COEP Technological University

Department of Computer Science and Engineering

Curriculum Structure

Post Graduate Diploma in

Data Science & Artificial Intelligence

(Effective from: A.Y. 2026-27)

PGDDSAI Computer Science and Engineering

Trimester - I

Sr. No.	Course Type	Course Code	Course Name	L	T	P	S	Cr	Evaluation Scheme (Weightages in %)				
									Theory			Laboratory	
									MSE	TA	ESE	CIE	ESE
01		TBD	Python Programming	2	0	2	1	3	30	20	50	100	--
02		TBD	Statistics and Probability	3	0	0	1	3	30	20	50	--	--
03		TBD	Database Management Systems and SQL	2	0	2	1	3	30	20	50	100	--
04		TBD	Machine Learning Fundamentals	3	0	2	1	4	30	20	50	100	--
05		TBD	Elective – I 1. Advanced SQL and NoSQL Databases 2. Cloud Computing for AI Applications	3	0		1	3	30	20	50	--	--
			Total	**13**	**0**	**6**	**5**	**16**					

Trimester - II

Sr. No.	Course Type	Course Code	Course Name	L	T	P	S	Cr	Evaluation Scheme (Weightages in %)				
									Theory			Laboratory	
									MSE	TA	ESE	CIE	ESE
01		TBD	Fundamentals to Artificial Intelligence	2	0	2	1	3	30	20	50	100	-
02		TBD	Deep Learning	3	0	2	1	4	30	20	50	100	-
03		TBD	Natural Language Processing	3	0	0	1	3	30	20	50	--	-
04		TBD	Big Data Analytics	3	0	2	1	4	30	20	50	100	--
05		TBD	Elective – II 1. Computer Vision 2. R Programming	3	0	0	1	3	30	20	50	-	-
			Total	**14**	**0**	**6**	**5**	**17**					

Trimester - III

Sr. No.	Course Type	Course Code	Course Name	L	T	P	S	Cr	Evaluation Scheme (Weightages in %)				
									Theory			Laboratory	
									MSE	TA	ESE	CIE	ESE

01		TBD	Generative AI and Large Language Models	2	0	2	1	3	30	20	50	100	--
02		TBD	MLOps and AI Deployment	3	0	0	1	3	30	20	50	--	--
03		TBD	Internship / Project	0	0	0	0	6	--	--	--	50	50
			Total	**5**	**0**	**2**	**2**	**12**					

Python Programming

Teaching Scheme	Evaluation Scheme
Lectures: 2 Hrs/ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: 2 Hrs/ Week	Laboratory: CIE: 100 Marks

Course outcomes

1. Understand the fundamentals of Python programming including syntax, variables, operators, and control structures.
2. Develop Python programs using functions, modules, and built-in data structures.
3. Apply file handling techniques to create simple Python applications.
4. Perform numerical computing operations using NumPy arrays and mathematical functions.
5. Analyse and manipulate datasets using Pandas data structures and operations.
6. Develop simple data analysis applications using Python libraries.

Course content

Unit 1: Introduction to Python **[4 Hrs]**

Overview of Python programming language: history, features, and applications; Installing and setting up Python environment; Basic syntax and data types; Variables and input/output operations; Type conversion and comments in Python; Practical programming exercises.

Unit 2: Control Structures and Operators **[4 Hrs]**

Operators: arithmetic, relational, logical, assignment, bitwise, and special operators; Conditional statements: if, elif, else; Looping constructs: for and while loops; Nested loops; break and continue statements; Practical exercises on implementing control structures.

Unit 3: Data Structures **[4 Hrs]**

Introduction to Python data structures: lists, tuples, dictionaries, and sets; Indexing, slicing, and iteration; Operations and methods on data structures; Searching and sorting operations on data structures; Practical exercises on data manipulation.

Unit 4: Functions, Modules, and File Handling **[4 Hrs]**

Understanding functions in Python; Defining and calling functions; Function arguments and return values; Scope of variables; Types of functions: user-defined, recursive, and lambda functions; Introduction to modules and importing modules; File handling in Python; Reading and writing text and CSV files using read(), write(), and related methods.

Unit 5: NumPy for Numerical Computing **[4 Hrs]**

Introduction to NumPy library; Installing and importing NumPy; Creating NumPy arrays; Array indexing and slicing; Mathematical and statistical operations on arrays; Reshaping and broadcasting; Universal functions; Random number generation using NumPy.

Unit 6: Pandas for Data Analysis **[4 Hrs]**

Introduction to Pandas library; Series and DataFrame objects; Reading and writing CSV files using Pandas; Data selection and filtering; Handling missing values; Sorting and grouping data; Statistical analysis using Pandas; Data cleaning and preprocessing.

Self-study **[8 Hrs]**

Advanced applications of loops and conditional statements; Programs on pattern generation and mathematical computations; Advanced operations on lists, dictionaries, tuples, and sets; Recursive and lambda function applications; Creation and use of Python modules; Processing of text and CSV files; Introduction to real-world applications of NumPy; Array manipulation and matrix operations using NumPy; Real-world dataset analysis using Pandas.

Textbooks

1. Wes McKinney, "Python for Data Analysis", O'Reilly Media, 2022, ISBN-13: 978-1098104030.
2. Mark Lutz, "Learning Python", O'Reilly Media, 2013, ISBN-13: 978-1449355739.

Reference Books

1. John Zelle, "Python Programming: An Introduction to Computer Science", Franklin, Beedle & Associates, 2016, ISBN-13: 978-1590282748.
2. Jake VanderPlas, "Python Data Science Handbook", O'Reilly Media, 2016, ISBN-13: 978-1491912058.

Web references

1. https://docs.python.org/3/
2. https://www.w3schools.com/python/

Laboratory assignments

1. Write Python programs using variables, operators, and input/output statements for simple mathematical applications.
2. Develop programs using conditional statements and loops to generate patterns and perform number-based computations.
3. Create applications using lists, tuples, dictionaries, and sets for storing and manipulating data.
4. Develop Python programs using user-defined functions, recursion, lambda functions, and modules.
5. Build a menu-driven calculator application integrating functions and control structures.
6. Implement file handling programs to read, write, append, and process text and CSV files.
7. Develop applications for searching, sorting, and manipulating data using Python data structures.
8. Perform numerical computations using NumPy arrays, matrix operations, and statistical functions.
9. Analyse and preprocess CSV datasets using Pandas including filtering, grouping, sorting, and handling missing values.
10. Create a simple data analysis application integrating Python programming concepts, NumPy, and Pandas.

Statistics and Probability

Teaching Scheme	Evaluation Scheme
Lectures: 3 Hrs/ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Course outcomes

1. Apply concepts of Linear Algebra including vectors, matrices, and matrix decomposition for data science applications.
2. Use dimensionality reduction techniques like PCA and LDA for feature extraction.
3. Perform descriptive and inferential statistics on real-world datasets.
4. Apply hypothesis testing techniques such as T-Test, Z-Test, and Chi-Square Test.
5. Understand probability theory and apply Bayes' Theorem and Naive Bayes classification.
6. Analyze probability distributions and perform time series forecasting using ARIMA models.

Course content

Unit 1: Linear Algebra for Data Science	**[6 Hrs]**

Vectors, scalars, vector projection, orthogonal vectors, linear combination, linear span, linear independence, cosine similarity; Matrices: basics, scalar and matrix, types of matrix, matrix operations, linear transformation, eigen values and eigen vectors. Hands-on: Creating different vectors and matrices, performing vector and matrix operations using Python.

Unit 2: Applied Linear Algebra	**[6 Hrs]**

Matrix Decomposition, LU Decomposition, Singular Value Decomposition, Dimensionality Reduction, Principal Component Analysis, Linear Discriminant Analysis; Regression: Linear Regression, Ordinary Least Square Method, Multiple Regression, Polynomial Regression, Logistic Regression.

Unit 3: Descriptive Statistics	**[6 Hrs]**

Measures of central tendency: mean and its types, median, mode; Measures of dispersion: range, mean absolute deviation, standard deviation, variance, quartiles, skewness, kurtosis; Measure of correlations: Pearson and Rank Correlation, covariance; Frequency Distribution: Histograms, Relative and cumulative frequency. Hands-on: Performing all descriptive statistics operations using Python.

Unit 4: Inferential Statistics and Scaling	**[6 Hrs]**

Statistical Inference, Sample and Population, Data Sampling and its methods, Resampling Techniques; Hypothesis Testing: Null and Alternate Hypothesis, Student's T-Test, Z-Score and Z-Test, Chi-Square Test; Data Scaling: Normalization and Standardization. Hands-on: Data sampling and resampling, Z-test and Chi-Square Test using Python.

Unit 5: Foundations of Probability	**[6 Hrs]**

Concepts of Probability, types and examples, probability axioms and basic principles, sample space, events, and probability calculations; Independent events, mutually exclusive events, marginal, conditional, and joint probability; Permutations and combinations; Bayes' Theorem and its applications; Naive Bayes Classification method. Hands-on: Calculating probabilities and applying Bayes' Theorem in Python.

Unit 6: Probability Distributions and Time Series Analysis	**[6 Hrs]**

Continuous random variables and PDF; Common continuous distributions: uniform, exponential, Poisson, normal; Standard normal distribution, t-distribution, chi-squared distribution; CDF and conditional PDF; Introduction to time series data; Trends, seasonality, and stationarity; ARIMA modelling for forecasting. Hands-on: Generating random numbers from distributions, fitting ARIMA models and making forecasts in Python.

Self-study	**[12 Hrs]**

Study advanced applications of linear algebra, regression techniques, statistical analysis, hypothesis testing, probability models, probability distributions, and time series forecasting methods using Python tools and real-world datasets for data science and machine learning applications.

Textbooks

1. William Mendenhall, Robert Beaver, Barbara Beaver, "Introduction to Probability and Statistics", Cengage Learning, 2013, ISBN-13: 978-1305269477.
2. David C. Lay, Steven R. Lay, Judi J. McDonald, "Linear Algebra and Its Applications", Pearson, 2019, ISBN-13: 978-0135188927.

Reference Books

1. David Freedman, Robert Pisani, Roger Purves, "Statistics", W. W. Norton & Company, 2007, ISBN-13: 978-0393929720.
2. James Stewart, "Calculus: Early Transcendentals", Cengage Learning, 2015, ISBN-13: 978-1285741550.
3. George E. P. Box et al., "Time Series Analysis: Forecasting and Control", Wiley, 2015, ISBN-13: 978-1118675021.

Web references

1. https://scipy.org/
2. https://statsmodels.sourceforge.net/
3. https://numpy.org/doc/stable/reference/routines.linalg.html

Database Management Systems and SQL

Teaching Scheme	Evaluation Scheme
Lectures: 2 Hrs/ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: 2 Hrs/ Week	Laboratory: CIE: 100 Marks

Course outcomes

1. Understand the principles and fundamentals of relational and NoSQL databases.
2. Learn to design, implement, and administer databases using MySQL and MongoDB.
3. Develop proficiency in querying and manipulating data using SQL and MongoDB query language.
4. Gain practical experience in database administration tasks such as backup, recovery, and performance tuning.
5. Apply database management techniques to solve real-world business problems.

Course content

Unit 1: Introduction to Databases and DBMS	**[4 Hrs]**
Overview of databases: types, models, and architectures; Introduction to relational databases and SQL (Structured Query Language); Introduction to NoSQL databases and MongoDB.	
Unit 2: Database Design and Data Modeling	**[4 Hrs]**
Understanding database design principles; Conceptual Design using entity-relationship (ER) modeling; Logical Database Design using relational model; Hands-on exercises on creating database schemas and data models.	
Unit 3: Querying Relational Databases with SQL	**[4 Hrs]**
Introduction to SQL: basic syntax, data retrieval, filtering, sorting, and aggregating; Writing SQL queries to retrieve and manipulate data in MySQL databases; Hands-on exercises on querying relational databases using SQL.	
Unit 4: Querying NoSQL Databases with MongoDB	**[4 Hrs]**
Introduction to MongoDB query language (MongoDB Query API); Writing MongoDB queries to retrieve and manipulate data; Hands-on exercises on querying NoSQL databases using MongoDB.	
Unit 5: Database Administration and Management	**[4 Hrs]**
Overview of database administration tasks: backup, recovery, security, and performance tuning; Managing MySQL databases: creating users, granting privileges, and configuring settings; Managing MongoDB databases: setting up authentication, configuring replication, and monitoring performance.	
Unit 6: Advanced Topics in Database Management	**[4 Hrs]**
Indexing and optimization techniques for improving query performance; Data consistency and integrity constraints in relational databases; Scalability and sharding in NoSQL databases; Case studies and applications of database management in various industries.	

Self-study	**[8 Hrs]**

Introduction to cloud-based database systems; Comparison between SQL and NoSQL databases; Study of different database models such as hierarchical, network, object-oriented, and graph databases; Advanced SQL concepts including joins, subqueries, views, and stored procedures; Introduction to transactions and ACID properties in databases; Database connectivity using Python or other programming languages; Overview of data warehousing and data mining concepts.

Textbooks

1. William Mendenhall, Robert Beaver, Barbara Beaver, "Introduction to Probability and Statistics", Cengage Learning, 2013, ISBN-13: 978-1305269477.
2. David C. Lay, Steven R. Lay, Judi J. McDonald, "Linear Algebra and Its Applications", Pearson, 2019, ISBN-13: 978-0135188927.

Reference Books

1. Seyed M.M. Tahaghoghi, Hugh E. Williams, "Learning MySQL: Get a Handle on Your Data", O'Reilly Media, 2020, ISBN-13: 978-1492054849.

Web references

1. https://dev.mysql.com/doc/
2. https://www.mongodb.com/docs/

Laboratory assignments

1. Create and manage a relational database using MySQL, including database creation, table creation, data insertion, updating, and deletion operations.
2. Design an Entity-Relationship (ER) diagram and convert it into a relational database schema for a real-world application such as a library or hospital management system.
3. Write SQL queries using SELECT, WHERE, ORDER BY, GROUP BY, HAVING, and aggregate functions to retrieve and analyze data from MySQL databases.
4. Implement SQL JOIN operations, subqueries, views, and constraints to manage relationships and maintain data integrity in relational databases.
5. Develop a mini project using MySQL for applications such as Student Management System, Inventory Management System, or Employee Management System.
6. Install and configure MongoDB and perform CRUD (Create, Read, Update, Delete) operations on NoSQL databases using MongoDB Query API.
7. Write MongoDB queries using filtering, projection, sorting, aggregation pipeline, and indexing techniques to manipulate and analyze NoSQL data.
8. Perform database administration tasks in MySQL and MongoDB including user creation, privilege management, backup, recovery, and authentication setup.
9. Analyze and optimize database performance using indexing, query optimization, and performance monitoring techniques in MySQL and MongoDB.
10. Develop a real-world database application integrating MySQL and MongoDB to demonstrate data modeling, querying, security, scalability, and database management concepts.

Machine Learning Fundamentals

Teaching Scheme	Evaluation Scheme
Lectures: 3 Hrs/ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: 2 Hrs/ Week	Laboratory: CIE: 100 Marks

Course outcomes

1. Understand the fundamental concepts of machine learning, including supervised, unsupervised, and reinforcement learning.
2. Apply various machine learning algorithms such as linear regression, logistic regression, decision trees, and ensemble methods.
3. Preprocess and clean datasets effectively, including handling missing values, encoding categorical variables, and scaling features.
4. Evaluate machine learning models using appropriate metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.
5. Perform model selection and hyperparameter tuning to optimize machine learning model performance.
6. Apply techniques for feature engineering including dimensionality reduction and feature selection.

Course content

Unit 1: Basics of Machine Learning and ML Lifecycle **[6 Hrs]**

Overview of machine learning concepts: supervised, unsupervised, and reinforcement learning; Understanding the ML lifecycle: data collection, preprocessing, model building, evaluation, and deployment; Applications of machine learning in healthcare, finance, marketing, and robotics.

Unit 2: Introduction to NumPy, Pandas, and Data Preprocessing **[6 Hrs]**

Introduction to NumPy: arrays, indexing, operations, and functions; Introduction to pandas: data structures (Series, DataFrame), data manipulation, and data analysis; Data preprocessing techniques: handling missing values, data normalization, encoding categorical variables; Hands-on exercises using NumPy and pandas.

Unit 3: Supervised Learning: Regression and Classification **[6 Hrs]**

Introduction to supervised learning: regression and classification; Linear regression: theory, implementation, and evaluation; Logistic regression, decision trees, and ensemble methods (Random Forest, Gradient Boosting) for classification; Evaluating models: metrics (RMSE, MAE, accuracy, precision, recall, F1-score).

Unit 4: Unsupervised Learning **[6 Hrs]**

Introduction to unsupervised learning: clustering and dimensionality reduction; K-means clustering, hierarchical clustering, and dimensionality reduction techniques (PCA, t-SNE); Evaluating clustering algorithms and applications of unsupervised learning; Hands-on exercises on clustering and dimensionality reduction.

Unit 5: Model Evaluation and Hyperparameter Tuning **[6 Hrs]**

Cross-validation techniques: k-fold cross-validation, stratified cross-validation; Hyperparameter tuning using grid search and randomized search; Model evaluation metrics: precision, recall, F1-score, ROC-AUC; Fine-tuning machine learning models for optimal performance.

Unit 6: Data Visualization with Matplotlib and Seaborn	[6 Hrs]

Introduction to data visualization: importance and principles; Line plot, scatter plot, bar plot, histogram, and other plot types using Matplotlib and Seaborn; Customizing plots: labels, colors, styles, annotations; Hands-on exercises on data visualization with real-world datasets.

Self-Study	[12 Hrs]

Advanced machine learning concepts, Basics of deep learning and neural networks, Reinforcement learning fundamentals, Explainable and ethical AI, Introduction to Generative AI and Large Language Models (LLMs), Basics of MLOps and AI deployment, Edge AI and federated learning concepts, Applications of machine learning in healthcare, finance, cybersecurity, robotics, and smart systems.

Textbooks

1. Andreas C. Müller and Sarah Guido, "Introduction to Machine Learning with Python", O'Reilly Media, 2016, ISBN-13: 978-1449369415.
2. Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", O'Reilly Media, 2019, ISBN-13: 978-1492032649.

Reference Books

1. Christopher M. Bishop, "Pattern Recognition and Machine Learning", Springer, 2006, ISBN-13: 978-0387310732.
2. Sebastian Raschka, Vahid Mirjalili, "Python Machine Learning", Packt Publishing, 2019, ISBN-13: 978-1789955750.

Web references

1. https://scikit-learn.org/stable/documentation.html
2. https://www.kaggle.com/learn

Laboratory assignments

1. Explain the types of machine learning and describe the stages involved in the machine learning lifecycle with suitable examples.
2. Create NumPy arrays and perform indexing, slicing, and mathematical operations on them.
3. Use pandas DataFrame operations to load, filter, sort, and analyze a CSV dataset.
4. Perform data preprocessing techniques such as handling missing values, normalization, and encoding categorical variables on a dataset.
5. Implement Linear Regression using Scikit-learn and evaluate the model using RMSE and MAE metrics.
6. Build classification models using Logistic Regression and Decision Tree algorithms and compare their performance using accuracy and F1-score.
7. Implement Random Forest and Gradient Boosting classifiers and analyze their prediction performance on a dataset.
8. Apply K-Means clustering and PCA on a dataset and visualize the clustered data points.
9. Perform hyperparameter tuning using Grid Search and Randomized Search with cross-validation techniques.
10. Create and customize different data visualizations using Matplotlib and Seaborn for a real-world dataset.

Elective – I: Advanced SQL and NoSQL Databases

Teaching Scheme	Evaluation Scheme
Lectures: 3 Hrs/ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Course outcomes

1. Understand advanced concepts of relational and NoSQL databases for scalable data management.
2. Design and optimize relational databases using normalization, indexing, and query optimization techniques.
3. Apply advanced SQL concepts including joins, subqueries, views, stored procedures, and triggers.
4. Perform database operations and aggregation using MongoDB query language.
5. Implement distributed and scalable NoSQL database architectures for big data applications.
6. Develop real-world database applications integrating SQL and NoSQL technologies.

Course content

Unit	Hours
Unit 1: Advanced Relational Database Concepts	**[6 Hrs]**
Overview of advanced database systems; Relational database architecture; Entity Relationship (ER) modelling; Normalization and denormalization techniques; Functional dependencies; Transactions and ACID properties; Database schema design principles. Hands-on: Designing normalized database schemas using MySQL.	
Unit 2: Advanced SQL Programming	**[6 Hrs]**
Advanced SQL queries; Joins and subqueries; Views and indexes; Stored procedures, functions, and triggers; Cursors and transactions; Query optimization techniques; Database security and user management. Hands-on: Implementing stored procedures, triggers, and advanced SQL queries in MySQL.	
Unit 3: Introduction to NoSQL Databases	**[6 Hrs]**
Overview of NoSQL databases; Types of NoSQL databases: document, key-value, column-family, graph databases; CAP theorem and BASE properties; Introduction to MongoDB architecture; Collections and documents; CRUD operations in MongoDB. Hands-on: Creating and managing collections and documents using MongoDB.	
Unit 4: Advanced MongoDB and Aggregation	**[6 Hrs]**
MongoDB Query API; Aggregation framework; Indexing and performance optimization in MongoDB; Replication and sharding; Data modelling in MongoDB; Transactions in MongoDB; Backup and recovery techniques. Hands-on: Implementing aggregation pipelines and indexing in MongoDB.	
Unit 5: Distributed Databases and Big Data Integration	**[6 Hrs]**
Distributed database systems; Data partitioning and replication; Scalability and fault tolerance; Integration of SQL and NoSQL databases with Big Data ecosystems; Database connectivity with Python applications; Real-time data storage techniques. Hands-on: Connecting MySQL and MongoDB with Python applications.	
Unit 6: Database Analytics and Real-World Applications	**[6 Hrs]**

Data warehousing concepts; ETL processes; Database analytics and reporting; Cloud databases and Database-as-a-Service (DBaaS); Case studies of SQL and NoSQL databases in healthcare, finance, and e-commerce applications. Hands-on: Building a mini project integrating SQL and NoSQL databases for analytics applications.

Self-study **[12 Hrs]**

Advanced database optimization and tuning techniques; NewSQL and distributed database systems; Graph databases and knowledge graphs; Cloud-native databases and serverless databases; Data governance, security, and privacy in modern databases; Integration of databases with AI, Big Data, and machine learning systems; Real-time analytics and stream data processing; Research trends in autonomous and intelligent database systems.

Textbooks

1. Seyed M.M. Tahaghoghi, Hugh E. Williams, "Learning MySQL", O'Reilly Media, 2020, ISBN-13: 978-1492054849.
2. Kristina Chodorow, Michael Dirolf, "MongoDB: The Definitive Guide", O'Reilly Media, 2020, ISBN-13: 978-1491954461.

Reference Books

1. Karthik Appigatla, "MySQL 8 Cookbook", Packt Publishing, 2021, ISBN-13: 978-1803234360.
2. Kyle Banker et al., "MongoDB in Action", Manning Publications, 2016, ISBN-13: 978-1617291609.
3. Carlos Coronel, Steven Morris, Peter Rob, "Database Systems: Design, Implementation, and Management", Cengage Learning, 2019, ISBN-13: 978-1337627900.

Web references

1. MySQL Documentation
2. MongoDB Documentation
3. W3Schools SQL Tutorial
4. MongoDB University

Elective – I: Cloud Computing for AI Applications

Teaching Scheme	Evaluation Scheme
Lectures: 3 Hrs/ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Course outcomes

1. Understand the fundamentals of cloud computing and its role in AI applications.
2. Apply virtualization and cloud service models for scalable AI infrastructure.
3. Deploy and manage AI and machine learning applications on cloud platforms.
4. Utilize cloud storage, databases, and distributed computing frameworks for AI workloads.
5. Implement containerization and orchestration technologies for AI model deployment.
6. Develop secure and scalable cloud-based AI solutions using modern cloud tools and services.

Course content

Unit 1: Introduction to Cloud Computing	[6 Hrs]

Overview of cloud computing; Characteristics and benefits of cloud computing; Cloud deployment models: public, private, hybrid, and community cloud; Cloud service models: IaaS, PaaS, and SaaS; Virtualization concepts and virtual machines; Introduction to leading cloud platforms such as AWS, Microsoft Azure, and Google Cloud Platform (GCP). Hands-on: Creating and configuring virtual machines on cloud platforms.

Unit 2: Cloud Infrastructure and Storage	[6 Hrs]

Cloud infrastructure components; Compute, networking, and storage services; Cloud storage types: object storage, block storage, and file storage; Distributed storage systems; Cloud databases: SQL and NoSQL databases on cloud platforms; Data backup and disaster recovery techniques. Hands-on: Configuring cloud storage and deploying databases on cloud platforms.

Unit 3: AI and Machine Learning on Cloud	[6 Hrs]

Overview of AI services in cloud computing; Cloud-based machine learning platforms; Training and deployment of machine learning models on cloud infrastructure; GPU and TPU computing for AI workloads; AutoML and managed AI services; Real-time AI applications using cloud APIs. Hands-on: Deploying machine learning models using cloud AI services.

Unit 4: Containerization and Orchestration	[6 Hrs]

Introduction to Docker and containerization; Creating and managing Docker containers; Docker images and Docker Hub; Container orchestration using Kubernetes; Scaling and load balancing AI applications; Microservices architecture for AI systems. Hands-on: Deploying AI applications using Docker and Kubernetes.

Unit 5: Big Data and Distributed Computing in Cloud	[6 Hrs]

Introduction to distributed computing; Hadoop and Spark in cloud environments; Data pipelines for AI applications; Real-time data streaming and processing; Cloud-based ETL processes; Serverless computing for AI applications. Hands-on: Processing large datasets using Apache Spark on cloud platforms.

Unit 6: Cloud Security and AI Deployment	**[6 Hrs]**
Cloud security concepts and challenges; Identity and access management (IAM); Data encryption and secure communication; Monitoring and logging cloud applications; CI/CD pipelines for AI deployment; Cost optimization and performance tuning in cloud environments. Hands-on: Deploying secure AI applications with monitoring and logging tools.	
Self-study	**[12 Hrs]**
Advanced cloud-native AI architectures; Edge AI and federated learning; MLOps and AI model lifecycle management; Serverless AI deployment and event-driven computing; Multi-cloud and hybrid cloud strategies; AI workload optimization using GPUs and TPUs; Cloud security, governance, and compliance for AI systems; Applications of cloud AI in healthcare, smart cities, IoT, cybersecurity, and generative AI systems.	

Textbooks

1. Thomas Erl, Ricardo Puttini, Zaigham Mahmood, "Cloud Computing: Concepts, Technology & Architecture", Pearson, 2013, ISBN-13: 978-0133387520.
2. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, "Mastering Cloud Computing", McGraw Hill, 2013, ISBN-13: 978-1259029950.

Reference Books

1. Josh Patterson, Adam Gibson, "Deep Learning: A Practitioner’s Approach", O'Reilly Media, 2017, ISBN-13: 978-1491914250.
2. Brendan Burns, Joe Beda, Kelsey Hightower, "Kubernetes: Up and Running", O'Reilly Media, 2022, ISBN-13: 978-1098110208.
3. Karl Matthias, Sean Kane, "Docker: Up & Running", O'Reilly Media, 2018, ISBN-13: 978-1491917572.

Web references

1. https://docs.aws.amazon.com/
2. https://learn.microsoft.com/azure/
3. https://cloud.google.com/docs
4. https://docs.docker.com/
5. https://kubernetes.io/docs/

Fundamentals to Artificial Intelligence

Teaching Scheme	Evaluation Scheme
Lectures: 2 Hrs/ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: 2 Hrs/ Week	Laboratory: CIE: 100 Marks

Course outcomes

1. Compare AI with human intelligence and traditional information processing and discuss its strength and limitations.
2. Apply the basic principles, models and algorithms of AI to recognize, model and solve problems
3. Demonstrate knowledge of basics of the theory and practice of Artificial Intelligence.
4. Design and carry out an empirical evaluation of different algorithms on problem formalization and state the conclusions that the evaluation supports.
5. Apply AI techniques and problem-solving strategies to common AI applications.

Course content

Unit 1: Introduction	**[4 Hrs]**
What is AI, History, AI problems, Production Systems, Problem characteristics, Intelligent Agents, Agent Architecture, AI Application (E-Commerce, & Medicine), AI Representation, Properties of internal representation, Future scope of AI, Issues in design of search algorithms.	
Unit 2: Heuristic search techniques	**[6 Hrs]**
Heuristic search, Hill Climbing, best first search, mean and end analysis, Constraint Satisfaction, A* and AO* Algorithm, Knowledge Representation: Basic concepts, Knowledge representation Paradigms, Structured representation of knowledge, ISA hierarchy, Frame notation, Resolution	
Unit 3: Learning & Planning	**[6 Hrs]**
What is Learning, Types of Learning (Rote, Direct instruction Analogy, Induction, Deduction), Planning: Block world, strips, Implementation using goal stack, Non linear planning with goal stacks, Hierarchical planning, Least commitment strategy.	
Unit 4: Advanced AI Topics	**[6 Hrs]**
Game playing: Min-max search procedure, Alpha beta cutoffs, waiting for Quiescence, Secondary search, Natural Language Processing (NLP): Introduction, Steps in NLP, Syntactic Processing, Semantic analysis, Discourse & Pragmatic Processing. Perception and Action: Perception, Action, Robot Architecture, Machine Learning: Definition of Learning Systems. Goals and applications of machine learning. Aspects of developing a learning system: training data, concept representation, function approximation.	

Unit 5: Neural Networks and Deep Learning	[6 Hrs]
Neurons and biological motivation. Linear threshold units. Perceptrons: representational limitation and gradient descent training. Multilayer networks and backpropagation, Hidden layers and constructing intermediate, distributed representations, Overfitting, learning network structure, Convolutional Neural Networks (CNNs) for image processing, Recurrent Neural Networks (RNNs) for sequential data	

Unit 6: AI Ethics, Future Trends, Applications and Case Studies	**[6 Hrs]**
Bias and fairness in AI, Ethical considerations in AI deployment, Future trends: Explainable AI, AI in healthcare, autonomous systems, AI in robotics, computer vision, and speech recognition, Case studies: Self-driving cars, recommendation systems, fraud detection.	
Self-study	**[8 Hrs]**
Introduction to advanced AI concepts; Intelligent agents and autonomous systems; Basics of reinforcement learning and deep learning; Explainable AI and ethical AI; Generative AI and Large Language Models (LLMs); Applications of AI in healthcare, robotics, cybersecurity, finance, and smart systems.	

Textbooks

1. Elaine Rich and Kerin Knight, Artificial Intelligence, 3rd Edition, McGraw Hill. ISBN13: 9780070087705
2. Eugene, Charniak, Drew Mcdermott, Introduction to artificial intelligence, AddisonWesley. ISBN 0-07-052263-4.

Reference Books

1. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 3rd Edition. ISBN 0-13-103805-2.
2. Tom Mitchell, Machine Learning, McGraw Hill. ISBN-10: 1259096955
3. Herbert A. Simon, The Sciences of the Artificial, MIT Press, 3rd Edition, 1998. ISBN: 9780262190510.
4. George F Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson Edu., 4th Edition. ISBN-13: 978-0-321-54589-3.
5. Ian Goodfellow and Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, 2016. ISBN: 0262035618.
6. Reinforcement Learning: An Introduction – Richard Sutton, Andrew Barto

Web references

1. https://scikit-learn.org/
2. https://www.tensorflow.org/
3. https://pytorch.org/
4. https://keras.io/
5. https://spark.apache.org/mllib/
6. https://platform.openai.com/docs
7. https://colab.research.google.com/

Laboratory assignments

1. Implement A* algorithm.
2. Implement AO* algorithm.
3. Implementation of other Searching algorithms.
4. Implementation of Min/MAX search procedure for game Playing.
5. Implementation of variants of Min/ Max search procedure.
6. Implementation of mini-Project using the concepts studied in the AI course.

Deep Learning

Teaching Scheme	Evaluation Scheme
Lectures: 3 Hrs/ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: 2 Hrs/ Week	Laboratory: CIE: 100 Marks

Course outcomes

1. Understand Deep Learning Fundamentals: Demonstrate knowledge of deep learning concepts and their biological inspirations.
2. Design Neural Networks: Implement and train basic neural networks using TensorFlow and Keras.
3. Apply Optimization Techniques: Utilize gradient descent variants and regularization methods to improve model performance.
4. Build Advanced Neural Networks: Construct and train CNNs and RNNs for image and sequence tasks.
5. Implement Autoencoders: Use autoencoders and VAEs for data compression and anomaly detection.
6. Deploy Models Effectively: Deploy deep learning models as APIs and ensure scalability in real-world applications.

Course content

Unit 1: Introduction to Deep Learning	[6 Hrs]

What is Deep Learning? Deep Learning vs. Machine Learning; Applications and Use Cases; Biological Inspiration; Neurons in the Brain; Biological vs. Artificial Neurons; Overview of ANNs: Layers (Input, Hidden, Output); Perceptron, Feedforward Neural Networks; Basic Terminology: Weights, Biases, Neurons, Activation Functions; Training a Neural Network: Forward Pass, Loss Functions (MSE, Cross-Entropy); Backpropagation and Gradient Descent. Hands-on: Build a basic neural network from scratch using MNIST.

Unit 2: Deep Learning with TensorFlow and Keras	[6 Hrs]

Introduction to TensorFlow and Keras; TensorFlow vs. Keras; Installing and Setting up the Environment; Basic Neural Networks with Keras: Sequential Model API, Defining Layers, Activation Functions; Model Training and Evaluation; Saving and Loading Models; Plotting and Monitoring Loss Curves, Accuracy Metrics; Using TensorBoard for Visualization. Hands-on: Build and train a fully connected neural network for classification using Keras.

Unit 3: Deep Learning Model Optimization	[6 Hrs]

Gradient Descent Variants: SGD, Mini-batch Gradient Descent, Adam, RMSprop, Adagrad; Learning Rate Schedules, Momentum, Nesterov Accelerated Gradient; Regularization Techniques: L1, L2 Regularization, Dropout, Early Stopping. Hands-on: Apply regularization techniques and use Adam optimizer to improve model performance.

Unit 4: Advanced Neural Networks	[6 Hrs]

Understanding Convolutions: Convolution Operation, Filters/Kernels, Feature Maps, Strides, Padding; Pooling Layers; CNN Architectures: LeNet, AlexNet, VGG, ResNet; Transfer Learning with CNNs; RNN Architecture: Vanishing Gradient Problem, BPTT; LSTM Networks: Cell Structure (Forget, Input, Output Gates); GRUs and Applications. Hands-on: Build a CNN for CIFAR-10 classification; Implement an LSTM for text generation or stock price prediction.

Unit 5: Autoencoders and Variational Autoencoders (VAEs)	[6 Hrs]

Introduction to Autoencoders: Encoder and Decoder Architecture, Bottleneck Layer; Types of Autoencoders: Sparse, Denoising, Contractive; Variational Autoencoders (VAEs): Latent Space Representation, KL Divergence and Reconstruction Loss; Applications: Image Reconstruction, Anomaly Detection, Data Denoising. Hands-on: Implement a basic autoencoder for image reconstruction on MNIST.

Unit 6: Model Deployment and Scalability	**[6 Hrs]**

Deploying Deep Learning Models: Model Serialization (SavedModel, H5 formats); Model Deployment on Cloud (AWS); Building APIs for Deep Learning Models using REST API with Flask or FastAPI; Handling Scalability: Distributed Training, Model Parallelism; Using TensorFlow Serving and Kubernetes for Scalability. Hands-on: Deploy a deep learning model as an API using Flask and test in a real-time environment.

Self-study	**[12 Hrs]**

Advanced deep learning architectures and optimization techniques; Transformer networks and attention mechanisms; Generative AI and Large Language Models (LLMs); Graph Neural Networks (GNNs) and multimodal deep learning; Federated learning and edge AI; Explainable AI (XAI) and trustworthy deep learning systems; AI model compression and efficient inference; Real-world applications of deep learning in healthcare, autonomous systems, cybersecurity, robotics, and generative AI.

Textbooks

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", MIT Press, 2016, ISBN-13: 978-0262035613.

Reference Books

1. Charu C. Aggarwal, "Neural Networks and Deep Learning: A Textbook", Springer, 2nd edition, 2020, ISBN-10: 3030332512.
2. Christopher M. Bishop, "Pattern Recognition and Machine Learning", Springer, 2006, ISBN-10: 0387310738.

Web references

1. https://www.tensorflow.org/
2. https://keras.io/

Laboratory assignments

1. Implement a basic Artificial Neural Network (ANN) from scratch using Python and NumPy to understand neurons, weights, biases, activation functions, forward propagation, and backpropagation.
2. Build and train a feedforward neural network using TensorFlow and Keras for handwritten digit classification on the MNIST dataset.
3. Perform model evaluation using accuracy, loss curves, confusion matrix, and TensorBoard visualization for deep learning models.
4. Apply optimization techniques such as SGD, Adam, RMSprop, momentum, and learning rate scheduling to improve neural network performance.
5. Implement regularization techniques including Dropout, Early Stopping, L1, and L2 regularization to reduce overfitting in deep learning models.
6. Develop a Convolutional Neural Network (CNN) for image classification using the CIFAR-10 dataset and compare different CNN architectures.
7. Implement transfer learning using pretrained CNN models such as VGG16, ResNet, or MobileNet for custom image classification tasks.
8. Build and train a Recurrent Neural Network (RNN) or LSTM model for sequence-based applications such as sentiment analysis, text generation, or stock price prediction.
9. Implement an Autoencoder and Variational Autoencoder (VAE) for image reconstruction, denoising, or anomaly detection using the MNIST dataset.
10. Deploy a trained deep learning model as a REST API using Flask or FastAPI and test the model in a real-time

prediction environment.

Natural Language Processing

Teaching Scheme	Evaluation Scheme
Lectures: 3 Hrs/ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Course outcomes

1. Apply the concepts of morphology, syntax, semantics, discourse and pragmatics of natural language.
2. Demonstrate semantics and pragmatics of English language for text processing.
3. Analyse and apply Language Modelling for Natural Language Processing.
4. Apply semantic techniques for Natural Language Processing.
5. Apply POS Tagging using Neural Models.
6. Apply Information Extraction models to the problems of Natural Language Processing.

Course content

Unit 1: Overview and Language Modeling **[6 Hrs]**

Overview: Origins, what is NLP, Fundamental and Scientific goals, Engineering goals, stages of NLP and challenges; Language and Grammar; Processing Indian Languages; NLP Applications; Language Modeling: Various Grammar-based Language Models, Statistical Language Model; Empirical Laws of language: Zipf's law, Heap's law.

Unit 2: Basic Text Processing **[6 Hrs]**

Tokenization: word token, word type, sentence segmentation, feature extraction; Word segmentation, text segmentation, normalization, case folding; Spelling Correction; Morphology; Lemmatization; N-gram Language Modeling; Context-sensitive spelling correction; Probabilistic language model; Auto completion prediction; Evaluation and perplexity; Smoothing techniques.

Unit 3: Stemming and POS Tagging **[6 Hrs]**

Stemming: Stemming, Porter's Algorithm; POS Tagging: Sequence labeling tasks of NLP, POS tagging, POS tag sets, Hidden Markov Model (HMM), HMM characterization: Likelihood of a sequence, Best state sequence, Re-estimation; Models for Sequential tagging: Maximum Entropy, Conditional Random Field.

Unit 4: Information Extraction **[6 Hrs]**

Web Scraping; Introduction to Named Entity Recognition and Relation Extraction; Natural Language Generation: Architecture; Machine Translation; Sentiment and Emotion Analysis; Text Entailment; Question Answering; Analytics and Social Networks; Multilingual word embedding; Introduction to Transformer learning.

Unit 5: Computational Semantics **[6 Hrs]**

Lexical semantics: WordNet and FrameNet; Word Sense Disambiguation; Distributional Semantics and Word-Space models; Word/Sentence/Text embeddings.

Unit 6: Machine Translation **[6 Hrs]**

Rule-based techniques; Statistical Machine Translation (SMT); Cross-Lingual Translation; Sentiment Analysis; Question Answering; Text Entailment; Discourse Processing; Dialog and Conversational Agents; Natural Language Generation.

Self-study **[12 Hrs]**

Advanced transformer architectures and attention mechanisms; Pre-trained language models such as BERT, GPT, T5, and LLaMA; Retrieval-Augmented Generation (RAG); Prompt engineering and fine-tuning techniques; Multimodal NLP systems integrating text, speech, and images; Conversational AI and AI agents; NLP applications in healthcare, finance, social media, and cybersecurity; Ethical issues, bias, hallucination, and responsible AI in NLP systems.

Textbooks

1. Delip Rao, Brian McMahan, "Natural Language Processing with PyTorch", O'Reilly Media, 1st edition, 2019, ISBN: 9781491978238.

Reference Books

1. Aman Kedia, Mayank Rasu, "Hands-On Python Natural Language Processing", Packt Publishing, 2020, ISBN-10: 1838989595.

Web references

1. https://www.nltk.org/
2. https://spacy.io/

Big Data Technologies

Teaching Scheme	Evaluation Scheme
Lectures: 3 Hrs/ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: 2 Hrs/ Week	Laboratory: CIE: 100 Marks

Course outcomes

1. Understand modern Big Data architectures, distributed systems, and cloud-native analytics platforms.
2. Apply Hadoop, Apache Spark, and modern data engineering tools for large-scale data processing.
3. Perform batch, stream, and real-time analytics using distributed computing frameworks.
4. Utilize cloud-based Big Data ecosystems and data lake architectures for scalable analytics.
5. Implement machine learning and AI workflows on Big Data platforms.
6. Develop industry-oriented Big Data solutions using modern analytics, orchestration, and monitoring tools.

Course content

Unit 1: Foundations of Big Data and Modern Data Architecture	**[6 Hrs]**

Introduction to Big Data; Characteristics of Big Data: Volume, Variety, Velocity, Veracity, and Value; Structured, semi-structured, and unstructured data; Modern data architecture; Data warehouses, data lakes, and lakehouses; Batch processing vs. stream processing; Big Data applications in AI, healthcare, finance, cybersecurity, and IoT; Introduction to cloud-native analytics platforms. Hands-on: Exploring real-world Big Data datasets and cloud analytics platforms.

Unit 2: Hadoop Ecosystem and Distributed Storage	**[6 Hrs]**

Introduction to Hadoop ecosystem; Hadoop Distributed File System (HDFS); Hadoop architecture and cluster components; NameNode, DataNode, ResourceManager, and NodeManager; Distributed storage concepts and fault tolerance; Hadoop cluster setup and configuration; Introduction to distributed file systems and object storage. Hands-on: Installing Hadoop and performing HDFS operations.

Unit 3: Apache Spark and Distributed Data Processing	**[6 Hrs]**

Introduction to Apache Spark; Spark architecture and execution model; RDDs, DataFrames, and Datasets; Spark SQL and distributed querying; Data transformations and actions; Spark optimization techniques; Real-time analytics using Spark Streaming; Introduction to Delta Lake and modern analytics pipelines. Hands-on: Processing large datasets using Spark SQL and DataFrames

Unit 4: Data Engineering and Streaming Analytics	**[6 Hrs]**

Data ingestion concepts; Apache Kafka for real-time data streaming; Apache Sqoop and Apache Flume; ETL and ELT pipelines; Workflow orchestration using Apache Airflow; Real-time analytics and event-driven architectures; Data pipeline monitoring and logging. Hands-on: Building streaming pipelines using Kafka and Spark Streaming.

Unit 5: Big Data Analytics and Machine Learning	**[6 Hrs]**

Machine learning on Big Data platforms; Spark MLlib for scalable machine learning; Distributed model training; Feature engineering and preprocessing for large datasets; Introduction to MLOps and model monitoring; AI-powered analytics systems; Graph analytics and recommendation systems. Hands-on: Implementing machine learning pipelines using Spark MLlib.

Unit 6: Cloud Big Data Platforms and Modern Applications **[6 Hrs]**

Cloud computing for Big Data; Big Data services on AWS, Azure, and Google Cloud Platform; Data security and governance; Containerization using Docker; Orchestration using Kubernetes; Serverless analytics and scalable AI systems; Real-world case studies in smart cities, healthcare, fintech, and generative AI applications. Hands-on: Deploying Big Data analytics workflows on cloud platforms.

Self-study **[12 Hrs]**

Advanced distributed computing frameworks; Data lakehouse architecture and modern analytics systems; Real-time event processing and stream analytics; Big Data integration with Artificial Intelligence and Generative AI; Edge analytics and IoT data processing; Federated data systems and data mesh concepts; Advanced cloud-native Big Data platforms; Research trends in scalable analytics, intelligent data systems, and autonomous data engineering.

Textbooks

1. Bill Chambers, Matei Zaharia, "Spark: The Definitive Guide", O'Reilly Media, 2018, ISBN-13: 978-1491912218.
2. Tom White, "Hadoop: The Definitive Guide", O'Reilly Media, 2015, ISBN-13: 978-1491901632.
3. Tyler Akidau, Slava Chernyak, Reuven Lax, "Streaming Systems", O'Reilly Media, 2018, ISBN-13: 978-1491983874.

Reference Books

1. Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia, "Learning Spark", O'Reilly Media, 2020, ISBN-13: 978-1492050049.
2. Nathan Marz, James Warren, "Big Data: Principles and Best Practices of Scalable Real-Time Data Systems", Manning Publications, 2015, ISBN-13: 978-1617290343.
3. Mohammed Guller, "Big Data Analytics with Spark", Apress, 2015, ISBN-13: 978-1484201022.

Web references

1. https://spark.apache.org/
2. https://hadoop.apache.org/
3. https://kafka.apache.org/
4. https://airflow.apache.org/

Laboratory assignments

1. Explore and analyze real-world Big Data datasets to understand the characteristics of Big Data including Volume, Variety, Velocity, Veracity, and Value.
2. Install and configure Hadoop ecosystem components and perform file storage, retrieval, and management operations using HDFS commands.
3. Implement distributed data processing applications using Apache Spark with RDDs, DataFrames, and Spark SQL for large-scale analytics.
4. Develop batch and real-time analytics pipelines using Spark Streaming for processing streaming data from multiple sources.
5. Build data ingestion pipelines using Apache Kafka, Apache Sqoop, and Apache Flume for structured and unstructured data collection.
6. Create ETL and ELT workflows using Apache Airflow for automating and orchestrating Big Data processing tasks.
7. Implement scalable machine learning models using Spark MLlib for classification, clustering, and recommendation system applications.
8. Perform feature engineering, preprocessing, and distributed model training on large datasets using Big

Data analytics frameworks. 9. Deploy containerized Big Data applications using Docker and Kubernetes for scalable cloud-native analytics solutions. 10. Develop a cloud-based Big Data analytics project using AWS, Azure, or Google Cloud Platform integrating Hadoop, Spark, Kafka, and machine learning workflows for real-world applications.

Elective – II: Computer Vision

Teaching Scheme	Evaluation Scheme
Lectures: 3 Hrs/ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Course outcomes

1. Understand fundamentals of image formation, representation, and camera models.
2. Apply image processing techniques for enhancement, filtering, and edge detection
3. Detect, describe, and match visual features in images.
4. Perform image segmentation and morphological processing.
5. Analyse motion, depth, and 3D structure from images and videos.
6. Design basic computer vision applications using classical and modern techniques

Course content

Unit 1: Introduction to Computer Vision and Image Formation	**[6 Hrs]**
Overview of computer vision and applications; human vision vs computer vision; image acquisition systems; pinhole camera model; geometric camera models; digital image representation; sampling and quantization; color spaces and color models (RGB, HSV, YCbCr); basic image transformations.	
Unit 2: Image Processing and Filtering Techniques	**[6 Hrs]**
Image enhancement techniques; contrast stretching and histogram equalization; spatial domain filtering; smoothing filters; sharpening filters; frequency domain analysis; noise models; noise reduction techniques; edge detection operators (Roberts, Sobel, Prewitt, Canny).	
Unit 3: Feature Detection and Description	**[6 Hrs]**
Interest point detection; corner detection (Harris, Shi-Tomasi); blob detection; feature descriptors (SIFT, SURF, ORB conceptual understanding); feature matching techniques; distance metrics; RANSAC for robust matching; applications in image alignment and stitching.	
Unit 4: Image Segmentation and Morphological Processing	**[6 Hrs]**
Thresholding methods; region-based segmentation; edge-based segmentation; clustering approaches (K-means, Mean Shift); graph-based segmentation; morphological operations (erosion, dilation, opening, closing); watershed algorithm; segmentation evaluation metrics.	
Unit 5: Motion Analysis and Depth Estimation	**[6 Hrs]**
Motion detection techniques; optical flow concepts and methods; background subtraction; object tracking basics; stereo vision; depth estimation; camera calibration; epipolar geometry; structure from motion (introductory concepts).	
Unit 6: Object Detection, Recognition, and Modern Vision Systems	**[6 Hrs]**

Template matching; object recognition pipeline; face detection (Viola–Jones approach); image classification fundamentals; introduction to deep learning for computer vision; convolutional neural networks overview; transfer learning concepts; real-world applications using OpenCV and deep vision models.

Self-study **[12 Hrs]**

Advanced deep learning architectures for computer vision; Object detection models (YOLO, Faster R-CNN, SSD); Semantic and instance segmentation techniques; Vision Transformers (ViT) and attention mechanisms; Video analytics and action recognition; 3D computer vision and point cloud processing; Edge AI and real-time vision systems; Applications of computer vision in healthcare, autonomous vehicles, surveillance, robotics, and augmented reality.

Textbooks

1. Richard Szeliski, "Computer Vision: Algorithms and Applications", Springer, 2022, ISBN-13: 978-3030343729.
2. Adrian Rosebrock, "Deep Learning for Computer Vision with Python", PyImageSearch, 2019, ISBN-13: 978-1734149401.
3. Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", Pearson, 2018, ISBN-13: 978-0133356724.

Reference Books

1. Simon J. D. Prince, "Computer Vision: Models, Learning, and Inference", Cambridge University Press, 2012, ISBN-13: 978-1107011793.
2. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", MIT Press, 2016, ISBN-13: 978-0262035613.
3. Jan Erik Solem, "Programming Computer Vision with Python", O'Reilly Media, 2012, ISBN-13: 978-1449316549.

Web references

1. https://opencv.org/
2. https://docs.opencv.org/
3. https://pytorch.org/vision/stable/index.html
4. https://www.tensorflow.org/tutorials/images

Elective – II: R Programming

Teaching Scheme	Evaluation Scheme
Lectures: 3 Hrs/ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Course outcomes

1. Perform vector and matrix operations using R programming.
2. Apply conditional statements and control structures in R programming.
3. Manipulate and manage data stored in files using R programming.

Course content

Unit 1: Introduction to R Programming	[6 Hrs]
Getting started with RStudio, workspace in R, packages in R, installing packages in R, Built-in Datasets in R, defining variables, constants and strings, operators in R	
Unit 2: Data Structures and Functions	**[10 Hrs]**
Vectors, operations on vectors, generating vectors using seq(), Repeating Vector Constants with rep(), NA and NULL values, arrays and matrices, matrix indexing, matrix operations, List, list functions, data frames, extracting sub-data frames, using the rbind() and cbind() functions and alternatives, factors, common functions used with factors, split() and by()	
Unit 3: R Programming Structures	**[8 Hrs]**
Conditions and loops, control functions, apply family, default values for arguments, lazy evaluation in functions, objects and classes, debugging	
Unit 4: Data Manipulation in R	**[8 Hrs]**
Importing and exporting data in R, Reading text files, writing and saving data objects to file in R, data Manipulation using dplyr package, exploratory data analysis using R, data cleaning, data cleaning, ggplot2 package, statistics with R, machine learning implementation in R	

Self-study	[12 Hrs]
Advanced data visualization techniques using ggplot2 and Shiny; Statistical modeling and predictive analytics in R; Time series analysis and forecasting; Introduction to big data analytics using R; Integration of R with Python and databases; Research-oriented data analysis and report generation using R Markdown; Case studies on machine learning and real-world analytics applications using R.	

Textbooks

1. R Programming for Data Science — Leanpub, 2016, ISBN-13: 978-1365061462.
2. The Art of R Programming — No Starch Press, 2011, ISBN-13: 978-1593273842.

Reference Books

1. Hands-On Programming with R — O'Reilly Media, 2014, ISBN-13: 978-1449359010.
2. R for Data Science — O'Reilly Media, 2017, ISBN-13: 978-1491910399.
3. Practical Data Science with R — Manning Publications, 2014, ISBN-13: 978-1617291562.
4. Machine Learning with R — Packt Publishing, 2023, ISBN-13: 978-1804611302.
5. Advanced R — CRC Press, 2nd Edition, 2019, ISBN-13: 978-0815384571.

Web references

1. https://www.r-project.org/
2. https://posit.co/download/rstudio-desktop/
3. https://cran.r-project.org/

Generative AI and Large Language Models

Teaching Scheme	Evaluation Scheme
Lectures: 2 Hrs/ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: 2 Hrs/ Week	Laboratory: CIE: 100 Marks

Course outcomes

1. Understand the fundamentals of Generative AI and Large Language Models (LLMs).
2. Apply deep learning and transformer architectures for generative AI applications.
3. Utilize pre-trained foundation models for text, image, and multimodal generation tasks.
4. Implement prompt engineering, fine-tuning, and retrieval-augmented generation techniques.
5. Develop AI-powered conversational systems and content generation applications.
6. Analyze ethical, security, and deployment challenges in Generative AI systems.

Course content

Unit	Hours
Unit 1: Introduction to Generative AI	**[4 Hrs]**
Overview of Generative AI; Evolution of AI and foundation models; Discriminative vs. generative models; Applications of Generative AI in text, image, audio, and video generation; Introduction to neural networks and deep learning foundations; Overview of generative AI ecosystems and tools. Hands-on: Exploring Generative AI applications using Python and cloud platforms.	
Unit 2: Deep Learning Foundations for Generative Models	**[6 Hrs]**
Artificial Neural Networks (ANN); Convolutional Neural Networks (CNN); Recurrent Neural Networks (RNN); Sequence models and embeddings; Attention mechanism and self-attention; Introduction to transformer architecture; Training and optimization techniques. Hands-on: Implementing neural network models using TensorFlow/PyTorch.	
Unit 3: Large Language Models and Transformers	**[6 Hrs]**
Transformer architecture; Encoder-decoder models; BERT, GPT, T5, and LLaMA models; Tokenization and embeddings; Pre-training and fine-tuning concepts; Context windows and inference; Open-source and proprietary LLM ecosystems. Hands-on: Using Hugging Face Transformers and pre-trained language models.	
Unit 4: Prompt Engineering and Retrieval-Augmented Generation	**[6 Hrs]**
Introduction to prompt engineering; Zero-shot, one-shot, and few-shot prompting; Chain-of-thought prompting; Prompt optimization techniques; Retrieval-Augmented Generation (RAG); Vector databases and semantic search; Embedding models and document retrieval systems. Hands-on: Building prompt-based and RAG-powered applications.	
Unit 5: Multimodal and Generative AI Applications	**[6 Hrs]**
Text generation and summarization; Conversational AI and chatbots; Image generation models: GANs, Diffusion Models, Stable Diffusion; Audio and video generation; Multimodal AI systems integrating text, images, and speech; AI agents and autonomous workflows. Hands-on: Developing multimodal AI applications using APIs and open-source	

frameworks.

Unit 6: Deployment, Ethics, and Future of Generative AI **[6 Hrs]**

Model deployment using APIs and cloud platforms; Fine-tuning and parameter-efficient tuning (LoRA, PEFT); AI safety, bias, hallucination, and alignment; Responsible AI and ethical considerations; Security and privacy in LLM applications; Future trends in Generative AI and autonomous AI systems. Hands-on: Deploying a Generative AI application using FastAPI or cloud AI services.

Self-study **[8 Hrs]**

Advanced prompt engineering techniques; Study of recent Large Language Models and multimodal foundation models; Research papers on Generative AI architectures; AI agents and autonomous systems; Federated learning and edge AI concepts; Optimization techniques for efficient LLM deployment; Explainable AI (XAI) and trustworthy AI; Industry case studies on Generative AI applications in healthcare, cybersecurity, finance, education, and software engineering.

Textbooks

1. Denis Rothman, "Transformers for Natural Language Processing", Packt Publishing, 2021, ISBN-13: 978-1803247339.
2. Delip Rao, Brian McMahan, "Natural Language Processing with PyTorch", O'Reilly Media, 2019, ISBN-13: 978-1491978238.
3. Lewis Tunstall, Leandro von Werra, Thomas Wolf, "Natural Language Processing with Transformers", O'Reilly Media, 2022, ISBN-13: 978-1098103248.

Reference Books

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", MIT Press, 2016, ISBN-13: 978-0262035613.
2. Francois Chollet, "Deep Learning with Python", Manning Publications, 2021, ISBN-13: 978-1617296864.
3. Sebastian Raschka, "Build a Large Language Model (From Scratch)", Manning Publications, 2024, ISBN-13: 978-1633437166.

Web references

1. https://platform.openai.com/docs
2. https://pytorch.org/
3. https://www.tensorflow.org/
4. https://developers.google.com/machine-learning

Laboratory assignments

1. Explore Generative AI applications and foundation models using Python libraries and cloud-based AI platforms for text, image, and audio generation tasks.
2. Implement basic Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs) using TensorFlow or PyTorch.
3. Build and analyze transformer-based models demonstrating attention mechanisms, self-attention, and sequence modeling concepts.
4. Use Hugging Face Transformers to perform text classification, summarization, translation, and text generation using pre-trained LLMs such as BERT and GPT.
5. Implement tokenization, embeddings, and fine-tuning techniques for Large Language Models using open-source transformer frameworks.
6. Design and evaluate prompt engineering techniques including zero-shot, one-shot, few-shot, and chain-of-thought prompting for AI applications.
7. Develop a Retrieval-Augmented Generation (RAG) system using vector databases, embedding models, and semantic search techniques.

8. Create conversational AI applications and chatbots using LLM APIs, prompt engineering, and memory-based interactions.
9. Develop multimodal Generative AI applications integrating text, image, and audio generation using APIs and open-source models such as Stable Diffusion.
10. Deploy a Generative AI application using FastAPI, Flask, or cloud AI services with considerations for scalability, AI ethics, security, bias, and responsible AI practices.

MLOps and AI Deployment

Teaching Scheme	Evaluation Scheme
Lectures: 3 Hrs/ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Course outcomes

1. Understand the fundamentals of MLOps and AI deployment pipelines.
2. Apply version control, containerization, and orchestration techniques for machine learning systems.
3. Implement CI/CD pipelines for automated model training, testing, and deployment.
4. Utilize cloud platforms and monitoring tools for scalable AI applications.
5. Deploy machine learning and deep learning models using APIs and microservices.
6. Develop secure, scalable, and production-ready AI systems following MLOps best practices.

Course content

Unit 1: Introduction to MLOps and AI Lifecycle	**[6 Hrs]**

Overview of MLOps; AI and machine learning lifecycle; Challenges in deploying machine learning models; Data engineering and model management; DevOps vs. MLOps; Continuous Integration, Continuous Delivery, and Continuous Deployment (CI/CD); Introduction to production-ready AI systems. Hands-on: Setting up an MLOps development environment.

Unit 2: Version Control and Experiment Tracking	**[6 Hrs]**

Version control using Git and GitHub; Data and model versioning; Experiment tracking tools; MLflow and Weights & Biases (W&B); Reproducibility in machine learning workflows; Pipeline automation basics. Hands-on: Tracking machine learning experiments and managing versions using MLflow.

Unit 3: Containerization and Orchestration	**[6 Hrs]**

Introduction to Docker and containerization; Creating Docker images and containers; Managing dependencies in AI applications; Introduction to Kubernetes; Container orchestration and scaling; Microservices architecture for AI systems. Hands-on: Containerizing machine learning models using Docker.

Unit 4: CI/CD Pipelines for Machine Learning	**[6 Hrs]**

CI/CD concepts for AI systems; Automated model training and testing; Jenkins, GitHub Actions, and GitLab CI/CD; Automated deployment pipelines; Infrastructure as Code (IaC); Workflow automation and scheduling. Hands-on: Building CI/CD pipelines for machine learning applications.

Unit 5: Cloud Deployment and Monitoring	**[6 Hrs]**

Cloud platforms for AI deployment: AWS, Azure, and Google Cloud; Model deployment using REST APIs and FastAPI/Flask; Serverless AI deployment; Monitoring machine learning models; Logging, drift detection, and performance monitoring; Scalable deployment architectures. Hands-on: Deploying machine learning models on cloud platforms.

Unit 6: Advanced MLOps and Production AI Systems	[6 Hrs]
Model serving frameworks; TensorFlow Serving and TorchServe; Feature stores and data pipelines; Security and governance in AI systems; Responsible AI and ethical deployment; Edge AI and real-time inference systems; Future trends in MLOps and AI infrastructure. Hands-on: Developing and deploying a complete end-to-end MLOps pipeline.	

Self-study	[12 Hrs]
Scalable AI systems; Kubernetes basics for AI deployment; Serverless AI deployment; Real-time and batch inference; AI applications in healthcare, finance, cybersecurity, education and recommendation systems; Future trends in MLOps, Generative AIOps and AutoML. Hands-on: Mini project on end-to-end AI deployment pipeline.	

Textbooks

1. Noah Gift, Alfredo Deza, "Practical MLOps", O'Reilly Media, 2021, ISBN-13: 978-1098103019.
2. Mark Treveil, Nicolas Omont, Clement Stenac, Kenji Lefevre, "Introducing MLOps", O'Reilly Media, 2020, ISBN-13: 978-1492083290.
3. Emmanuel Ameisen, "Building Machine Learning Powered Applications", O'Reilly Media, 2020, ISBN-13: 978-1492045113.

Reference Books

1. Chip Huyen, "Designing Machine Learning Systems", O'Reilly Media, 2022, ISBN-13: 978-1098107963.
2. Ben Wilson, "Machine Learning Engineering in Action", Manning Publications, 2022, ISBN-13: 978-1617298714.
3. Josh Patterson, Adam Gibson, "Deep Learning: A Practitioner's Approach", O'Reilly Media, 2017, ISBN-13: 978-1491914250.

Web references

1. https://mlflow.org/
2. https://github.com/features/actions
3. https://fastapi.tiangolo.com/
4. https://cloud.google.com/ai-platform