager Learners, Parameters for measuring Accuracy.

Data Mining Prediction methods: Linear and nonlinear regression, Logistic Regression Use of open source data mining tool – WEKA, XLMiner, MOA.

Suggested Readings:

1. Christopher D. Manning, PrabhakarRaghavan, and HinrichSchutze: Introduction to information retrieval, Cambridge University Press.
2. J. Han, M Kamber: Data Mining Concepts and Techniques, Morgan Kaufmann.
3. M. Dunham: Data Mining - Introductory and Advance Topics, Pearson Education.
4. F. Wilfrid Lancaster: Information Retrieval Systems: Characteristics, Testing and Evaluation, New York: Wiley.
5. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practical/Lab Assignments

1. Analyze and compare different Information Retrieval Systems and their functionalities.
2. Create manual and automatic indexes for a set of documents using different indexing techniques.
3. Implement basic data structures such as inverted file structure, N-gram data structures, and PAT data structure.
4. Implement statistical indexing and natural language indexing algorithms for a given set of documents.
5. Implement document and term clustering algorithms and analyze the hierarchy of clusters generated.
6. Evaluate search statements, similarity measures, and relevance feedback mechanisms using a test dataset.
7. Design and implement visualizations for different datasets using tools like D3.js or Matplotlib.
8. Implement software and hardware text search algorithms and compare their efficiency and effectiveness.
9. Implement audio, image, and video retrieval algorithms and evaluate their performance.
10. Design and implement a data warehouse schema, perform ETL processes, and apply data mining algorithms such as association rules and classification methods.
11. Given a collection of documents, implement an inverted file structure. Write a program that reads the documents, tokenizes the text, and builds an inverted index. Test your implementation by querying the index with sample search terms and returning the list of documents containing those terms.
12. Implement the use of a set of text documents and apply the K-Means clustering algorithm to group similar documents together.
13. Implement the term frequency-inverse document frequency (TF-IDF) for feature extraction.
14. Visualize the clusters and evaluate the clustering quality using metrics.
15. Implement two different stemming algorithms and compare their performance in terms of accuracy and efficiency on a sample dataset.
16. Design a star schema for a retail data warehouse that includes dimensions such as time, product, store, and customer, and a fact table for sales transactions.
17. Implement C4.5 decision tree algorithm and evaluate the model's performance.

Any other Programs/Lab Assignments given by the teachers.

Name of the Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	**-----**
Name of the Course	**Machine Learning and Python Programming**	**Course Code**	**24CSM201DS04**
Hours/Week	**3+0+2**	**Credits (L:T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25**	**Time of Examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objective:

The objective of this course is to provide the learners with comprehensive knowledge of fundamental machine learning concepts and model evaluation techniques to explore various methods of machine learning using case studies in the domain of Python programming language.

Course Outcomes:

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

CO1: Acquire fundamental knowledge of learning theory
CO2: Design and evaluate various machine learning algorithms
CO3: Use machine learning methods for multivariate data analysis in various scientific fields
CO4: Choose and apply appropriate Machine Learning Techniques for analysis, forecasting, categorization and clustering of the data
CO5: Using Python programming for various machine learning problems.

Unit-I

Machine Learning Concepts: Introduction to Machine Learning, Machine Learning applications, Types of learning: Supervised, Unsupervised and semi-supervised, reinforcement learning techniques, Models of Machine learning: Geometric model, Probabilistic Models, Logical Models, Grouping and grading models, Parametric and non-parametric models, Predictive and descriptive learning, Classification concepts, Binary and multi-class classification.

Learning Theory: Feature Extraction, Feature Construction and Transformation, Feature Selection, Dimensionality Reduction: Subset selection, the Curse of dimensionality, Principle Components analysis, Independent Component analysis, Factor analysis, Multidimensional scaling, Linear discriminant analysis, Bias/Variance tradeoff, Union and Chhernoff/ Hoeffding bounds, VC dimension, Probably Approximately Correct (PAC) learning, Concept learning, the hypothesis space, Least general generalization, Internal disjunction, Paths through the hypothesis space, model Evaluation and selection

Unit-II

Geometric Models: Regression, Logistic regression , Assessing performance of regression - Error measures, Overfitting, Least square method, Multivariate Linear regression, Regression for Classification, Perceptron, Multi-layer perceptron, Simple neural network, Kernel based methods, Support vector machines(SVM), Soft margin SVM, Support Vector Machines as a linear and non-linear classifier, Limitations of SVM, Concept of Relevance Vector, K-nearest neighbor algorithm.

Logical, Grouping and Grading Models: Decision Tree Representation, Alternative measures for selecting attributes, Decision tree algorithm: ID3, Minimum Description length decision trees, Ranking and probability estimation trees, Regression trees, Clustering trees, Rule learning for subgroup discovery, Association rule mining, Distance based clustering-K-means algorithm, Choosing number of clusters,Clustering around medoids – silhouettes, Hierarchical clustering, Ensemble methods: Bagging and Boosting.

Unit-III

Probabilistic Models: Uncertainty, Normal distribution and its geometric interpretations, Baye's theorem, Naïve Bayes Classifier, Bayesian network, Discriminative learning with maximum likelihood, Probabilistic models with hidden variables, Hidden Markov model, Expectation Maximization methods, Gaussian Mixtures and compression based models.

Case Studies on Advanced Machine Learning Techniques: Diagnosis of human disease, Diagnosis of crop disease, Text mining tasks like semantic analysis, author profiling, author identification, language identification, summarization etc., Prediction & forecasting, Fraud detection, Learning to rate vulnerabilities and predict exploits.

Unit-IV

Programming with Python: Introduction to Python and Computer Programming; Data Types, Variables, Basic Input-Output Operations, Basic Operators; Boolean Values, Conditional Execution, Loops, Lists and List Processing, Logical and Bitwise Operations; Functions, Tuples, Dictionaries, and Data Processing; Modules, Packages, String and List Methods, and Exceptions; The Object-Oriented Approach: Classes, Methods, Objects, and the Standard Objective Features; Exception Handling, and Working with Files.

Suggested Readings:

1. C.M. Bishop: Pattern Recognition and Machine learning, Springer.
2. Hastie, Tibshirani, Friedman: Introduction to statistical machine learning with applications in R, Springer.
3. Tom Mitchell: Machine Learning, McGraw Hill.
4. Parag Kulkarni: Reinforcement and Systemic Machine learning for Decision Making, Wiley-IEEE Press.
5. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practical/Lab Assignments

1. Implement a simple supervised learning algorithm like linear regression using Python libraries like scikit-learn.
2. Extract features from a dataset using techniques like one-hot encoding and normalization, and evaluate their impact on model performance.
3. Apply principal component analysis (PCA) to reduce the dimensionality of a dataset and visualize the results.
4. Train regression models to predict continuous values and logistic regression models for binary classification tasks.
5. Implement decision tree algorithms like ID3 and K-means clustering for a given dataset and evaluate their performance.
6. Implement SVM using libraries like scikit-learn for both linear and non-linear classification tasks and compare their performance.
7. Use the Naïve Baye's algorithm to classify text data into different categories, such as spam vs. non-spam emails.
8. Implement an HMM for speech recognition or part-of-speech tagging in natural language processing tasks.
9. Choose a case study such as fraud detection or disease diagnosis and develop a machine learning model to address the problem.
10. Implement basic Python programs covering data types, variables, loops, functions, and exception handling.
11. Consider a high-dimensional dataset and apply Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) to reduce the dimensionality.
12. Implement a Support Vector Machine (SVM) classifier using a Python library (e.g., scikit-learn) on a binary classification dataset.
13. Implement a Naïve Bayes classifier to perform sentiment analysis on a text dataset (e.g., movie reviews). Preprocess the text data (tokenization, stop-word removal, stemming) and evaluate the classifier's performance.
14. Write a Python program to load a dataset, perform basic data processing (e.g., handling missing values, normalization) and visualize the data using pandas, NumPy, and matplotlib libraries.
15. Develop a Python application containing classes for handling different types of data (e.g., text files, CSV files, JSON files). Implement methods for reading, processing, and writing data to files. Ensure proper exception handling to manage errors such as file not found, read/write errors, and data format issues.

Any other Programs/Lab Assignments given by the teachers.

Name of the Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	-----
Name of the Course	**Artificial and Computational Intelligence**	**Course Code**	**24CSM201DS05**
Hours/Week	**3+0+2**	**Credits (L:T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25**	**Time of Examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objective:
The objective of this course is to provide the learners with comprehensive knowledge and fundamentals of Artificial Intelligence and concepts of computational intelligence on real-world problems. It enables them to analyze fuzzy systems and neuro-fuzzy systems and gain relevant exposure so as to solve real world problems.

Course Outcomes:
By the end of the course the students will be able to:

CO1: Identify the need of Intelligent Agents in problem solving and learn the concept of artificial intelligence, problem solving and searching process.
CO2: Understand the concepts of knowledge and handling of uncertain and probabilistic knowledge
CO3: Design and analyze a learning technique for a given system in different AI application domains like marketing, healthcare, banking, finance, education, etc.
CO4: Learn the concepts of computational intelligence evolutionary computation and neural networks.
CO5: Handle the uncertainty in knowledge using fuzzy logic and understand concepts of fuzzy logic.

Unit-I

Introduction and Intelligent Agents: What is AI? Foundations History of Artificial Intelligence; State of the Art Intelligent Agents: Agents and Environments; Good Behaviour: Concept of Rationality, Nature of Environments, and Structure of Agents.
Case Studies (if any): Intelligent agents in autonomous systems.

Problem-solving: Solving Problems by Searching: Problem-Solving Agents, Uninformed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions, Beyond Classical Search Local Search Algorithms and Optimization Problems, Local Search in Continuous Spaces, Searching with Nondeterministic Actions, Searching with Partial Observations, Online Search Agents and Unknown Environments.
Case Studies (if any): Search techniques for a sliding tile problem.

Unit-II

Knowledge, Reasoning, and Planning: Knowledge based Agents, Types of knowledge, Knowledge acquisition and its techniques; Knowledge representation: Level of representation; First-Order Logic and its Inference.

Reasoning and Uncertain knowledge: Introduction and Types of reasoning, Quantifying Uncertainty, Probabilistic Reasoning, Probabilistic Reasoning over Time, Bayes Theorem in reasoning, Bayesian Belief Network, Making Simple Decisions, Making Complex Decisions.**Planning:** Components of a Planning system, Classical Planning, Planning and Acting in the Real World.
Case Studies (if any): Application of planning to a production system.

Unit-III

Computational Intelligence: Introduction to Computational Intelligence, Biological and Artificial Neural Network (ANN), artificial neural network models; learning in artificial neural networks; neural network and its applications.**Evolutionary Computing:** Fundamentals of evolutionary computation, Design and Analysis of Genetic Algorithms, Evolutionary Strategies, comparison of GA and traditional search methods.

Optimization: Genetic Operators and Parameters, Genetic Algorithms in Problem Solving; Optimization: Particle Swarm Optimization, Ant Colony Optimization, Artificial Immune Systems; Other Algorithms: Harmony Search, Honey-Bee Optimization, Memetic Algorithms, Co-Evolution, Multi-Objective Optimization, Tabu Search, Constraint Handling.

Unit-IV

Fuzzy Systems: Crisp sets, Fuzzy sets: Basic types and concepts, characteristics and significance of paradigm shift, Representation of fuzzy sets, Operations, membership functions, Classical relations and fuzzy relations, fuzzyfication, defuzzyfication, fuzzy reasoning, fuzzy inference systems, fuzzy control system, fuzzy clustering, applications of fuzzy systems. Neuro-fuzzy systems, Neuro-fuzzy modelling, Neuro-fuzzy control.

Case Studies: Credit card Fraud Analysis, Sentiment Analysis, Recommendation Systems and Collaborative filtering, Uber Alternative Routing.

Suggested Readings:

1. Russell S. and Norvig P.: Artificial Intelligence: A Modern Approach, Prentice-Hall.
2. Elaine Rich, Kevin Knight and Nair: Artificial Intelligence, TMH.
3. Luger G. F. and Stubblefield W. A.: Artificial Intelligence: Structures and strategies forComplex Problem Solving, Addison Wesley.
4. Nilsson Nils J.: Artificial Intelligence: A New Synthesis, Morgan Kaufmann Publishers Inc.
5. Patrick Henry Winston: Artificial Intelligence, Addison-Wesley Publishing Company.
6. M. Mitchell: An Introduction to Genetic Algorithms, Prentice-Hall.
7. J.S.R. Jang, C.T. Sun and E. Mizutani: Neuro-Fuzzy and Soft Computing, PHI, Pearson Education.
8. Davis E. Goldberg: Genetic Algorithms: Search, Optimization and Machine Learning, Addison Wesley.
9. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practical/Lab Assignments

1. Implement a simple intelligent agent that interacts with a simulated environment, demonstrating rational behavior.
2. Implement breadth-first search on a graph.
3. Implement the A* search on a sliding tile problem.
4. Implement first-order logic for representing knowledge and inference rules for reasoning.
5. Implement a basic feedforward neural network using libraries like TensorFlow on a dataset of your choice.
6. Implement a basic feedforward neural network using PyTorch and train it on a simple dataset of your choice.
7. Implement genetic algorithms (GA) for solving optimization problemsor scheduling problems.
8. Implement a fuzzy inference system for a simulated credit card dataset.
9. Implement a fuzzy inference system for implementing sentiment analysis on some hypothetical dataset.
10. Develop a neuro-fuzzy model for a specific application (neuro-fuzzy control) and interpret its performance.
11. Implement particle swarm optimization (PSO) or ant colony optimization (ACO) for solving optimization problems, such as feature selection or routing optimization.
12. Construct a Bayesian Belief Network for a medical diagnosis problem (e.g., diagnosing a specific disease based on symptoms). Implement probabilistic reasoning using Bayes' Theorem to infer the probability of the disease given certain symptoms. Validate the model with sample data and discuss the advantages and limitations of probabilistic reasoning in uncertain environments.
13. Implement an artificial neural network (ANN) to classify handwritten digits using the MNIST dataset. Train the neural network using backpropagation and evaluate its performance on the test set. Experiment with different network architectures (number of layers, neurons per layer) and activation functions to optimize classification accuracy.
14. Design and implement a fuzzy inference system for a credit scoring application. Define fuzzy sets, membership functions, and rules to evaluate the creditworthiness of applicants based on input variables (e.g., income, debt, credit history). Perform fuzzification, inference, and defuzzification to obtain a final credit score. Test the system with sample data and analyze its decision-making process.
15. Create a neuro-fuzzy model to perform sentiment analysis on a dataset of movie reviews. Combine neural networks with fuzzy logic to classify the sentiment of each review as positive, negative, or neutral. Train the model on a labeled dataset and evaluate its accuracy on a test set. Discuss the advantages of integrating neural networks and fuzzy logic for handling uncertain and imprecise data.

Any other Programs/Lab Assignments given by the teachers.

Name of the Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	-----
Name of the Course	**Data Structures using Java**	**Course Code**	**24CSM201SE01**
Hours/Week	**2+0+4**	**Credits (L:T:P)**	**2:0:2**
Max. Marks.	**Theory: 50(35+15)** **Practical: 50**	**Time of Examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:
This course provides the learners with the theoretical understanding and practical implementation skills necessary to design, analyze, and utilize various data structures efficiently using Java programming language.

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

CO1: Demonstrate proficiency in implementing and manipulating fundamental data structures in Java
CO2: Analyze the time and space complexities of algorithms and data structures
CO3: Develop the ability to utilize advanced Java features including iterators, generics, and polymorphism
CO4: Proficiently handle exceptions, assertions, and inner classes in Java.
CO5: Design and implement abstract data types and principles of data structures to solve real-world problems .

Unit-I

Java Basics: Types and Operators, Packages, Classes and Objects, Instance Variables, Methods, Constructors, The Heap.

Overriding Methods, Polymorphism, Iterators, Generics, Complexity Analysis, Time Complexity, Asymptotic Notations, Space Complexity.

Unit-II

Exception Handling, Assertions, Inner Classes, Static Members, Interfaces, The Class Hierarchy, Extending Classes, Type Checking and Casting, Super constructors.

Abstract Data Type, Collection of Elements, Organization, Foundations Partition ADT, Abstract Collection, Positional Collection ADT, Abstract Positional Collection.

Unit-III

Array, Circular Array, Dynamic Array and Dynamic Circular Array, Sorted Array.

Singly Linked List, Doubly Linked List, Queue ADT, Stack ADT.

Unit-IV

Abstract Search Tree Class, Binary Search Tree, Balanced Binary Search Trees, B-Tree, B+Tree, Trie, Compressed Trie.

Graph ADT, Abstract Graph and Graph Algorithms, Adjacency Matrix, Adjacency List, Weighted Graph ADT, Abstract Weighted Graph and Weighted Graph Algorithms.

Suggested Readings:

1. Sally. A Goldman, Kenneth. J Goldman, A Practical Guide to Data Structures and Algorithms using Java, CRC Press.
2. M. A. Weiss, Data Structures and Algorithm Analysis in Java, Pearson.
3. T. H. Cormen et al., Introduction to Algorithms, MIT Press.
4. R. Sedgewick and K. Wayne, Algorithms, Addison-Wesley.
5. R. Lafore, Java Data Structures and Algorithms, SAMS Publishing.
6. Levitin, Introduction to the Design and Analysis of Algorithms, Pearson.
7. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Program/Assignments

1. Program to demonstrate basic Java fundamentals such as types, operators, classes, objects, instance variables, methods, constructors, and memory allocation on the heap.
2. Program to illustrate polymorphism through method overriding and demonstrates the usage of generics.
3. Program to demonstrate exception handling using try-catch blocks and illustrates the usage of inner classes.
4. Program to demonstrate the usage of interfaces and illustrates the concept of abstract data types using collections.
5. Program to illustrate type checking, casting, and the usage of super constructors.
6. Program to demonstrate the usage of ArrayList to store and manipulate a list of integers.
7. Program to illustrate the usage of LinkedList to implement a queue.
8. Program to demonstrate the usage of HashSet to store unique elements.
9. Program to illustrate the usage of HashMap to store key-value pairs.
10. Program to demonstrate the usage of TreeSet to store elements in sorted order.
11. Program to illustrate the usage of PriorityQueue to implement a min heap.
12. Program to demonstrate the usage of Stack to implement a stack.
13. Program to illustrate the usage of LinkedHashMap to maintain insertion order.
14. Program to demonstrate the usage of ArrayDequeue to implement a dequeue.
15. Program to illustrate the usage of TreeMap to store elements in sorted order by keys.

Any other program by the teacher/instructor

Name of the Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	-----
Name of the Course	**Full Stack Development-I**	**Course Code**	**24CSM201MV01**
Hours/Week	**2+0+4**	**Credits (L:T:P)**	**2:0:2**
Max. Marks.	**Theory: 50 (35+15)** **Practical: 50**	**Time of Examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objective:

The objective of this course is to equip students with the skills and knowledge necessary to develop modern web applications using the MERN stack. Students will learn about each component of the stack - MongoDB, Express.js, React.js, and Node.js - and understand their roles in web development.

Course Outcomes:

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

CO1: Analyze complex requirements and design scalable web applications using the MERN stack.
CO2: Develop RESTful APIs using Express.js and integrate them with MongoDB for data storage.
CO3: Build interactive and responsive user interfaces using React.js and implement state management solutions.
CO4: Implement server-side logic and event-driven programming using Node.js.
CO5: Apply error handling and middleware techniques in Express.js applications.

Unit-I

The MERN stack: overview of MongoDB, Express.js, React.js, and Node.js. Role of each in modern web development.

Setting up development environment: Installation, configuration- tools and dependencies for MERN stack development.

Unit-II

Building a basic MERN application: Creating a simple web application from scratch using MongoDB for data storage, Express.js for server-side logic, React.js for client-side rendering, and Node.js for backend development.

Server-Side Development: Node.js and event-driven programming, asynchronous nature of Node.js, event-driven architecture.

Unit-III

Creating RESTful APIs with Express.js: Designing and implementing RESTful APIs using Express.js framework for handling HTTP requests and responses.

Data modelling with MongoDB and Mongoose: defining MongoDB schemas, overview of Mongoose ORM, models development using Mongoose ORM, Interacting MongoDB database.

Unit-IV

Express.js applications: Error handling- error handling middleware, middleware in Express.js- implementing and using middleware for request processing.

Client-Side Development: introduction to React.js, its components, component-based architecture – fundamentals of React.js library, component based user interface development.

Suggested Readings:

1. S. Subramanian, Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and

Node, Apress.
2. K. Chinnathambi, Learning React: A Hands-On Guide to Building Web Applications Using React and Redux, Addison-Wesley Professional.
3. M. Casciaro, Node.js Design Patterns: Master best practices to build modular and scalable server-side web applications, Packt Publishing.
4. E. Hahn, Express.js in Action, Manning Publications.
5. S. Bradshaw, E. Brazil, and K. Chodorow, MongoDB: The Definitive Guide, O'Reilly Media.
6. M. Riva, React Design Patterns and Best Practices: Build easy to scale modular applications using the most powerful components and design patterns, Packt Publishing.
7. Accomazzo, N. Murray, and A. Lerner, Fullstack React: The Complete Guide to ReactJS and Friends, Fullstack.io.
8. Dayley and B. Dayley, Node.js, MongoDB, and AngularJS Web Development, Addison-Wesley Professional.
9. Mead, Learning Node.js Development: Learn the fundamentals of Node.js, and deploy and test Node.js applications on the web, Packt Publishing.
10. S. Grubor, Deployment with Docker: Apply continuous integration models, deploy applications quicker, and scale at large by converting applications into Docker containers, Packt Publishing.
11. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practical/Lab Assignments

1. Create a new React project using create-react-app to render "Hello World" on the screen using Node.js and npm.
2. Implement a simple REST API routes for CRUD operations on MongoDB using Express.js and Mongoose.
3. Implement event handlers to update state of a form based on user input for a form component created using React.
4. Implement client-side routing in a React application using React Router.
5. Implement user authentication in a MERN stack application using JSON Web Tokens (JWT).
6. Demonstrate the ability to install and configure the MERN stack components and ensure they work together correctly.
7. Create a functional web application, demonstrating the integration of all four technologies.
8. Demonstrating the handling of HTTP requests and responses.
9. Enhance the "To-Do List" application by adding middleware for logging requests and handling errors. Implement custom error handling middleware to manage different types of errors (e.g., 404 Not Found, 500 Internal Server Error) and Implement middleware and error handling in Express.js applications to ensure robust and maintainable server-side functionality.
10. Develop a user interface for a "Movie Review" application using React.js. Implement components for displaying a list of movies, adding new reviews, and filtering movies by genre. Use state management techniques to manage the application state and objective is to Build interactive and responsive user interfaces using React.js, applying component-based architecture and state management solutions.

Any other Programs/Lab Assignments given by the teachers.

Semester: Second
Session: 2024-2025

Name of the Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	-----
Name of the Course	**Big Data Analytics and R Programming**	**Course Code**	**24CSM202DS01**
Hours/Week	**3+0+2**	**Credits (L:T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25**	**Time of Examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objective:
The objective of this course is to make the learners understand big data fundamentals and leverage big data processing technologies along with core Hadoop ecosystem and its tools for data transformation and analysis. It enables them to utilize R for big data manipulation, mastering the R interpreter, data structures, control flow, functions, and exploring data manipulation packages.

Course Outcomes:
By the end of the course the students will be able to:

CO1: Understand and design the data analytics and life cycle to be used for any real world problem.
CO2: Develop insights into the data and present results through visualization techniques.
CO3: Demonstrate the use of Hadoop and its ecosystem elements to analyze big data.
CO4: Demonstrate use of advanced FOSS computing environments for big health care data.
CO5: Exhibit skills in Hadoop, Yarn, Spark and R programming required for data analysis.

Unit-I

Basics of Big data: Characteristics, Types, Sources, Architectures, Data analysis process, Data analytics lifecycle, Pre-processing data, Market and Business Drivers for Big Data Analytics, Business Problems Suited to Big Data Analytics.

Case Studies (if any): Case study on data analytics lifecycle.

Unit-II

Technologies for big data analytics: Distributed and Parallel Computing for Big Data, Cloud Computing and Big Data, In-Memory Computing Technology for Big Data, Introduction to Hadoop, HDFS, MapReduce, YARN, HBase, Combining HDFS and HBase.

Case Studies (if any): Using MapReduce to scale algorithms for Big Data analytics.

Unit-III

Hadoop ecosystem: Sqoop, Impala, Apache Flume, Pig, Hive, Data transformation and analysis using Pig, Data analysis using Hive and Impala, Mahout, Oozie, Zookeeper etc.
Case Studies (if any): Sentiment analysis.

Big data analytics with Apache Spark: Apache Spark, Spark core, Interactive data analysis with spark shell, Writing a spark application, Spark RDD Optimization Techniques, Spark Algorithm, Spark SQL.

Unit-IV

R Programming: R interpreter, Introduction to major R data structures: Vectors, matrices, arrays, list and data frames. Control Structures, Vectorized if and Multiple selection.Functions.**Installing, loading and using packages:**Read/write data from/in files, Extracting data from web-sites: Clean data, Transform data by sorting, adding/removing new/existing columns, centering, scaling and normalizing the data values, converting types of values,String in-built functions.Big data analytics with RHadoop: R and Hadoop, Text mining in RHadoop, Designing GUI: Building interactive application and connecting it with database

Suggested Readings:

1. DT Editorial Services: Big Data, Black Book: Covers Hadoop 2, MapReduce, Hive, YARN, Pig, R and Data Visualization.
2. David Dietrich, Barry Hiller: Data Science and Big Data Analytics, EMC education services, Wiley Publications.
3. Mohammed Guller: Big Data Analytics with Spark: A Practitioner's Guide to Using Spark for Large Scale Data Analysis.
4. David Loshin: Big Data Analytics From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph, Morgan Kaufmann.
5. VenkatAnkam, "Big Data Analytics", Packt Publishing.
6. Jenny Kim, Benjamin Bengfort: Data Analytics with Hadoop, O'Reilly Media, Inc.
7. Glenn J. Myatt: Making Sense of Data: A Practical Guide to Exploratory Data Analysis and Data Mining.
8. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practical/Lab Assignments

1. Learn about vectors, matrices, arrays, lists, and data frames in R, and perform basic operations and manipulations on these data structures.
2. Write R code to implement control structures like if-else statements and loops, and create user-defined functions to perform specific tasks or calculations.
3. Load a dataset (e.g., CSV file) into R, perform summary statistics
4. Use R functions and packages like readr, readxl, and httr to import data from various file formats (e.g., CSV, Excel) and extract data from websites using web scraping techniques.
5. Create visualizations such as histograms, scatter plots and boxplots for the above dataset.
6. Perform data manipulation tasks such as sorting, adding/removing columns, converting data types, and applying string functions to clean and prepare data for analysis.
7. Implement the handling of missing values, remove duplicates, and perform data transformation tasks such as normalization and standardization using R functions and packages.
8. Analyze market trends and business metrics from a hypothetical market and business dataset to generate insights using statistical analysis and visualization techniques.
9. Use R packages tm and sentimentr to analyze sentiment in text data on customer reviews dataset and classify them as positive, negative, or neutral.
10. Use R packages tm and sentimentr to analyze sentiment in text data on social media comments and classify them as positive, negative, or neutral.
11. Use R packages like tm and tidytext to analyze text data, perform tasks such as tokenization, stemming, and sentiment analysis, and extract insights from large text datasets.
12. Use PySpark to perform distributed data processing tasks on large datasets, apply machine learning algorithms, and analyze data at scale.
13. Use RHadoop to interact with Hadoop ecosystem components such as HDFS, MapReduce, Hive, and Pig, and perform data analysis tasks using R programming.
14. Implement data mining algorithms using RHadoop, such as association rule mining, clustering, and classification, and analyze large datasets stored in Hadoop.
15. Explore Spark RDD optimization techniques such as caching, partitioning, and broadcast variables to improve the performance of data processing tasks in Apache Spark.

Any other Programs/Lab Assignments given by the teachers.

Name of the Program	M.Sc. in Computer Science (Data Science & Machine Learning)	Program Code	-----
Name of the Course	**Computer Vision and Image Processing**	**Course Code**	**24CSM202DS02**
Hours/Week	**3+0+2**	**Credits (L:T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25**	**Time of Examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:
The objective of this course is to provide the learners with comprehensive knowledge to understand the core concepts of computer vision and its distinction from image processing and gain fundamental image processing exposure needed to understand and identify shapes, regions and explore image enhancement techniques that can be useful in development of smart applications.

Course Outcomes:
By the end of the course the students will be able to:

CO1: Understand the fundamentals of computer vision that are required to build smart camera applications.
CO2: Understand the foundation of Image Processing and concepts of Image enhancement
CO3: Understand the concepts of Image restoration and feature extraction.
CO4: Know about segmentation and compression of image.
CO5: Develop computer image processing programs and applications using OpenCV-Python Libraries.

Unit-I

Computer Vision: Concepts of Computer vision, Computer vision vs Image Processing, Geometric techniques in Computer vision- Image transformations, Camera projections, Camera calibration, Depth from Stereo, Two view structure from motion, Object tracking.

Machine Learning for Computer Vision: Introduction to Machine learning, types of Machine Learning; Image Classification: Supervised and Unsupervised Image classification, Algorithms for Image classification; Object detection: Concepts of Object detection, Methods of Object detection; Image Segmentation: Classes of segmentation, Types of segmentation, Methods of segmentation.

Unit-II

Image Processing Foundations: Review of image processing techniques – classical filtering operations – thresholding techniques – edge detection techniques – corner and interest point detection – mathematical morphology – texture.

Shapes and Regions: Binary shape analysis – connectedness – object labelling and counting – size filtering – distance functions – skeletons and thinning – deformable shape analysis – boundary tracking procedures – active contours – shape models and shape recognition – centroidal profiles – handling occlusion – boundary length measures – boundary descriptors – chain codes – Fourier descriptors – region descriptors – moments.
Image Enhancement: Spatial and Frequency domain-Histogram processing-Spatial filtering-Smoothening spatial filters Sharpening spatial filters- Discrete Fourier Transform-Discrete Cosine Transform-Haar Transform - Hough Transform-Frequency filtering-Smoothening frequency filters-Sharpening frequency filters-Selective filtering.

Unit-III

Image Restoration & Image Registration: Noise models - Degradation models-Methods to estimate the degradation-Image de-blurring Restoration in the presence of noise only spatial filtering-Periodic noise reduction by frequency domain filtering-Inverse filtering-Wiener Filtering. Geometrical transformation- Point based methods- Surface based methods-Intensity based methods

Feature Extraction: Region of interest (ROI) selection - Feature extraction: Histogram based features - Intensity features-Color, Shape features-Contour extraction and representation-Homogenous region extraction and representation-Texture descriptors - Feature Selection: Principal Component Analysis (PCA).

Image Segmentation: Discontinuity detection-Edge linking and boundary detection. Thresholding-Region oriented segmentation- Histogram based segmentation. Object recognition based on shape descriptors. Dilation and Erosion-Opening and Closing-Medial axis transforms-Objects skeletons-Thinning boundaries.

Unit-IV

Image Coding & Compression: Lossless compression versus lossy compression-Measures of the compression efficiency- Hufmann coding-Bitplane coding-Shift codes-Block Truncation coding- Arithmetic coding-Predictive coding techniques-Lossy compression algorithm using the 2-D. DCT transform-The JPEG 2000 standard Baseline lossy JPEG, based on DWT.

OpenCV: Installation of OpenCV-Python, OpenCVStandard Images and Data Sets, Python for IPCV, Python for Image Processing, Contrast Stretching, Linear Filtering, Histogram Equalization, Gaussian Convolution, Separable Gaussian Convolution, Gaussian Derivatives, Comparison of theory and practice, Canny Edge Detector, Histogram of Oriented Gradients, Preprocessing, Calculate the Gradient Images, Calculate HOG in 8x8 Cells, Block Normalization, Calculate the HOG feature vector, Visualizing the HOG.

Case Study: Human Iris location, hole detection, Generalized Hough Transform (GHT), spatial matched filtering, GHT for ellipse detection, object location, GHT for feature collation.

Suggested Readings:

1. D. L. Baggio et al.: Mastering OpenCV with Practical Computer Vision Projects‖, Packt Publishing.
2. Jan Erik Solem: Programming Computer Vision with Python: Tools and algorithms for analyzing images, O'Reilly Media.
3. Mark Nixon and Alberto S. Aquado: Feature Extraction & Image Processing for Computer Vision‖, Third Edition, Academic Press.
4. R. Szeliski: Computer Vision: Algorithms and Applications‖, Springer.
5. Simon J. D. Prince: Computer Vision: Models, Learning, and Inference‖, Cambridge University Press.
6. Richard Szeliski, Computer Vision: Algorithms and Applications (Texts in Computer Science).
7. E. R. Davies: Computer and Machine Vision: Theory, Algorithms and Practicalities.
8. ValliappaLakshmanan , Martin Görner: Practical Machine Learning for Computer Vision: End-to- End Machine Learning for Images, O'reilly
9. Forsyth Ponce, Computer Vision: A Modern Approach (2nd Edition), Pearson Publication.
10. Alberto FernándezVillán: Mastering OpenCV 4 with Python, Packt Publication.
11. David MillánEscrivá , Robert Laganiere: OpenCV 4 Computer Vision Application Programming Cookbook.
12. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practical/Lab Assignments

1. Load and display an image using Python.
2. Use Python libraries like OpenCV or PIL to read an image file and display it in Jupyter Notebook.
3. Apply geometric transformations (Rotate, scale, and translate) on an image of your choice. Use OpenCV or skimage or other libraries.
4. Perform global thresholding on an image to segment objects from the background and visualize the results.
5. Perform adaptive thresholding on an image to segment objects from the background and visualize the results.
6. Implement Sobel edge detection algorithm to highlight edges in an image.
7. Implement Canny edge detection algorithm to highlight edges in an image.
8. Implement Prewitt edge detection algorithm to highlight edges in an image.
9. Apply histogram equalization to an image using OpenCV or skimage to improve the contrast of an image of your choice.
10. Compute features extraction using histograms of oriented gradients (HOG) from an image using appropriate Python libraries.

11. Compute features extraction using corner points detection from an image using appropriate Python libraries.
12. Compute features extraction using texture features from an image using appropriate Python libraries.
13. Implement object detection models such as Haar cascades or deep learning-based models (e.g., YOLO) to detect objects in images.
14. Apply image restoration using Wiener filtering to remove blur and noise from degraded images.
15. Apply image restoration using inverse filtering to remove blur and noise from degraded images.
16. Implement image segmentation algorithms using watershed / region growing / graph-based segmentation to partition an image into regions.
17. Implement Huffman coding based compression algorithms to compress image of your choice.
18. Implement JPEG compression based compression algorithms to compress image of your choice.
19. Implement wavelet-based compression algorithms compression to compress image of your choice.
20. Write Python code to detect shapes such as circles or ellipses in an image using the Generalized Hough Transform.

Any other Programs/Lab Assignments given by the teachers.

Name of the Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	-----
Name of the Course	**Cloud Computing and IoT**	**Course Code**	**24CSM202DS03**
Hours/Week	**3+0+2**	**Credits (L:T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25**	**Time of Examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:
The objective of this course is to provide the learners with comprehensive knowledge to grasp cloud computing fundamentals and understand virtualization concepts, service management features along with cloud security challenges, standards and frameworks using services from Google, Amazon and Microsoft to gain practical exposure to its use in development of Internet of Things (IoT) and applications.

Course Outcomes:
By the end of the course the students will be able to:

CO1: Understand various cloud computing service models and use of various cloud services.
CO2: Perform service management in cloud computing and to know various security concepts.
CO3: Understand cloud functionality on the basis of virtualization.
CO4: To understand the applications of cloud computing in OSI Model for the IoT/M2M Systems.
CO5: Understand the architecture and design principles for IoT.

Unit-I

Cloud Computing Fundamentals: Definition & Evolution of Cloud Computing, Cloud types-NIST model, cloud cube model, Deployment models, Service models, Cloud Reference model, Characteristics & Computing Benefits and Limitations of Cloud, Cloud Architecture, Communication protocols; Cloud computing vs. Cluster computing vs. Grid computing; Applications: Technologies and Process required when deploying Web services; Deploying a web service from inside and Outside of a Cloud. **Services and Applications by Types**: IaaS, PaaS, SaaS, IDaaS, and CaaS.

Virtualization: Objectives, Benefits of Virtualization, Emulation, Virtualization for Enterprise, VMware, Server Virtualization, Data Storage Virtualization, Load balancing and Virtualization, Improving Performance through Load Balancing, Hypervisors, Machine Imaging, Porting of applications in the cloud. Concept of Software- Defined Networking (SDN), Network-Function Virtualization (NFV) and Virtual Network Functions (VNF).

Unit-II

Cloudand Service Management: Features of Network management system, Monitoring of an entire cloud computing deployment stack, lifecycle management of cloud services (six stages of lifecycle). Service Oriented Architecture (SOA), Event driven SOA, Enterprise Service bus, Service Catalogues, Service Level Agreements (SLAs); Managing Data - Scalability & Cloud Services, Database & Data Stores in Cloud , Large Scale Data Processing.

Cloud Security Concepts: Cloud security challenges, Cloud security approaches: encryption, tokenization/ obfuscation, cloud security alliance standards, cloud security models and related patterns.**Use of Platforms & Case Study in Cloud Computing:** Concepts of Platform as a Service, Use of PaaS application frameworks; Use of Google, Amazon and Microsoft Web Services. **Cloud vendors and ServiceManagement:** Amazon cloud, AWS Overview, Installation of AWS, Google app engine, azure cloud, salesforce.

Case Study on Open Source & Commercial Clouds: Eucalyptus, Microsoft Azure, Amazon EC2.

Unit-III

IoT Overview: Introduction to Internet of Things (IoT) and IoT Applications, Conceptual Framework & Architectural View of IoT, Technology Behind IoT, Sources of IoT, M2M communication, Modified OSI Model for the IoT/M2M Systems, data enrichment, data consolidation and device management at IoT/M2M

Gateway, Web communication protocols used by connected IoT/M2M devices, Message communication protocols (CoAP-SMS, CoAPMQ, MQTT,XMPP) for IoT/M2M devices.

Architecture and Design Principles for IoT: Internet connectivity, Internet-based communication, IPv4, IPv6, 6LoWPAN protocol, IP Addressing in the IoT, Application layer protocols: HTTP, HTTPS, FTP, TELNET and ports.

Unit-IV

Data Collection, Storage and Computing using a Cloud Platform: Introduction, Cloud computing paradigm for data collection, storage and computing, Cloud service models, IoT Cloud- based data collection, storage and computing services using Nimbits.

Prototyping and Designing Software for IoT Applications: Introduction, Prototyping Embedded device software, Programming Embedded Device Arduino Platform using IDE, Reading data from sensors and devices, Devices, Gateways, Internet and Web/Cloud services software development. Programming MQTT clients and MQTT server.**IoT Security:** Introduction to IoT privacy and security, Vulnerabilities, Security requirements and threat analysis, IoT Security Tomography and layered attacker model.

Suggested Readings:

1. Anthony T. Velte Toby J. Velte, Robert Elsenpeter: Cloud Computing: A Practical Approach, McGraw-Hill.
2. Kris Jamsa: Cloud Computing: SaaS, PaaS, IaaS, Virtualization and more.
3. Tim Mather, Subra Kumaraswamy, Shahed Latif: Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, O'Reilly Media Inc.
4. Ronald L. Krutz, Russell Dean Vines: Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Wiley-India.
5. Raj Kamal: Internet of Things-Architecture and design principles, McGraw Hill Education.
6. Jan Holler, VlasiosTsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle: From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence, Academic Press.
7. Peter Waher, Learning Internet of Things, PACKT publishing, BIRMINGHAM – MUMBAI
8. Bernd Scholz-Reiter, Florian Michahelles: Architecting the Internet of Things, Springer.
9. Daniel Minoli: Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications, Willy Publications
10. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time

List of Practical /Lab Assignments

1. Install IDE of Arduino and write a program using Arduino IDE to blink LED.
2. Interface LED and buzzer with Arduino to buzz for a period of time.
3. Interface RGB LED with Aurdino to obtain different colours and brightness using PWM.
4. a) Control a servo motor using Arduino with an input given through a push button (e.g: When the push button is pressed the servo motor has to rotate by 15 degrees). b) Rotate Stepper motor either clockwise or anti clockwise at 'n' number of steps using Arduino.
5. Write a program to read the data from the RFID tag and display the information on the display board using Arduino and control LED (e.g: if it is a valid card then the LED should be ON otherwise OFF).
6. Control any two actuators connected to the Arduino using Bluetooth/Wifi.
7. Interface analog/digital sensors with Arduino and analyse the corresponding readings. (Sensors like temperature, alcohol, humidity, pressure, gas, sound pollution, level, weight, flow, proximity, LDR, PIR, pulse, vibration, sound etc..)
8. Set up a web service deployment environment using a cloud platform (e.g., AWS, Google Cloud). Deploy a sample web service application from both inside and outside of the cloud environment. Compare the deployment process, resource utilization, and performance metrics.
9. Configure and manage virtual machines (VMs) using a hypervisor software (e.g., VMware, VirtualBox). Create multiple VM instances, allocate resources, and manage network configurations. Experiment with machine imaging and porting applications in virtualized environments.
10. Implement a lifecycle management process for cloud services, covering stages such as planning, deployment, monitoring, optimization, and retirement. Use cloud management tools to automate and

orchestrate the lifecycle tasks.

11. Design and implement a cloud-based data collection and storage system for IoT devices. Use cloud service models (e.g., IaaS, PaaS) to collect sensor data, store it securely, and perform real-time analytics. Ensure scalability and reliability of the system.
12. Design and implement a cloud-based data collection and storage system specifically tailored for IoT applications using a platform like Nimbits. Set up data ingestion pipelines to collect sensor data from IoT devices, store it in the cloud, and ensure scalability and reliability of the storage solution.

Any other Programs/Lab Assignments given by the teachers.

Name of the Program	M.Sc. in Computer Science (Data Science & Machine Learning)	Program Code	-----
Name of the Course	Deep Learning	Course Code	24CSM202DS04
Hours/Week	3+0+2	Credits (L:T:P)	3:0:1
Max. Marks.	Theory: 75 (50+25) Practical: 25	Time of Examination	3 Hours

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:
To equip the learners with a comprehensive understanding of deep learning concepts, techniques, and architectures, enabling them to apply these methods to real-world problems and advance the state-of-the-art in machine learning applications.

Course Outcomes:
By the end of the course the students will be able to:

CO1:Understand the fundamental principles, theory, and approaches of deep neural networks.
CO2: Learn the main variants of deep learning and their specific applications.
CO3:Understand the key concepts, issues, and practices when training and modelling with deep architectures.
CO4:Implement programming assignments related to neural network topics.
CO5:Understand the concept of deep reinforcement learning and applications of large-scale deep learning.

Unit-I

Fundamentals of Deep Learning: Introduction to Deep Learning: Concepts, historical context, and comparison with traditional machine learning. Bayesian Learning and Decision Surfaces. Linear Classifiers and Machines with Hinge Loss.
Deep Neural Networks: Architectural principles, building blocks, activation functions, and their types. Loss Functions and Empirical Risk Minimization. Regularization, Feature Engineering, Overfitting, Underfitting, and Hyperparameter Tuning.

Unit-II

Convolutional Neural Networks (CNN): Basics, building blocks, backpropagation, dropout layers, optimizers (Momentum, RMSProp, Adam), and architectures (LeNet, AlexNet, VGG16, ResNet). Transfer Learning - Techniques and use cases with image data. R-CNN and its types, Skip Connection Networks, Fully Connected CNNs.

Recurrent Neural Networks (RNN): Concepts, Bidirectional RNNs, Encoder-Decoder Models, Backpropagation Through Time (BPTT), and applications. **Advanced RNN Architectures:** LSTMs, GRUs, Seq2Seq Models, and Attention Mechanism.

Unit-III

Generative Models: Concept of Generative model, Generative vs Discriminative models, Restrictive Boltzmann Machines (RBMs), MCMC &Gibbs Sampling, Gradient Computations in RBMs, Deep Boltzmann Machines, Generative Adversarial Networks, Architectural and training details of GANs, Adversarial attacks on Neural Network.

Recent Trends: Variational Autoencoders (undercomplete, regularized, sparse, denoising),Representational power, layer, size and depth of autoencoders, Stochastic encoders and decoders.

Unit-IV

Deep Reinforcement Learning: Basics of Reinforcement Learning and Deep Reinforcement Learning (DRL).DRL Algorithms (Value and Policy Learning), Q-Learning, Deep Q-Networks. Policy-Based Methods, Policy Gradient, and Model-Based R L .**Large-Scale Deep Learning:** Introduction, Characteristics, Challenges and opportunities in handling massive datasets and complex architectures.

Applications Large-Scale Deep Learning: Computer Vision, Speech Recognition, Natural Language

Processing and Healthcare.

Suggested Readings:

1. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA: The MIT Press.
2. R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification. New York, NY: John Wiley & Sons.
3. F. Chollet, Deep Learning with Python. Shelter Island, NY: Manning Publications.
4. S. Raschka and V. Mirjalili, Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2, 3rd ed. Birmingham, U.K.: Packt Publishing.
5. J. Brownlee, Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python. California, USA: Machine Learning Mastery.
6. Goodfellow and Y. Bengio, Learning Deep Architectures for AI. Hanover, MA: Now Publishers Inc.
7. T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. New York, NY: Springer.
8. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd ed. Sebastopol, CA: O'Reilly Media.
9. D. Foster, Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play. Sebastopol, CA: O'Reilly Media.
10. J. Hawkins, S. Blakeslee, A Thousand Brains: A New Theory of Intelligence. New York, NY: Basic Books.
11. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practical/Lab Assignments

1. Implement a simple linear classifier using Python. Train it on a small dataset (e.g. Iris dataset) and evaluate its performance.
2. Visualize the decision surface of a linear classifier on a 2D dataset using matplotlib. Show how the decision surface changes with different parameters.
3. Implement a linear SVM from scratch using hinge loss and gradient descent. Train it on a binary classification dataset and evaluate its performance.
4. Implement gradient descent optimization for a simple regression problem. Plot the cost function over iterations to visualize the convergence.
5. Implement a simple MLP using a deep learning framework (e.g., TensorFlow or PyTorch). Train it on the MNIST dataset and evaluate its performance.
6. Implement backpropagation for the MLP from scratch. Train it on a small dataset and compare the results with the built-in functions from a deep learning framework.
7. Implement an autoencoder for image compression. Train it on the MNIST dataset and visualize the compressed representations and the reconstructed images.
8. Build a simple CNN using a deep learning framework. Train it on the CIFAR-10 dataset and evaluate its performance.
9. Use a pre-trained CNN (e.g., VGG16) for transfer learning. Fine-tune it on a new dataset (e.g., flower classification) and evaluate its performance.
10. Implement and compare different optimizers (SGD, Momentum, RMSProp, Adam) on a simple MLP. Plot the loss curves and analyze their convergence behaviors.
11. Implement early stopping and dropout in training a deep neural network. Train it on the CIFAR-10 dataset and evaluate the impact on performance and overfitting.
12. Implement and compare batch normalization, instance normalization, and group normalization in a CNN. Train it on the CIFAR-10 dataset and evaluate their effects on training speed and performance.
13. Implement a simple residual network (ResNet) and train it on the CIFAR-10 dataset. Evaluate its performance and compare it with a plain CNN of similar depth.
14. Implement a deep learning model for image denoising using autoencoders. Train it on a noisy version of the MNIST dataset and evaluate the denoising quality.
15. Implement a simple object detection model using a deep learning framework (e.g., YOLO or SSD). Train it on a small object detection dataset and evaluate its performance.

Any other Programs/Lab Assignments given by the teachers.

Name of the Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	**-----**
Name of the Course	**Graph Algorithms and Mining**	**Course Code**	**24CSM202DS05**
Hours/Week	**3+0+2**	**Credits (L:T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25**	**Time of Examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:
The objective of this course is to make the learners comprehend graphs and its concepts to explore and apply graph mining concepts in various applications using techniques for mining frequent subgraphs and use it for effective data representation and discover new information from various types of diverse applications.

Course Outcomes:
By the end of the course the students will be able to:

CO1: Understand the graph theory and graph mining foundations.
CO2: Analyze Graph Mining methods.
CO3: Formulate and solve graph-related problems.
CO4: Apply graph mining algorithms to analyze large-scale datasets on various domains.
CO5: Learn Graph mining algorithms with practical exposure in various domains using free data sets.

Unit-I

Introduction to Graphs: Introduction to Graphs and basic terminology, Representations of Graph, Types of Graphs, Basic algorithms for decomposing graphs into parts, Connectivity of graphs, Matching on graphs. Data Mining: Overview of Data Mining, Basic Mining algorithms for Item Sets and Sequences. Evaluation of Mining approaches.

Unit-II

Graph Algorithms: Graph Coloring, Graphs on surface, Directed graphs, Shortest path algorithms, Algorithms to discover minimum spanning tree, Flows in Networks and Flow algorithms, Searching Graphs and Related algorithms.

Unit-III

Graph Mining: Motivation for Graph Mining, Applications of Graph Mining, Mining Frequent Sub graphs – Transactions, BFS/Apriori Approach (FSG and others), DFS Approach (GSPAN and others), Diagonal and Greedy Approaches, Constraint-based mining and New algorithms, Mining Frequent Sub graphs, Graph visualizations.

Unit-IV

Applications of Graph Mining: Web Mining, Centrality analysis, Link analysis algorithms, Graph clustering and Community Detection, Node classification and Link prediction, Influential spreaders, Influence maximization, Geo-social and location based networks.

Suggested Readings:

1. Diestel,R.: Graph Theory, 4th ed. Springer-Verlag, Heidelberg
2. J. Han and M. Kamber: Data Mining–Concepts and Techniques, 2nd Edition, Morgan Kaufman Publishers.
3. Bing Liu, Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, Springer publishing.
4. Jure Leskovec, AnandRajaraman, Jeff Ullman. Mining of Massive Datasets. Cambridge University Press
5. David Easley and Jon Kleinberg. Networks, Crowds, and Markets. Cambridge University Press.
6. DeepayanChakrabarti and Christos Faloutsos: Graph Mining: Laws, Tools, and Case Studies.

7. Synthesis Lectureson Data Mining and Knowledge Discovery, Morgan & Claypool Publishers.
8. Albert-LászlóBarabási. Network Science. Cambridge University Press.
9. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practical/Lab Assignments

1. Represent the given graph using adjacency matrix.
2. Represent graph using adjacency list.
3. Implement Depth-First Search (DFS) on a graph of your choice.
4. Implement Breadth-First Search (BFS) on a graph of your choice.
5. Implement Shortest Path Algorithm using Dijkstra's algorithm.
6. Implement Bellman-Ford algorithm for finding the shortest path in a graph.
7. Implement the Minimum Spanning Tree (MST) algorithm using Kruskals approach.
8. Implement the Minimum Spanning Tree (MST) algorithm using Prim's approach.
9. Identify the Connected Components in an undirected graph.
10. Identify the Connected Components in a directed graph.
11. Implement an algorithm to check if a graph is bipartite or not in a hypothetical real-world problem.
12. Implement the detection of community using the Girvan-Newman algorithm
13. Utilize the community detection in a hypothetical social network graph to detect communities and extract some data analysis over it.
14. Implement the PageRank algorithm on a small web graph for ranking of the nodes.
15. Implement other basic data mining algorithms for discovering frequent item sets and sequences in transactional datasets using Apriori and FP-growth to mine frequent patterns and check the performance of the mining approaches in terms of runtime and memory usage.

Any other Programs/Lab Assignments given by the teachers.

Name of the Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	-----
Name of the Course	**Advanced Machine Learning and Generative AI**	**Course Code**	**24CSM202SE01**
Hours/Week	**2+0+4**	**Credits (L:T:P)**	**2:0:2**
Max. Marks.	**Theory: 50(35+15)** **Practical: 50**	**Time of Examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objective:

The objective of this course is to enable the learners with ample knowledge of deep learning architectures so as to analyze various complex tasks for optimization using concepts of training of deep learning models. It builds enough theoretical foundations of generative AI that is useful in exploring various generative models including its research and ethical considerations.

Course Outcomes:

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

CO1: Master advanced model architectures for diverse machine learning tasks.
CO2: Explore cutting-edge optimization techniques to enhance model performance.
CO3: Delve into advanced topics like Bayesian methods and GANs for robust learning.
CO4: Apply machine learning effectively across real-world domains.
CO5: Develop practical skills through hands-on implementation and experimentation with state-of-the-art algorithms.

Unit-I

Advanced Deep Learning Architectures: Recurrent Neural Networks (RNNs) and their variants (LSTMs, GRUs) for sequential data (text, time series).

Convolutional Neural Networks (CNNs) with advanced layers (e.g., inception modules, residual connections) for improved performance, Understanding the role of hyper parameter tuning and regularization techniques in deep learning models, Exploring optimization algorithms beyond gradient descent (Adam, RMSprop) for efficient training.

Unit-II

Generative Model Fundamentals: Theoretical foundations of generative AI: generative modelling paradigms and applications, Introduction to Generative Adversarial Networks (GANs): architecture, training dynamics, and loss functions.

Variational Autoencoders (VAEs): Understanding the latent space and applications in data generation, Exploring other generative model architectures (e.g., Autoregressive models, Generative Pre-training Transformers).

Unit-III

Applications of Generative AI: Implementing generative models using deep learning frameworks (e.g., Tensor Flow, PyTorch) for practical applications, Generating realistic images (e.g., faces, landscapes) using GANs and exploring creative applications (art generation).

Text generation and manipulation with generative models (e.g., creating realistic dialogue, machine translation),Investigating the potential of generative AI in various domains (e.g., drug discovery, material science)

Unit-IV

Ethical Considerations and Research in Advanced Machine Learning: Evaluating the ethical implications of generative AI: potential for bias, deepfakes, and misuse.

Mitigating bias in generative models and promoting responsible AI development practices, Analyzing research papers in advanced machine learning and generative AI, Presenting your own research findings on a chosen topic in advanced machine learning or generative AI.

Suggested Readings:

1. Pattern Recognition and Machine Learning" by Christopher M. Bishop,
2. Deep Learning" by Ian Goodfellow, YoshuaBengio, and Aaron Courville,
3. Bayesian Reasoning and Machine Learning" by David Barber,
4. Machine Learning: A Probabilistic Perspective" by Kevin P. Murphy.
5. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practical/Lab Assignments

1. Implementing a Simple RNN for Time Series Prediction
2. Implement LSTM for Text Generation
3. Implement the CNN with Inception Module for Image Classification
4. Implement Hyperparameter Tuning with Keras Tuner in python
5. Create a program using Adam Optimizer for Training a Neural Network
6. Use python programming to have a basic GAN implementation
7. Write python code implementing VariationalAutoencoder (VAE) for Data Generation
8. Implementing an Autoregressive Model for Sequence Generation
9. Write python code to implement GAN for Image Generation.
10. Conduct an ethical analysis of a recent research paper in the field of advanced machine learning or generative AI. Identify any potential ethical implications, biases, or risks associated with the proposed methods or applications. Discuss strategies for mitigating these concerns and promoting responsible AI development practices.

Any other Programs/Lab Assignments given by the teachers.

Name of the Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	**-----**
Name of the Course	**Full Stack Development-II**	**Course Code**	**24CSM202MV01**
Hours/Week	**2+0+4**	**Credits (L:T:P)**	**2:0:2**
Max. Marks.	**Theory: 50 (35+15)** **Practical: 50**	**Time of Examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objective:

The objective of this course is to provide the learners with comprehensive knowledge and practical skills in full stack web development, covering both frontend and backend technologies. It will enable them to design, develop, and deploy dynamic web applications using modern web development frameworks and tools.

Course Outcomes:

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

CO1: Understand the principles and technologies involved in full stack web development.
CO2: Develop responsive and interactive user interfaces using frontend frameworks and libraries.
CO3: Implement server-side logic and build RESTful APIs for data interaction.
CO4: Integrate frontend and backend components to create full stack web applications.
CO5: Deploy web applications to production environments and manage development pipelines.

Unit-I

Frontend Development: Introduction to HTML, CSS, and JavaScript, **Responsiveness:** general layouts, building responsive with CSS frameworks, overview of Bootstrap.

JavaScript: overview, implementing dynamic behavior using JavaScript and DOM manipulation, Introduction to frontend frameworks such as React.js or Angular.

Unit-II

Backend Development: Introduction to server-side programming, overview of Node.js, overview of Python, Building server-side applications with frameworks like Express.js (for Node.js), Flask (for Python), connecting to databases and performing CRUD operations, implementing authentication and authorization mechanisms.

Unit-III

Database Management: introduction to relational and non-relational databases, designing and modelling databases using SQL or NoSQL.

Database operations: Performing database operations and queries, introduction to database management systems like MySQL, MongoDB, or PostgreSQL.

Unit-IV

Full Stack Development: Integrating frontend and backend components, building elementary RESTful APIs for data communication, implementing state management and client-server interactions, testing and debugging full stack applications.

Project Deployment: deploying web applications, production environments, configuring and managing web servers, hosting services, developing a full stack web application from scratch.

Suggested Readings:

1. E. Freeman and E. Robson, Head First HTML and CSS: A Learner's Guide to Creating Standards-Based Web Pages, O'Reilly Media.
2. J. Duckett, JavaScript and JQuery: Interactive Front-End Web Development, Wiley.
3. B. Hogan, HTML5 and CSS3: Building Responsive Websites, Wiley.

4. A. Mead, The Complete Node.js Developer Course (3rd Edition), Udemy.
5. M. Wilson, Learning React: A Hands-On Guide to Building Web Applications Using React and Redux, Addison-Wesley Professional.
6. R. Grider, Node with React: Fullstack Web Development, Udemy.
7. Y. Rhee, MongoDB: The Definitive Guide, O'Reilly Media.
8. A. Mead, The Complete React Developer Course (with Hooks and Redux), Udemy.
9. F. Gallego, Learning Flask Framework: Python Web Development with Flask, Packt Publishing.
10. C. Hermann, Full-Stack JavaScript Development: Develop, Test and Deploy with MongoDB, Express, Angular and Node on AWS, Apress.
11. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practical/Lab Assignments

1. Create a simple HTML page with a title, headings, paragraphs, and a list.
2. Apply basic CSS to an HTML page to enhance its appearance.
3. Use Bootstrap to create a responsive layout with a navbar, grid system, and cards.
4. Create a webpage where clicking a button changes the content of a paragraph.
5. Create a simple React component that displays a greeting message.
6. Set up a simple server using Express.js that responds with "Hello World".
7. Set up a simple server using Flask that responds with "Hello World".
8. Implement basic CRUD operations in an Express.js app with MongoDB.
9. Implement user authentication in an Express.js app using Passport.js.
10. Implement basic CRUD operations in a Flask app with SQLite.
11. Create a database, a table, and perform basic CRUD operations using SQL.
12. Perform basic CRUD operations in MongoDB.
13. Build a simple RESTful API for managing items.
14. Create a simple full-stack application with a React frontend and an Express backend.
15. Deploy a full-stack application using a cloud service (e.g., Heroku, AWS).
16. Create a website layout with multiple pages and navigation menus. Utilize Bootstrap's grid system and components to ensure responsiveness. Customize the design using CSS to match a specific theme or brand identity.
17. Develop a web page with dynamic elements such as dropdown menus, interactive forms, and slideshow galleries. Use JavaScript to handle user events, manipulate the DOM, and update content dynamically.
18. Create a RESTful API for managing a collection of data (e.g., users, products) with endpoints for creating, reading, updating, and deleting resources. Use MongoDB as the database to store and retrieve data.
19. Build a web application with user authentication and authorization features using Node.js, Express.js, MongoDB, and a frontend framework (e.g., React.js). Implement user registration, login, and logout functionalities, and restrict access to certain routes based on user roles.
20. Deploy the web application to a cloud-based hosting platform.

Any other Programs/Lab Assignments given by the teachers.

Semester: Third
Option-I (Only Course Work)& Option-II (Course Work and Research)
Session: 2025-26

Name of the Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	**-----------**
Name of the Course	**Predictive Analytics for IoT**	**Course Code**	**25CSM203DS01**
Hours/Week	**4+0+0**	**Credits (L:T:P)**	**4:0:0**
Max. Marks.	**Theory: 100 (70+30)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

This course introduces students to equip students with advanced predictive analytics techniques and tools for processing, analyzing, and deriving insights from IoT data, emphasizing practical applications and problem-solving in real-world IoT systems.

Course Outcomes:

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

CO1: Identify and address IoT data challenges such as noise, missing values, and data integration.
CO2: Apply predictive maintenance and anomaly detection techniques in IoT environments.
CO3: Use deep learning and machine learning frameworks for IoT data forecasting and analysis.
CO4: Implement edge analytics and optimize predictive models for resource-constrained IoT devices.
CO5: Design predictive analytics solutions for diverse IoT applications like smart cities, healthcare, and transportation.

Unit – I

Challenges and opportunities in predictive analytics for IoT data, Characteristics of IoT Data: Realtime streaming, high frequency, and historical data. Advanced Data Preprocessing: handling missing and noisy data (e.g., imputation, filtering). Feature engineering for IoT: Aggregation, time-lag features, Fourier transforms.

Unit – II

Exploring IoT data with Python: Using NumPy, and matplotlib for IoT data exploration. Cleaning and standardizing IoT data: Handling missing values, outlier detection, and normalization. Advanced data exploration techniques: Correlation analysis, data distribution visualization, and anomaly detection**.**

Feature Engineering Overview: Importance and techniques in the context of IoT. Feature extraction from IoT data: Aggregating time-series data, creating derived features, and handling categorical data**.**

Unit – III

Feature selection: Techniques like Recursive Feature Elimination (RFE) and correlation-based feature selection. Training predictive models: Supervised learning for fault prediction and maintenance scheduling.

Analyzing model performance: Metrics such as precision, recall, F1-score, and confusion matrix. Scaling predictive analytics solutions: Deploying models using Azure Machine Learning and monitoring performance.

Unit – IV

IoT Applications: Smart Cities: Traffic flow prediction, energy optimization. Smart Agriculture: Crop yield prediction, irrigation forecasting. Healthcare: Predictive patient monitoring systems.

Security and Privacy in IoT Analytics: Challenges in securing IoT data pipelines. Blockchain for IoT data sharing.

Suggested Readings:

1. C. Aggarwal, Machine Learning for IoT, Springer.

2. A. Gaddam, D. Eswaran, Internet of Things and Predictive Analytics for Smart Cities, Wiley.
3. Russell S. & Norvig P., Artificial Intelligence: A Modern Approach, Prentice-Hall.
4. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O’Reilly Media.
5. Sudipta Mukherjee, Intelligent IoT Systems: Machine Learning for IoT, Springer.
6. Cathy O'Neil & Rachel Schutt, Doing Data Science, O'Reilly Media.
7. Jure Leskovec, Anand Rajaraman & Jeffrey Ullman, Mining of Massive Datasets, Cambridge University Press.
8. Laura Igual & Santi Seguí, Introduction to Data Science: A Python Approach to Concepts, Techniques, and Applications, Springer.T. Erl, AI and Analytics for IoT Systems, Pearson.
9. M. Bishop, Pattern Recognition and Machine Learning, Springer.
10. J. Leskovec, A. Rajaraman, J. Ullman, Mining of Massive Datasets, Cambridge University Press.
11. S. Brown, Practical Python for IoT Systems, Packt Publishing.
12. P. G. Sastry, IoT Analytics: Principles and Applications, Wiley India.
13. A. K. Tiwari, Data Science and Analytics for IoT: Advances and Challenges, Springer.
14. B. Balamurugan and M. Manogaran, Predictive Analytics and Data Mining for IoT Applications, Elsevier.
15. V. Kumar, Edge Computing and Predictive Analytics in IoT: A Practical Guide, McGraw Hill Education.
16. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

Name of the Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	-----------
Name of the Course	**Data Visualization & Interpretation**	**Course Code**	**25CSM203DS02**
Hours/Week	**2+0+4**	**Credits (L:T:P)**	**2:0:2**
Max. Marks.	**Theory: 50 (35+15)** **Practical: 50**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

This course introduces students to equip students with the theoretical and practical knowledge required to design, interpret, and implement effective data visualizations for meaningful insights**.**

Course Outcomes:

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

CO1: Understand fundamental principles of data visualization and identify effective visual encoding techniques.
CO2: Apply key techniques to explore, clean, and structure data for visualization.
CO3: Create impactful visualizations using advanced tools and libraries of Power BI.
CO4: Interpret complex data and use storytelling techniques to communicate insights effectively.
CO5: Develop interactive visualizations using Power BI**.**

Unit – I

Introduction to Data Visualization: Importance and evolution of data visualization in decision-making and analytics**.**

Principles of Visualization: Cognitive principles of visual perception: Pre-attentive processing, Gestalt principles, and effective use of color.

Visual Encodings: Marks and channels, data types (categorical, ordinal, quantitative), choosing visual encodings for specific data types**.**

Unit – II

Design Principles: Edward Tufte's design philosophy, the balance between simplicity and detail, avoiding visual clutter, and data-ink ratio. Types of Charts: Overview and appropriate usage of bar charts, line charts, scatter plots, histograms, pie charts, and heatmaps.

Exploratory Data Analysis (EDA): Descriptive statistics (mean, variance, skewness, kurtosis), detecting outliers.

Unit – III

Advanced Visualization Techniques: Multidimensional data visualizations (pair plots, parallel coordinates, and bubble charts). Hierarchical data visualizations (tree maps, dendrograms). Timeseries visualizations (line graphs, area charts, and interactive timelines).

Data Labeling and Annotations: Best practices for annotations, legends, and titles to make visualizations self-explanatory.

Unit – IV

Overview of Power BI: Introduction to Power BI, its architecture, and its components (Power Query, Power Pivot, and Power View). Interface: understanding the Power BI workspace, menus, and navigation panel, connecting Power BI to various data sources like Excel, CSV, and databases**.**

Power Query Editor: cleaning and transforming data (removing duplicates, handling null values**).**

Types of Visualizations: Bar charts, line charts, pie charts, tables, matrix views, and maps. Customizing

Visuals: formatting visual elements, adding labels, tooltips, and colors**.**

Suggested Readings:

1. A. Kirk, Data Visualization: A Handbook for Data-Driven Design, Sage Publications.
2. N. Yau, Data Points: Visualization That Means Something, Wiley.
3. T. Munzner, Visualization Analysis and Design, CRC Press.
4. C. O'Neil and R. Schutt, Doing Data Science, O'Reilly Media.
5. J. Healy, Data Visualization: A Practical Introduction, Princeton University Press.
6. A. Cairo, The Truthful Art: Data, Charts, and Maps for Communication, Wiley.
7. M. Bostock, Interactive Data Visualization for the Web, O'Reilly Media.
8. R. Chang, Practical Web Based Data Visualization with Python, Apress.
9. A. Radocchia, Microsoft Power BI Quick Start Guide, Packt Publishing.
10. G. Helms, Learn Power BI, Packt Publishing.
11. A. S. Rozario, Beginning Power BI: A Practical Guide to Self-Service Data Analytics with Excel 2016 and Power BI Desktop, Apress.
12. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practical/Lab Assignments

1. Import datasets from Excel, CSV file into Power BI.
2. Create a Bar Chart to Visualize a categorical data.
3. Build a Line Chart that show some trends over a year.
4. Design a Pie Chart to represent market share of a product category of your choice.
5. Create a Column Chart to compare some data values of your choice.
6. Filter Data Using Slicers of some hypothetical sales data by product or region.
7. Build a Map Visualization that displays customer locations or sales across different countries using a map.
8. Build a Scatter Plot to analyze the relationship between advertising spend and revenue data of your choice.
9. Create appropriate chart customized visualization to show the use and effect of labels and legends.
10. Create appropriate chart customized visualization to show the use and effect of tooltips.
11. Create appropriate chart customized visualization to show the use and effect of colors.

Any other Programs/Lab Assignments given by the teachers.

Name of the Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	**-----------**
Name of the Course	**Natural Language Processing**	**Course Code**	**25CSM203DS03**
Hours/Week	**2+0+4**	**Credits (L:T:P)**	**2:0:2**
Max. Marks.	**Theory: 50 (35+15)** **Practical: 50**	**Time of Examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

The aim of the course is to provide a comprehensive view of building real-world natural language processing (NLP) applications**.**

Course Outcomes:

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

CO1: Understand key concepts from NLP that are used to describe and analyze language.
CO2: Realize semantics and pragmatics of the English language for text processing.
CO3: Analyze the syntax, semantics and pragmatics of a statement written in a natural language.
CO4: Demonstrate the state-of-art algorithms and techniques for text- based processing of natural language with respect to morphology.
CO5: To understand computational phonology**.**

Unit-I

Overview of Natural Language Processing: Origins of NLP, Challenges of NLP, Stages of NLP, Applications such as information extraction, question answering, and machine translation. The problem of ambiguity, Why NLP is difficult.

Word Level Analysis: Regular Expressions, Finite-State Automata, Morphology Parsing, Spelling Error Detection and Correction, Words and Word Classes, Part-of-Speech Tagging**.**

Regular Expressions: Regular Expressions, Automata, Similarity Computation: Regular Expressions, patterns, FA, Formal Language, NFSA, Regular Language and FSAs, Raw Text Extraction and Tokenization, Extracting Terms from Tokens, Vector Space Representation and Normalization, Similarity Computation in Text.

Unit-II

Morphology and Finite-State Transducers: Inflection, Derivational Morphology, Finite- State Morphological Parsing, The Lexicon and Morphotactics, Morphological Parsing with Finite State Transducers, Combining FST Lexicon and Rules, Lexicon-free FSTs: The Porter Stemmer, Human Morphological Processing.

Matrix Factorization and Topic Modeling: Introduction, Singular Value Decomposition, Nonnegative Matrix Factorization, Probabilistic Latent Semantic Analysis, Latent Dirichlet Allocation**.**

Unit-III

Computational Phonology and Text-to-Speech: Speech Sounds and Phonetic Transcription, The Phoneme and Phonological Rules, Phonological Rules and Transducers, Advanced Issues in Computational Phonology, Machine Learning of Phonological Rules, Mapping Text to Phones for TTS, Prosody in TTS .

Probabilistic Models of Pronunciation and Spelling: Dealing with Spelling Errors, Spelling Error Patterns, Detecting NonWord Errors, Probabilistic Models, Applying the Bayesian method to spelling, Minimum Edit Distance, English Pronunciation Variation, The Bayesian method, Pronunciation in Humans**.**

Unit-IV

N-gram Language Models: The role of language models. Simple N-gram models. Estimating parameters and smoothing. Evaluating language models. Smoothing, Backoff, Deleted Interpolation, N-grams for Spelling and Pronunciation, Entropy.

Markov Model and POS Tagging: Overview of Hidden Markov Models, Parameter estimation, Information sources in tagging: Markov model taggers, The Viterbi Algorithm Revisited**,**

Word Classes and Part-of-Speech Tagging: Tagsets for English, Part of Speech Tagging, Rule-based Part-of-speech Tagging, Stochastic Part-of-speech Tagging, Transformation- Based Tagging.

Suggested Readings:

1. Christopher D. Manning and Hinrich Schuetze: Foundations of Statistical Natural Language Processing MIT press.
2. Steven Bird, Ewan Klein and Edward Loper: Natural Language Processing with Python, O'Reilly Media.
3. Allen J., Natural Language understanding, Benjamin/Cummings Publishing Company Inc.
4. Jensen K., Heidorn G.E., Richardson S.D., Natural Language Processing: The PLNLP Approach, Springer.
5. Siddiqui and Tiwari U.S., Natural Language Processing and Information Retrieval, Oxford University Press
6. Roach P., Phonetics, Oxford University Press.
7. Nitin Indurkhya, Fred J. Damerau "Handbook of Natural Language Processing", Second Edition, CRC Press.
8. Any other book(s) covering the contents of the paper in more depth**.**

Note: Latest and additional good books may be suggested and added from time to time**.**

List of Practical / Lab Assignments

1. WAP to tokenize a given text into words and split it into sentences using the nltk library.
2. Write a program to implement regular expressions to identify and extract specific patterns such as email addresses from a given text.
3. Write a program to implement a morphological parser that divides words into their prefix, stem and suffix based on predefined affix rules.
4. Wap to perform part of speech.
5. Wap to detect and correct spelling errors in a sentence using the TextBlob library, demonstrating basic text correction techniques.
6. Write a program to generate unigrams, bigrams, and trigrams from a given text.
7. Write a program to extract and tokenize raw text data, demonstrating the initial steps in text pre-processing.
8. Write a program to normalize text by converting it to lowercase, removing punctuation, and filtering out stop-words, preparing the text for further analysis.
9. Write a program to implement a Python program that distinguishes between inflectional and derivational morphemes, analysing word forms for morphological changes.
10. Write a program to use the Porter Stemmer algorithm in Python to perform stemming on text, demonstrating the reduction of words to their root forms.
11. Write a program to implement Nonnegative Matrix Factorization (NMF) for topic modelling in Python, extracting underlying topics from a set of documents.
12. Write a program to use Singular Value Decomposition (SVD) for dimensionality reduction.
13. Write a program to create and visualize topic models from a text corpus using Python, helping to understand the distribution and relationship of topics across documents.
14. Write a program to implement a Python program that converts written text into its phonetic transcription, simulating the process of phonetic analysis in speech.
15. Write a program to build a finite-state transducer (FST) that applies phonological rules to convert input text into its phonologically processed form.

Any other Programs/Lab Assignments given by the teachers.

Name of the Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	**-----------**
Name of the Course	**Reinforcement Learning**	**Course Code**	**25CSM203DS04**
Hours/Week	**4+0+0**	**Credits (L:T:P)**	**4:0:0**
Max. Marks.	**Theory: 100 (70+30)**	**Time of Examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

The aim of the course is to provide a comprehensive knowledge about Reinforcement Learning, understanding RL framework, how to apply basic RL algorithms for simple decision making problems and application areas of RL.

Course Outcomes:

By the end of the course, the students will be able to:
CO1: Understand the basics of Reinforcement Learning.
CO2: Understand RL Framework and Markov Decision Process.
CO3: To understand and apply basic RL algorithms for simple sequential decision-making problems in uncertain conditions
CO4: To understand Hierarchical reinforcement learning.
CO5: To understand applications of reinforcement learning in different fields.

Unit-I

Introduction: Basics of RL, RL task formulation (action space, state space, environment definition), Defining RL framework, The Reinforcement Learning problem: evaluative feedback, nonassociative learning, Rewards and returns, Markov Decision Processes, Value functions, optimality and approximation.

Bandit Problems: Explore-exploit dilemma, Binary Bandits, Learning automata, exploration schemes Dynamic programming: value iteration, policy iteration, asynchronous DP, generalized policy iteration.

Unit-II

Monte-Carlo methods and Temporal Difference Learning: Monte Carlo: Prediction, Estimation of Action values, Control and Control without Exploring Starts, Policy evaluation, Roll outs, On Policy and Off Policy learning, Temporal Difference Prediction: TD (0), Optimality of TD (0), SARSA: On Policy TD control, Q-learning: Off Policy TD control, R-learning, Games and after states.

Eligibility traces: n-step TD prediction, TD (lambda), forward and backward views, Q(lambda), SARSA (lambda), replacing traces and accumulating traces.

Unit-III

Function Approximation: Value prediction, gradient descent methods, linear function approximation, Control algorithms, Fitted Iterative Methods Policy Gradient methods: non-associative learning - REINFORCE algorithm, exact gradient methods, estimating gradients, approximate policy gradient algorithms, actor-critic methods.

Deep Reinforcement Learning: Deep Q-Networks, Double Deep-Q Networks (DQN, DDQN, Dueling DQN, Prioritized Experience Replay)

Unit-IV

Hierarchical RL: MAXQ framework, Options framework, HAM framework, Inverse reinforcement learning, Maximum Entropy Deep Inverse Reinforcement Learning, Generative Adversarial Imitation Learning, Recent Trends in RL Architectures, Applications of Reinforcement learning: NLP, healthcare, finance, education, robotics, games computer vision.

Suggested Readings:

1. Sutton, Richard S., and Andrew G. Barto, Reinforcement learning: An introduction," First Edition, MIT press.
2. Sugiyama, Masashi, "Statistical reinforcement learning: modern machine learning approaches, First Edition, CRC Press.
3. "Bandit algorithms," First Edition, Cambridge University Press.

4. Lattimore, T. and C. Szepesvári, "Reinforcement Learning Algorithms: Analysis and Applications," First Edition, Springer.
5. Alexander Zai and Brandon Brown "Deep Reinforcement Learning in Action," First Edition, Manning Publications.
6. Saba Szepesvari. Algorithms for Reinforcement Learning. Morgan & Claypool Publishers.
7. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

Name of Program	**M.Sc in Computer Science (Data Science & Machine Learning)**	**Program Code**	-----------
Name of the Course	**Information Security and Privacy**	**Course Code**	**25CSM203DS05**
Hours/Week	**2+0+4**	**Credits (L:T:P)**	**2:0:2**
Max. Marks.	**Theory: 50 (35+15)** **Practical: 50**	**Time of Examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objective:

The objective of this course is to provide the learners the ability to understand the concepts related to information security and privacy with an emphasis on designing, analyzing, and managing secure systems. This course emphasised on foundational security models, cryptographic techniques, network security, privacy theories and regulatory frameworks, and emerging challenges in secure system management.

Course Outcomes:

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

CO1: Identify and assess threats, vulnerabilities, and risks in diverse computing environments using standard models and frameworks.
CO2: Implement symmetric and asymmetric cryptographic algorithms and protocols for secure communication and data integrity.
CO3: Critically evaluate privacy-preserving techniques and their implications in compliance with regulations.
CO4: Design and propose secure system architectures incorporating access control, secure network design, and incident response planning.
CO5: Assess emerging technologies (such as cloud security, IoT security, and AI-driven threats) and integrate up-to-date best practices into security management strategies.

Unit-I

Foundations of Information Security and Privacy: Overview of Information Security - History, definitions, and importance in today's digital world.

Security Models & Principles: Confidentiality, integrity, availability (CIA Triad); defence in depth; risk management (NIST framework).

Threat Modelling & Attack Vectors: Social engineering, malware, insider threats, APTs.

Fundamentals of Privacy: Definitions, privacy principles, and the interplay between security and privacy.

Unit-II

Cryptography, Network Security, and Access Control: Cryptographic Primitives and Protocols - Symmetric/asymmetric encryption, hash functions, digital signatures, PKI.

Secure Communication Protocols: TLS/SSL, IPSec, secure email, VPNs.

Network Security Techniques: Firewalls, intrusion detection and prevention, segmentation, zero-trust architectures.

Access Control Models: Discretionary, mandatory, and role-based access control (RBAC).

Unit-III

Privacy, Data Protection, and Legal/Regulatory Frameworks: Privacy Theories and Models - Contextual integrity, privacy by design.

Data Protection Techniques: Data anonymization, pseudonymization, differential privacy. Privacy Laws (e.g. GDPR, HIPAA, CCPA, IT Act 2000, IT Rules 2011, PDP Bill 2019, etc.) and other privacy frameworks.

Ethical Issues in Privacy: Balancing individual privacy rights and organizational interests.

Unit-IV

Advanced Topics and Emerging Trends in Information Security: Security Management & Incident Response - Security policy development, auditing, and compliance frameworks.

Digital Forensics and Cyber Incident Handling: Evidence collection, analysis techniques, legal considerations.

Emerging Technologies and Threats: Cloud security, IoT, AI-based threats, blockchain security.

Future Directions: Research trends, advanced cryptographic protocols (post-quantum cryptography), privacy-enhancing technologies.

Suggested Readings:

1. William Stallings: Network Security Essentials (Applications and Standards), Pearson Education.
2. Behrouz A. Forouzan: Cryptography & Network Security by TMH.
3. Ross Anderson: Security Engineering- A Guide to Building Dependable Distributed Systems, Wiley.
4. Ferguson, Schneier, and Kohno: Cryptography Engineering - Design Principles and Practical Applications, Wiley.
5. Julia Lane: Privacy, Big Data, and the Public Good: Frameworks for Engagement, Cambridge University Press.
6. Matt Bishop: Computer Security- Art and Science, Addison-Wesley.
7. Mark Stamp: Information Security, Wiley – India.
8. Godbole: Information Systems Security, Wiley Student Edition.
9. William Stallings: Cryptography and Network Security, Pearson Education.
10. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practical/Lab Assignments

1. Identify assets, threats, and vulnerabilities and develop risk mitigation strategies using available frameworks (e.g. STRIDE, DREAD or any other).
2. Apply best practices to secure a Linux server (disabling unnecessary services, configuring firewalls, SELinux/AppArmor, and secure SSH).
3. Code and test implementations of symmetric encryption (e.g., AES), asymmetric encryption (e.g., RSA), and hash functions (e.g., SHA-256).
4. Set up a web server with HTTPS using TLS/SSL, generate and deploy certificates, and assess security with tools like OpenSSL.
5. Create a Certificate Authority (CA), issue certificates, and use digital signatures to verify data integrity and authenticity.
6. Install and configure an IDS (e.g., Snort/Suricata), generate simulated attack traffic, and analyze logs for anomalous behavior.
7. Use penetration testing tools (Nmap, Metasploit, Wireshark) to scan a target system, identify vulnerabilities, and produce a remediation report.
8. Configure a sample system (web or file server) to enforce RBAC, design roles, assign permissions, and test access scenarios.
9. Work with a dataset to implement differential privacy techniques (using open-source libraries) and evaluate the impact on data utility.
10. Apply k-anonymity, l-diversity, or t-closeness techniques to a real-world dataset and analyze the trade-off between privacy and data usefulness.
11. Develop a comprehensive security policy and incident response plan, then simulate a breach to test the response procedures.
12. Analyze a provided disk image with forensic tools (such as Autopsy or FTK Imager) to recover evidence, examine logs, and document findings.

13. Deploy a private cloud (using platforms like OpenStack or AWS free-tier), configure identity and access management, secure networking, and implement encryption.
14. Set up a simulated IoT network (via virtual devices or Raspberry Pi), implement security measures (encryption, secure boot, authentication), and perform vulnerability scanning.
15. Investigate and implement (or simulate) a post quantum key exchange algorithm (e.g., lattice based cryptography) and compare its performance and security properties to classical methods.

Note:*These lab assignments can be adapted or expanded based on available resources, emerging threats, and student background. Many of the referenced tools (e.g., OpenSSL, Snort, Wireshark, OpenStack, Python cryptography libraries) are freely available. Instructors should ensure that lab environments are set up in a controlled (virtualized or isolated) lab network.*

Any other Programs/Lab Assignments given by the teachers.

Academic Session (2025-26) Option-(I & II)

Semester-III

SEC3/Internship 3/Project Work 1@ 4 Credits

Semester: Third

Option-III (Only Research)

Session: 2025-26

Name of the Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	-----
Name of the Course	Research Methodology	**Course Code**	25CSM204SE01
Hours/Week	**4+0+0**	**Credits (L:T:P)**	**4:0:0**
Max. Marks.	**100**	**Time of Examination**	---------

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objective:
The objective of this course is to understand how to conduct research, research process and different tools used in research.

Course Outcomes:
By the end of the course the students will be able to:
CO1: Learn the concept of research, research process, types of research and basics formats of report writing.
CO2: Learn the use of statistical analytic techniques for data analysis and testing of hypothesis.
CO3: Identify the differences between measurement and scaling and how sample is selected and determined using various approaches.
CO4: To understand sources of data collection and how data is collected from different sources.
CO5: To understand the concept of interpretation and role of computer in mathematical and statistical analysis with applications of relevant research methodologies used in computer science.

Unit-I

Introduction of Research Methodology: Meaning of research, Objectives of research, Types of research, Research approaches, Significance of research, Research methods versus methodology, Research and scientific methods, Research processes, Criteria for good research, Research problem, Selecting the problem, Necessity of defining the problem, Techniques involved in defining a problem.

Research Modelling: Types of Models, Model building and stages, Data consideration and testing, Heuristic and Simulation modelling. Report Writing: Pre writing considerations, Thesis writing, Formats of report writing, formats of publications in Research journals.

Unit-II

Measurement and Scaling Techniques: Measurement: concept, Levels and components of Measurement, Techniques of Developing Measurement Tools, sources of Error in measurement, Tests of Sound Measurement. Scaling: Meaning of Scaling, Bases of Scales- classification, important scaling techniques-Rating and Ranking. Approaches of the scale construction, different types of scales-Arbitrary Scales, Differential Scales, Summated Scales, Cumulative Scales, factor Scales.

Sampling: Sampling Design, steps in sample design, criteria of selecting a sampling procedure, characteristics of a good sample design, different types of sample design

Unit-III

Data Analysis: Data Analysis: Parametric and Nonparametric data, Descriptive and Inferential Analysis.

Research Design: Meaning and need for research design, features of a good design. Important concepts relating to research design: Dependent and independent variables, Extraneous variables, Control, Research design in case of exploratory research studies, Research design in case of hypothesis- testing research studies, basic principles of experimental designs, Important Experimental Designs.

Sampling Theory, Sandler's A-test, Concept of standard errors, Estimating Population mean (μ), Sample size and its Determination.; Research hypothesis,Experimental and non-experimental hypothesis – Testing research, Experimental and control group.

Unit-IV

Qualitative Research: Themes of qualitative Research, Research Strategies; Data collection Techniques, combining qualitative and quantitative research.

Data Analysis and Interpretation of Data: Data Analysis: Parametric and Nonparametric data, Descriptive and Inferential Analysis. Interpretation of Data: Forms of Interpretation, Prerequisites for Interpretation, Precautions in Interpretation, conclusions and Generalizations, sources of Errors in Interpretations, Mathematical and statistical analysis using software tools like MAT Lab, SPSS or free wares tools. The computer: Its role in research.

Suggested Readings:

1. Montgomery, Douglas C., Design and Analysis of Experiments, (Wiley India)
2. Montgomery, Douglas C. & Runger, George C., Applied Statistics & Probability for Engineers (Wiley India)
3. Kothari C.K., Research Methodology- Methods and Techniques (New Age International, New Delhi)
4. Krishnaswamy, K.N., Sivakumar, Appa Iyer, Management Research Methodology; Integration of Principles, Methods and Techniques (Pearson Education, New Delhi)
5. Panneselvam, R., Research Methodology Prentice Hall of India, New Delhi.
6. Fisher, Hafner, Design of Experiments. Ranjit Kumar., Research Methodology.
7. Day Robert A., How to write and publish a scientific paper.

Note: Latest and additional good books may be suggested and added from time to time.

Academic Session (2025-26) Option-(III)

Semester-III

SEC/Vocational Course/Internship @ 4 Credits

Academic Session (2025-26) Option-(III)

Semester-III

Research Thesis/ Project @ 20 Credits

Semester: Fourth
Option-I (Only Course Work)
Session: 2025-26

Name of the Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	**-----**
Name of the Course	**Blockchain Technology**	**Course Code**	**25CSM204DS01**
Hours/Week	**2+0+4**	**Credits (L:T:P)**	**2:0:2**
Max. Marks.	**Theory: 50 (35+15)** **Practical: 50**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

Introduce foundational concepts of blockchain technology, including decentralization, cryptography, and consensus mechanisms. Develop skills to design and implement smart contracts and decentralized applications (DApps). Explore real-world applications of blockchain in finance, supply chain, healthcare, and governance. Analyze challenges such as scalability, security, and regulatory compliance. Prepare students to critically evaluate blockchain platforms and tools for practical use cases.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Explain blockchain fundamentals, including decentralization, cryptography, and consensus algorithms.
CO2: Design and deploy smart contracts on platforms like Ethereum.
CO3: Analyze the architecture of public, private, and consortium blockchains.
CO4: Develop basic decentralized applications (DApps) using blockchain frameworks.
CO5: Evaluate security risks, scalability solutions, and regulatory implications of blockchain systems.

Unit – I

Introduction to Blockchain and Cryptography: Evolution of Blockchain: History, Issues and needs of Blockchain, features, Benefits and challenges of Blockchain, Blockchain structure, Working of Blockchain, Types of blockchain, Blockchain Network and Nodes.

Cryptography Basics: Hash functions, Properties of Hash functions, Digital signatures, Public-Private key pairs. Symmetric vs. asymmetric encryption.

Use Cases: Cryptocurrencies, decentralized finance (DeFi), NFTs.

Unit – II

Blockchain Mechanics and Consensus Algorithms: Transaction lifecycle: validation, mining, block creation. Wallets, addresses, and UTXO (Unspent Transaction Output) model, Decentralized Autonomous Organization (DAO), GHOST, Sidechain, Namecoin (NMC).**Consensus Mechanisms:** Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS). Byzantine Fault Tolerance (BFT), Practical Byzantine Fault Tolerance (PBFT).

Smart Contracts: Definition, purpose, and examples (e.g., Ethereum's Solidity). Life Cycle of a Smart Contract, Oracle problem and off-chain data.**Blockchain Platforms:** Bitcoin, Ethereum, Hyperledger Fabric.

Unit – III

Blockchain Development and Tools: Development Environment Setup: Tools: Remix IDE, Truffle Suite, Meta
Mask, Ganache. Testnets vs. mainnets. Smart Contract Programming: Solidity syntax (variables, functions, modifiers, events).

Decentralized Applications (DApps): Frontend integration (Web3.js, Ethers.js). Inter Planetary File System (IPFS).**Token Standards:** ERC-20 (fungible tokens), ERC-721 (NFTs).

Unit – IV

Scalability Solutions: Layer 2 protocols (Lightning Network, Plasma, Rollups), Sharding, sidechains.**Blockchain Security:** 51% attacks, double-spending, Sybil attacks, Auditing smart contracts.**Regulatory and Ethical Considerations:** Legal frameworks (GDPR, AML), environmental impact of PoW.

Enterprise Applications of Blockchain: Auto executing contracts, Three signature escrow, Triple entry accounting, Elections and voting, Property records, titles Micropayments, Notary, challenges and research Issue in Blockchain, Supply chain, healthcare, and identity management etc.

Suggested Readings:

1. Paul Laurence, “Blockchain: Step-By-Step Guide to Understand.
2. Frank walrtin, “Blockchain: The comprehensive beginner?
3. Daniel Drescher, “Blockchain basics: A non-technical introduction in 25 steps”, Apress.
4. Mark Gates, “Block Chain: Ultimate guide to understanding block chain, bitcoin, crypto currencies, smart contracts and the future of money”, Wise Fox Publishing and Mark Gates.
5. Andreas Antonopoulos, “Mastering Bitcoin: Unlocking Digital Crypto currencies”, O’Reilly Media Inc.
6. Paul Vigna and Micheal J.Casey. The Age of Cryptography.
7. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practical /Lab Assignments

1. Generate SHA-256 Hash: Write a program in Python that takes an input string and computes its SHA-256 hash.
2. Simulate a Simple Blockchain: Create a basic blockchain structure by defining a Block class, computing block hashes, and linking blocks.
3. Build a Merkle Tree: Demonstrate how transactions are aggregated in a Merkle tree.
4. Compare Centralized vs. Decentralized Data Storage: Illustrate the difference between centralized and decentralized data management.
5. Implement a basic PoW algorithm to find a nonce that yields a hash with a specified condition.
6. Simulate a simple PoS mechanism where block creation is based on participants’ stakes.
7. Create a basic decentralized autonomous organization (DAO) voting simulation.
8. Remix IDE: Simple Storage Contract: Write a contract that allows you to store, update, and retrieve an integer value.
9. Develop a contract that implements functions to increment and decrement a counter variable, demonstrating state changes and basic arithmetic.
10. Create a contract using mappings and structs to associate addresses with user profiles or balances.
11. Demonstrate Solidity inheritance by creating a base contract and one or more derived contracts; also show how to use interfaces for contract interaction.
12. Create a basic ERC20 token contract implementing standard functions such as transfer, approve, and transfer From to understand token mechanics.

Any other Programs/Lab Assignments given by the teachers.

Name of the Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	**-----**
Name of the Course	**AR/VR Systems and Wearable Computing**	**Course Code**	**25CSM204DS02**
Hours/Week	**2+0+4**	**Credits (L:T:P)**	**2:0:2**
Max. Marks.	**Theory: 50 (35+15) Practical: 50**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objective:
Introduce students to the fundamentals and evolution of Augmented Reality (AR), Virtual Reality (VR), Mixed Reality (MR), and wearable computing. Explain the hardware and software architectures that power immersive systems and wearable devices. Develop an understanding of human-computer interaction principles as they apply to AR/VR and wearable technologies. Provide hands-on experience with prototyping, simulation, and development tools for creating AR/VR applications and wearable computing solutions.

Course Outcomes:
By the end of the course the students will be able to:

CO1: Describe the evolution, key concepts, and various applications of AR/VR and wearable computing.
CO2: Analyze the hardware components (sensors, displays, trackers) and software architectures used in immersive systems.
CO3: Demonstrate basic programming and design skills for developing AR/VR applications and wearable device prototypes.
CO4: Design user-centric interfaces for immersive and wearable technologies that address ergonomic and usability challenges.
CO5: Evaluate the security, privacy, and ethical implications of deploying AR/VR and wearable computing solutions in real-world scenarios.

Unit – I

Foundations of AR/VR and Wearable Computing: Definitions and key terms: AR, VR, MR, XR, and Wearable Computing. Historical evolution: From early virtual reality experiments to modern wearable devices. Overview of real-world applications: gaming, healthcare, education, industry, and smart living.

Fundamental Concepts and System Architecture: Basic components of AR/VR systems: sensors, displays, tracking, input devices. Wearable computing hardware: smartwatches, fitness trackers, smart glasses, and other body-worn devices. Comparison of centralized vs. distributed processing in AR/VR

Human factors and ergonomics: Considerations in designing for the human body.

Unit – II

Sensor Technologies and Signal Processing: Overview of sensors used in AR/VR and wearables: IMU (accelerometers, gyroscopes), GPS, proximity, and ambient sensors. Basics of analog vs. digital signals and the role of ADC/DAC in sensor data. Calibration techniques and sensor fusion for robust tracking.

Display and Interaction Devices: Display technologies: Head Mounted Displays (HMDs), projection-based displays, smart glasses, and micro displays. Interaction devices: controllers, gesture recognition systems, haptics, and voice interfaces. Integration of sensors with actuators in wearable devices.

Networking and Connectivity: Communication technologies: Bluetooth, Wi-Fi, NFC, and emerging IoT protocols. Latency and bandwidth considerations for real-time AR/VR applications. Basic principles of embedded systems for wearable (e.g., Arduino, ESP32, and system-on-chip solutions).

Unit – III

Development Tools and Frameworks: Overview of popular AR/VR development platforms: Unity, Unreal Engine, ARKit, ARCore. Introduction to simulation and prototyping tools for wearable computing. Setting up development environments and basic project structure.

3D Graphics, Rendering, and Content Creation: Fundamentals of 3D modelling, animation, and rendering pipelines. Techniques for creating immersive environments and realistic user interfaces. Optimizing graphics for low latency and high frame rates.

Human-Computer Interaction (HCI) and UX Design: Principles of spatial interaction and gesture-based input. Designing intuitive user interfaces for AR/VR and wearable. Evaluating user experience and usability in immersive applications.

Case studies: Analysis of successful AR/VR apps and wearable interfaces.

Unit – IV

Advanced Tracking and Localization: Techniques in SLAM (Simultaneous Localization and Mapping) and sensor fusion. Enhancing accuracy and responsiveness in dynamic environments.

Performance Optimization and Power Management: Strategies for real-time rendering and minimizing latency. Power optimization techniques for prolonged wearable device operation.

Security, Privacy, and Ethical Considerations: Threat models: device tampering, data interception, and privacy breaches in AR/VR. Security measures: encryption, secure authentication, and firmware updates. Ethical and legal considerations, including GDPR, user consent, and health impacts.

Suggested Readings:

1. Dieter Schmalstieg, “Augmented Reality: Principles and Practice”, Pearson Education India
2. Bimber, Oliver, and Ramesh Raskar. “Spatial Augmented Reality: Merging Real and Virtual Worlds”, A K Peters/CRC Press.
3. William R. Sherman, “Understanding Virtual Reality: Interface, Application, and Design (The Morgan Kaufmann Series in Computer Graphics)”, Morgan Kaufmann Publishers
4. Starner, Thad, “Wearable Computing”, MIT Press.
5. Rajiv Chopra, “Virtual and Augmented Reality”, Khanna Book Publishing
6. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practical /Lab Assignments

1. Familiarize with industry-standard tools like Unity, Maya, 3DS MAX, ARtoolkit, Vuforia, and Blender.
2. Learn how to work with primitive objects (cubes, spheres, cylinders) in a 3D scene.
3. Download 3D objects from an asset store and integrate them into a scene.
4. Create your own 3D models using modeling tools like Blender or Maya.
5. Build a three-dimensional realistic scene and develop a simple VR-enabled mobile application.
6. Add dynamic audio and text-based special effects to the VR application.
7. Develop VR applications that leverage motion trackers and sensors for full haptic interactivity.
8. Build AR applications with interactive features for e-learning environments, virtual walkthroughs, or historic site visualizations.
9. Develop a simple MR-enabled gaming application that blends real and virtual elements.

Any other Programs/Lab Assignments given by the teachers.

Name of the Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	-----
Name of the Course	**Social Network Analysis**	**Course Code**	**25CSM204DS03**
Hours/Week	**4+0+0**	**Credits (L:T:P)**	**4:0:0**
Max. Marks.	**Theory: 100 (70+30)**	**Time of Examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

Students learn how to identify key individuals and groups in social systems, to detect and generate fundamental network structures.

Course Outcomes:

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

CO1: To understand the concept of semantic web and related applications.
CO2: To develop semantic web related applications.
CO3: To learn knowledge representation using ontology.
CO4: To understand human behavior in social web and related communities.
CO5: To learn visualization of social networks.

Unit-I

Introduction to Semantic Web: Development of Semantic Web - Emergence of the Social Web - Social Network analysis: Development of Social Network Analysis - Key concepts and measures in network analysis - Electronic sources for network analysis: Electronic discussion networks, Blogs and online communities - Web-based networks - Applications of Social Network Analysis.

Modelling, Aggregating and Knowledge Representation: Ontology and their role in the Semantic Web: Ontology-based knowledge Representation - Ontology languages for the Semantic Web: Resource Description Framework - Web Ontology Language - Modelling and aggregating social network data: State-of-the-art in network data representation - Ontological representation of social individuals - Ontological representation of social relationships - Aggregating and reasoning with social network data - Advanced representations.

Unit-II

Extraction And Mining Communities In Web Social Networks: Extracting evolution of Web Community from a Series of Web Archive - Detecting communities in social networks - Definition of community - Evaluating communities - Methods for community detection and mining - Applications of community mining algorithms - Tools for detecting communities social network infrastructures and communities - Decentralized online social networks - Multi-Relational characterization of dynamic social network communities.

Unit-III

Predicting Human Behaviour and Privacy Issues: Understanding and predicting human behaviour for social communities - User data management - Inference and Distribution - Enabling new human experiences - Reality mining - Context - Awareness - Privacy in online social networks - Trust in online environment - Trust models based on subjective logic - Trust network analysis - Trust transitivity analysis - Combining trust and reputation - Trust derivation based on trust comparisons - Attack spectrum and countermeasures.

Unit-IV

Visualization And Applications of Social Networks: Graph theory - Centrality - Clustering – NodeEdge Diagrams - Matrix representation - Visualizing online social networks, Visualizing social networks with matrix-based representations - Matrix and Node-Link Diagrams - Hybrid representations - Applications - Cover networks - Community welfare - Collaboration networks - Co-Citation networks.

Suggested Readings:

1. Peter Mika,― Social Networks and the Semantic Web‖, First Edition, Springer.
2. Borko Furht, ―Handbook of Social Network Technologies and Applications‖, 1st Edition, Springer.
3. Guandong Xu , Yanchun Zhang and Lin Li, ―Web Mining and Social Networking – Techniques and applications, First Edition, Springer.

4. Dion Goh and Schubert Foo, ―Social information Retrieval Systems: Emerging Technologies and Applications for Searching the Web Effectively‖, IGI Global Snippet.
5. Max Chevalier, Christine Julien and Chantal Soulé-Dupuy, ―Collaborative and Social Information Retrieval and Access: Techniques for Improved user Modelling, IGI Global Snippet.
6. John G. Breslin, Alexander Passant and Stefan Decker, ―The Social Semantic Web, Springer.
7. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

Name of Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	-----
Name of the Course	**Decision Sciences**	**Course Code**	**25CSM204DS04**
Hours Per Week	**3+0+2**	**Credits**	**3:0:1**
Maximum Marks	**Theory: 75 (50+25)** **Practical: 25**	**Time of Examinations**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

To develop data-driven decision-making skills by applying mathematical models, optimization techniques, and predictive analytics. Students will explore decision-making under uncertainty, risk management, and leveraging computational tools to analyze and solve complex business and operational problems, enhancing their strategic thinking and analytical capabilities.

Course Outcomes:

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

CO1: Understand decision-making processes and their mathematical foundations.
CO2: Apply optimization techniques to resource allocation problems.
CO3: Use probabilistic models to handle uncertainty in decision-making.
CO4: Build predictive models for strategic decisions.
CO5: Leverage software tools like R, Python, or MATLAB for decision analysis.

Unit – I

Fundamentals of Decision Sciences: Introduction: Decision Science Framework and Importance, Data and Decision Making: Descriptive, Predictive, and Prescriptive Analytics, Decision Trees: Construction and Interpretation, Probability in Decision Making: Bayes Theorem, Risk Analysis, Case Study: Decision Analysis in Real-life Scenarios.

Unit – II

Optimization Techniques: Linear Programming: Concepts, Graphical and Simplex Methods, Integer and Non-linear Programming, Sensitivity Analysis and Duality, Decision Making under Uncertainty: Minimax, Maximin, and Expected Value Criteria, Case Study: Resource Allocation Problem.

Unit – III

Simulation and Forecasting: Monte Carlo Simulation for Decision Making, Forecasting Models: Moving Averages, Exponential Smoothing, ARIMA, Decision Support Systems (DSS): Design and Implementation, Risk Management and Analysis Tools, Case Study: Demand Forecasting for Supply Chain Optimization.

Unit – IV

Advanced Decision-Making Techniques: Game Theory: Concepts and Applications, Zero Sum game with dominance, Pure and Mixed Strategy , Applications, Multi-Criteria Decision Analysis (MCDA): AHP, TOPSIS, Big Data in Decision Sciences: Role of AI and Machine Learning, Ethical Considerations in Decision Sciences, Case Study: Developing a DSS for Strategic Business Decisions.

Suggested Readings:

1. Winston, W. L. Operations Research: Applications and Algorithms. Cengage Learning.
2. Clemen, R. T., & Reilly, T. Making Hard Decisions with Decision Tools Suite. Cengage Learning.

3. Hillier, F. S., & Lieberman, G. J. Introduction to Operations Research. McGraw Hill.
4. Brian Christian, Algorithms to Live By: The Computer Science of Human Decisions, 2016.
5. John Brockman, Thinking: The New Science of Decision-Making, Problem-Solving, and Prediction, 2013.
6. Suresh Chandra Satapathy, Information and Decision Sciences, Part of the book series: Advances in Intelligent Systems and Computing (AISC, volume 701), 2018.
7. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practical/Lab Assignments

1. Write a program to solve a linear programming problem using the Simplex method to optimize an objective function.
2. Write a program to construct a decision tree using the ID3 algorithm to classify data based on attribute values.
3. Write a program to use random sampling to simulate multiple scenarios and estimate the potential outcomes of a decision.
4. Write a program to Apply Bayes' theorem to make decisions under uncertainty by calculating posterior probabilities.
5. Write a program to create a simple voting system that uses majority voting for decision-making purposes.
6. Write a program to implement the Analytic Hierarchy Process (AHP) to evaluate and prioritize multiple decision criteria.
7. Write a program to calculate the expected value of various decision alternatives based on known probabilities and payoffs.
8. Write a program to implement a program to analyze the sensitivity of a decision model's output with respect to its inputs.
9. Write a program to implement simple linear regression to predict outcomes for decision-making.
10. Write a program to implement multi-objective optimization using genetic algorithms.

Any other Programs/Lab Assignments given by the teachers.

Name of Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	-----
Name of the Course	**High Dimensional Data**	**Course Code**	**25CSM204DS05**
Hours Per Week	**4+0+0**	**Credits**	**4:0:0**
Maximum Marks	**Theory: 100 (70+30)**	**Time of Examinations**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objective:
To explore the challenges of analyzing high-dimensional data, focusing on dimensionality reduction, feature selection, and scalable algorithms. Students will learn to handle the "curse of dimensionality," improve machine learning models, and apply computational techniques for effective data analysis in fields like bioinformatics, finance, and AI.

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

CO1: Recognize issues related to high-dimensional data.
CO2: Apply dimensionality reduction techniques like PCA and LDA.
CO3: Use advanced feature selection methods to improve model accuracy.
CO4: Implement algorithms optimized for high-dimensional datasets.
CO5: Conduct practical analysis using libraries like Scikit-learn, TensorFlow, or PyTorch.

Unit – I

Introduction to High Dimensional Data: Definition and Challenges: Curse of Dimensionality, Sources of High Dimensional Data: Genomics, Image Data, Text Data, Data Preprocessing Techniques: Normalization, Feature Selection, Feature Extraction, Dimensionality Reduction: PCA, t-SNE, LDA, Case Study: Handling High Dimensional Genomics Data.

Unit – II

Statistical Analysis and Modeling: Basics of Statistical Inference for High Dimensional Data, Regression Models: Ridge, Lasso, Elastic Net, Clustering in High Dimensions: DBSCAN, Spectral Clustering, Visualization Techniques for High Dimensional Data, Case Study: High Dimensional Data in Healthcare Analytics.

Unit – III

Machine Learning for High Dimensional Data: Support Vector Machines and Kernel Methods, Deep Learning Architectures for High Dimensional Data, Ensemble Techniques: Random Forests, Gradient Boosting, Feature Importance and Model Interpretation Techniques, Case Study: Analyzing High Dimensional Image Data.

Unit – IV

Advanced Topics and Applications: Big Data Tools: Hadoop, Spark for High Dimensional Data, Applications in Genomics, Finance, and Image Processing, Scalability Issues and Solutions, Ethical Considerations in High Dimensional Data Analysis, Case Study: High Dimensional Data Analytics in E-commerce.

Suggested Readings:

1. Hastie, T., Tibshirani, R., & Friedman, J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.
2. Aggarwal, C. C. Data Classification: Algorithms and Applications. Chapman and Hall/CRC.
3. Murphy, K. P. Machine Learning: A Probabilistic Perspective. MIT Press.
4. Peter Bühlmann , Sara van de Geer, Statistics for High-Dimensional Data Methods, Theory and Applications,

2011.
5. John Wright (Author), Yi Ma (Author), High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications, 2022.
6. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

Academic Session (2025-26) Option-(I)

Semester-IV

SEC4/Internship 4/Project Work 1@ 4 Credits

Semester: Fourth

Option-II (Course Work And Research)

Session: 2025-26

Name of the Program	**M.Sc. in Computer Science (Data Science & Machine Learning)**	**Program Code**	**-----**
Name of the Course	Research Methodology	**Course Code**	25CSM204SE01
Hours/Week	**4+0+0**	**Credits (L:T:P)**	**4:0:0**
Max. Marks.	**Theory: 100**	**Time of Examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:
The objective of this course is to understand how to conduct research, research process and different tools used in research.

Course Outcomes:
By the end of the course the students will be able to:
CO1: Learn the concept of research, research process, types of research and basics formats of report writing.
CO2: Learn the use of statistical analytic techniques for data analysis and testing of hypothesis.
CO3: Identify the differences between measurement and scaling and how sample is selected and determined using various approaches.
CO4: To understand sources of data collection and how data is collected from different sources.
CO5: To understand the concept of interpretation and role of computer in mathematical and statistical analysis with applications of relevant research methodologies used in computer science.

Unit-I

Introduction of Research Methodology: Meaning of research, Objectives of research, Types of research, Research approaches, Significance of research, Research methods versus methodology, Research and scientific methods, Research processes, Criteria for good research, Research problem, Selecting the problem, Necessity of defining the problem, Techniques involved in defining a problem.

Research Modelling: Types of Models, Model building and stages, Data consideration and testing, Heuristic and Simulation modelling. Report Writing: Pre writing considerations, Thesis writing, Formats of report writing, formats of publications in Research journals.

Unit-II

Measurement and Scaling Techniques: Measurement: concept, Levels and components of Measurement, Techniques of Developing Measurement Tools, sources of Error in measurement, Tests of Sound Measurement. Scaling: Meaning of Scaling, Bases of Scales- classification, important scaling techniques-Rating and Ranking. Approaches of the scale construction, different types of scales-Arbitrary Scales, Differential Scales, Summated Scales, Cumulative Scales, factor Scales.

Sampling: Sampling Design, steps in sample design, criteria of selecting a sampling procedure, characteristics of a good sample design, different types of sample design

Unit-III

Data Analysis: Data Analysis: Parametric and Nonparametric data, Descriptive and Inferential Analysis.

Research Design: Meaning and need for research design, features of a good design. Important concepts relating to research design: Dependent and independent variables, Extraneous variables, Control, Research design in case of exploratory research studies, Research design in case of hypothesis- testing research studies, basic principles of experimental designs, Important Experimental Designs.

Sampling Theory, Sandler's A-test, Concept of standard errors, Estimating Population mean (μ), Sample size and its Determination.; Research hypothesis,Experimental and non-experimental hypothesis – Testing research, Experimental and control group.

Unit-IV

Qualitative Research: Themes of qualitative Research, Research Strategies; Data collection Techniques, combining qualitative and quantitative research.

Data Analysis and Interpretation of Data: Data Analysis: Parametric and Nonparametric data, Descriptive and Inferential Analysis. Interpretation of Data: Forms of Interpretation, Prerequisites for Interpretation, Precautions in Interpretation, conclusions and Generalizations, sources of Errors in Interpretations, Mathematical and statistical analysis using software tools like MAT Lab, SPSS or free wares tools. The computer: Its role in research.

Suggested Readings:

1. Montgomery, Douglas C., Design and Analysis of Experiments, (Wiley India)
2. Montgomery, Douglas C. & Runger, George C., Applied Statistics & Probability for Engineers (Wiley India)
3. Kothari C.K., Research Methodology- Methods and Techniques (New Age International, New Delhi)
4. Krishnaswamy, K.N., Sivakumar, Appa Iyer, Management Research Methodology; Integration of Principles, Methods and Techniques (Pearson Education, New Delhi)
5. Panneselvam, R., Research Methodology Prentice Hall of India, New Delhi.
6. Fisher, Hafner, Design of Experiments. Ranjit Kumar., Research Methodology.
7. Day Robert A., How to write and publish a scientific paper.

Note: Latest and additional good books may be suggested and added from time to time.

Academic Session (2025-26) Option-(II)

Semester-IV

SEC4/Internship 4 @ 4 credits

Academic Session (2025-26) Option-(II)

Semester-IV

Research Thesis/ Project @ 20 Credits

Option–III(Only Research)

Semester IV

Academic Session (2025-26) Option-(III)

Semester-IV

SEC/VOC/Internship@ 4 Credits

Academic Session (2025-26) Option-(III)

Semester-IV

Research Thesis/ Project @ 20 Credits