NATIONAL INSTITUTE OF ELECTRONICS AND INFORMATION TECHNOLOGY
DEEMED TO BE UNIVERSITY UNDER DISTINCT CATEGORY
(AN AUTONOMOUS INSTITUTION UNDER MINISTRY OF ELECTRONICS & IT, GOVT. OF INDIA)

Curriculum and Syllabi for

M.Tech in Computer Science and Engineering (Artificial Intelligence and Machine Learning)

(2025-26 Scheme)

रा.इ.सू.प्रौ.सं
NIELIT
Deemed to be University

Main Campus at Ropar and Constituent Units at Aizawl, Agartala, Aurangabad, Calicut, Gorakhpur, Imphal, Itanagar, Kekri, Kohima, Patna and Srinagar

Vision

To be a center of excellence in Artificial Intelligence and Machine Learning education and research, empowering graduates to innovate, solve complex problems, and contribute responsibly to the advancement of technology and society.

Mission

1. To impart in-depth knowledge and practical skills in Artificial Intelligence, Machine Learning, and related technologies.
2. To foster innovation, research, and entrepreneurship in solving real-world challenges using intelligent systems.
3. To promote ethical values, lifelong learning, and interdisciplinary collaboration among students and professionals.
4. To Build industry-ready professionals equipped with critical thinking, leadership, and communication skills.
5. To contribute to the socio-economic development through sustainable and inclusive AI applications.

Program Education Objectives

Graduates of the M.Tech (AI & ML) program will:

1. Apply advanced AI and ML knowledge to develop innovative, scalable, and intelligent solutions for industry and society.
2. Pursue successful careers or research in academia, R&D, or entrepreneurial ventures in the AI domain.
3. Demonstrate ethical behavior, leadership, teamwork, and commitment to sustainable development.
4. Engage in continuous learning and adapt to evolving technologies in AI and related fields.
5. Contribute meaningfully to interdisciplinary and cross-cultural problem-solving environments.

Program Outcomes

Postgraduates of the program will be able to:

1. Apply Knowledge: Apply advanced mathematical, statistical, and algorithmic knowledge to model, analyze, and solve complex AI and ML problems.
2. Modern Tool Usage: Use state-of-the-art tools and platforms for data analysis, machine learning, deep learning, and deployment of AI solutions.
3. Research and Innovation: Conduct independent research, analyze scientific literature, and develop new AI techniques with a focus on real-world impact.
4. Ethics and Communication: Demonstrate ethical practices, effective communication, and leadership in multidisciplinary and multicultural environments.
5. Lifelong Learning: Engage in continuous learning to remain updated with emerging AI trends, tools, and practices.

Program Specific Outcomes

After completing the M.Tech in AI & ML, graduates will be able to:

1. Design and implement intelligent systems using machine learning, deep learning, and reinforcement learning techniques for domains such as vision, NLP, and robotics.
2. Apply AI techniques to optimize solutions for business, healthcare, agriculture, and social challenges through research or industry projects.
3. Deploy and scale AI systems using cloud platforms, APIs, and MLOps practices with considerations of performance, ethics, and sustainability.

CATEGORY WISE CREDIT DISTRIBUTION

Category	Category Code	Credits
Skill / Employment Enhancement Courses (Project/Internship/Seminar/Dissertation/Research)	EE	36
Open Elective Courses	OE	8
Audit Courses (without grade or credit)	AU	0
Professional Elective Courses	PE	16
Program Core Courses	PC	20
Total Credits (Common track)		**80**

EVALUATION AND ASSESSMENT

1. Performance of a student in a semester shall be evaluated through continuous Class assessment, Tutorial/Lab assessment, Mid-Semester Examination (MSE) and End-Semester Examination (ESE). Both the MSE and ESE shall be the University examination and will be conducted as notified by the Controller of Examinations (CoE) of the University.

2. The continuous assessment shall be based on assignments, tutorials, paper presentation / quizzes/ viva-voce / flipped classes, lab work / projects / fieldwork and attendance, etc.

3. The MSE/ESE shall be comprising of written papers.

4. The overall assessment of the students will be done as per the following scheme:

S. No.	Assessment Type	Marks Weightage
1.	Mid Semester Examination	20
2.	End Semester Examination	40
3.	Continuous Assessment - Practical /Lab/ Tutorial	25
4.	Continuous Assessment - Theory	15
Total		**100**

For more details, please refer the applicable ordinance for this programme and Assessment SOPs.

CURRICULUM

Semester 1						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-MTCAIM-101	DOAI250040	Mathematics for Machine Learning	3	1	0	4
PC-MTCAIM-102	DOAI250032	Machine Learning	3	0	2	4
PC-MTCAIM-103	DOAI250004	Artificial Intelligence	3	1	0	4
PE-MTCAIM-104	Elective	Program Elective				4
PE-MTCAIM-105	Elective	Program Elective				4
AU-MTCAIM-106	Elective	Audit Elective				0

Semester 2						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-MTCAIM-201	DOAI250024	Deep Learning	2	0	4	4
PC-MTCAIM-202	DCSE250001	Advanced Algorithms and Analysis	3	1	0	4
PE-MTCAIM-203	Elective	Program Elective				4
OE-MTCAIM-204	Elective	Open Elective				4
EE-MTCAIM-205	DASH250095	Research Methodology and IPR	3	1	0	4
AU-MTCAIM-206	Elective	Audit Elective				0

Semester 3						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PE-MTCAIM-301	Elective	Program Elective				4
OE-MTCAIM-302	Elective	Open Elective				4
EE-MTCAIM-303	EE	Dissertation Phase I				14

Semester 4						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
EE-MTCAIM-401	EE	Dissertation Phase II				18

Elective Courses

Elective Courses for PE Category						
Course Code	**Course Title**	**L**	**T**	**P**	**Cr**	**For Sem./Code**
DOAI250009	Big Data Analytics	3	0	2	4	104
DOAI250033	Machine Learning for Cyber Security	3	1	0	4	104
DOAI250043	Pattern Recognition	3	0	2	4	104
DCSE250009	Programming Framework for Machine Learning	2	0	4	4	105
DOAI250019	Data Mining and Warehousing	3	0	2	4	105
DOAI250023	Data Visualization	2	0	4	4	105
DOAI250041	Natural Language Processing	3	0	2	4	203
DOAI250049	Social Media Analytics	3	1	0	4	203
DOEE250121	Internet of Things	3	0	2	4	203
DOAI250027	Generative AI	3	0	2	4	301
DOAI250042	Optimisation Techniques	3	1	0	4	301
DOAI250048	Reinforcement Learning	3	0	2	4	301

Elective Courses for OE Category
Students may opt courses under MOOC, NPTEL, SWAYAM, NSQF/NCVET aligned courses or other relevant technical subjects not included in their core curriculum. The exercise of option shall be subject to the Course Schedule of the respective Constituent Unit, the credit requirements and availability of seats. Guidelines for choosing MOOC/Online courses are given separately and shall have to be adhered to.

Elective Courses for AU (Audit) Category					
Course Code	**Course Title**	**L**	**T**	**P**	**Cr**
DASH240027	Disaster Management	2	0	0	0
DASH240047	English for Research Paper Writing	2	0	0	0
DASH240052	Environmental Science	3	0	0	0
DASH240085	Pedagogy Studies	2	0	0	0
DASH240086	Personality Development through Life Enlightenment Skills	2	0	0	0
DASH240097	Sanskrit for Technical Knowledge	2	0	0	0
DASH240103	Stress Management by Yoga	2	0	0	0
DASH240104	Technical Presentation and Report Writing	0	0	2	0
DASH240106	Value Education	2	0	0	0
DASH240124	Constitution of India - AU	2	0	0	0
DASH250023	Constitution of India	2	1	0	0
DASH250027	Disaster Management	3	0	0	0
DASH250034	Energy and Environmental Engineering	3	0	0	0
DASH250047	English for Research Paper Writing	2	0	0	0
DASH250054	Essence of Indian Knowledge and Tradition	2	0	0	0
DASH250085	Pedagogy Studies	2	0	0	0

DASH250086	Personality Development	2	0	0	0
DASH250097	Sanskrit for Technical Knowledge	3	0	0	0
DASH250103	Stress Management by Yoga	3	0	0	0
DASH250106	Value Education	2	0	0	0
DASH250117	Innovative Thinking and Entrepreneurship Skills	1	0	0	0
DASH250118	Intellectual Property Rights	2	0	0	0
DCSA240080	Training on Computer Networking	0	0	2	0
DCSA240304	Lab Based on Statistical Package	0	0	2	0
Any other Scheduled Course as per the Course Schedule of the respective Constituent Unit can also be chosen, subject to availability of seats. However, there shall be no assessment and grading for the course chosen as Audit Course.					

The choice of all type of Electives available shall be as per the Course Schedule of the respective Constituent Unit.

Guidelines for Massive Open Online Courses (MOOC) / approved Online Courses

1. Relevant Swayam, MOOC, NPTEL, NIELIT NSQF and any other online courses approved by UGC/AICTE shall also be allowed as Open Electives(OE).

2. One credit of MOOC course should be minimum of 30 hours.

3. A student can choose any approved online course as Open Elective, subject to minimum credit requirements.

4. The students willing to opt OE under MOOC in their next semester, will submit their request to the MOOC Coordinator at their respective campus.

5. Once approved, the campus shall display the list of accepted courses.

6. The student has to submit certificate issued by the institution offering the MOOCs course, along with the number of credits and grades, to get credits transferred into his/her marks certificate issued by NDU.

7. The transfer of credits shall be as adopted by NDU.

Detailed Syllabus

Course Code	Course Name	L	T	P	Cr
DOAI250040	Mathematics for Machine Learning	3	1	0	4

Course Outcomes:

CO1: Grasp core linear algebra and calculus concepts essential for machine learning.
CO2: Analyze and build simple machine learning models using mathematical principles.
CO3: Perform optimization of loss functions using gradient-based methods.
CO5: Apply regularization and matrix decomposition to improve model performance.

Module 1:

Foundations of Linear Algebra
Vectors and matrices: operations, transpose, dot product
Norms and projections
Linear transformations: rotation, scaling, reflection
Determinants, rank, and matrix inverse
Eigenvalues and eigenvectors

Module 2:

Vector Spaces and Systems of Equations
Systems of linear equations and Gaussian elimination
Representation in matrix form
Gaussian elimination and row echelon form
Linear independence

Module 3:

Differential Calculus and Optimization
Functions, limits, and continuity
Derivatives and partial derivatives
Gradient and chain rule
Backpropagation basics
First-order optimization: Gradient Descent
Second-order optimization: Newton’s method
Visual interpretation of function optimization

Module 4:

Linear Models and Regularization
Simple and multiple linear regression
Cost functions and analytical solutions using Normal Equation
Assumptions in linear models
Regularization: Ridge and Lasso Regression
Model evaluation: MSE, R^2 score

Module 5:

Mathematical Tools for Advanced ML Models
Introduction to Information Theory: Entropy, KL Divergence

Support Vector Machines (SVM): Geometry and math foundations
Mathematical foundations of neural networks: Activation functions, loss functions, weight updates

Labs / Practicals:

1.Compute vector projections and dot products using NumPy.
2.Visualize linear transformations on 2D vectors.
3.Find eigenvalues and eigenvectors of given matrices.
4.Use Gaussian elimination to solve a system of equations.
5.Plot 3D surfaces and gradient vectors of loss functions.
6.Implement linear regression.
7.Apply Ridge and Lasso regression to real-world datasets using Scikit-learn.
8.Calculate and compare MSE and R^2 for models with and without regularization
9.Compute entropy and KL divergence between distributions.
10.Visualize SVM decision boundaries in 2D datasets.
11.Manually implement a simple neural network with forward and backward pass using NumPy.

References:

1.Higher Engineering Mathematics, By B.S. Grewal, Khanna Publishers
2.Mathematics for Machine Learning, By Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong, Cambridge University Press
3.Linear Algebra and Learning from Data, By Gilbert Strang, Wellesley-Cambridge Pres
4.The Elements of Statistical Learning, By Trevor Hastie, Robert Tibshirani, and Jerome Friedman, Springer
5.Introduction to Machine Learning, By E. Alpaydin , PHI
6.Pattern Recognition and Machine Learning, By Christopher M. Bishop, Springer
7.Deep Learning, By Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT Press

Course Code	Course Name	L	T	P	Cr
DOAI250032	Machine Learning	3	0	2	4

Course Outcomes:

CO1: Identify and extract relevant features from raw IoT data suitable for applying appropriate Machine Learning (ML) techniques.
CO2: Evaluate and compare the strengths and limitations of various ML algorithms across diverse application domains, particularly in IoT.
CO3: Select and implement appropriate ML models based on data characteristics and application requirements.
CO4: Mathematically formulate and analyze ML models and algorithms, including their underlying assumptions and performance metrics.
CO5: Design and validate machine learning-based solutions for real-world problems in IoT, demonstrating practical understanding and technical proficiency.

Module 1:

Module 1
Introduction: Introduction to Machine Learning: Introduction. Different types of learning, Hypothesis space and inductive bias, Evaluation. Training and test sets, cross validation, Concept of over fitting, under fitting, Bias and Variance.
Linear Regression: Introduction, Linear regression, Simple and Multiple Linear regression, Polynomial regression, evaluating regression fit.

Module 2:

Module 2
Decision tree learning: Introduction, Decision tree representation, appropriate problems for decision tree learning, the basic decision tree algorithm, hypothesis space search in decision tree learning, inductive bias in decision tree learning, issues in decision tree learning, Python exercise on Decision Tree.
Instance based Learning: K nearest neighbor, the Curse of Dimensionality, Feature Selection: forward search, backward search, univariate, multivariate feature selection approach, Feature reduction (Principal Component Analysis), Python exercise on KNN and PCA. Recommender System: Content based system, Collaborative filtering based.

Module 3:

Probability and Bayes Learning: Bayesian Learning, Naïve Bayes, Python exercise on Naïve Bayes, Logistic Regression.
Support Vector Machine: Introduction, the Dual formulation, Maximum margin with noise, nonlinear SVM and Kernel function, solution to dual problem.

Module 4:

Artificial Neural Networks: Introduction, Biological motivation, ANN representation, appropriate problem for ANN learning, Perceptron, multilayer networks and the back propagation algorithm.

Module 5:

Ensembles: Introduction, Bagging and boosting, Random forest, Discussion on some research papers.
Clustering: Introduction, K-mean clustering, agglomerative hierarchical clustering, Python exercise on k-mean clustering.

Labs / Practicals:

1 Write a Python program that imports a dataset of your choice, displays the first five rows, and prints the data type of the DataFrame as a whole. Then, generate and display the statistical summary (mean, median, standard deviation, etc.) for each column and provide complete information about the DataFrame.

2 Write a Python program to import (or create) a dataset with missing values, display the count of missing values per column, and handle them using both imputation and dropping methods. Then, compare the dataset before and after handling missing values, and display the final cleaned dataset.
3 Write a Python program to load (or create) a dataset, split it into training and testing sets, and standardize the features using StandardScaler from sklearn. preprocessing. Finally, display the dataset before and after standardization to observe the changes.
4 Write a Python program to load (or create) a dataset, separate it into predictor variables (X) and target variable (y), and then split the dataset into training and testing sets. Finally, print both the training and testing datasets.
5 Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Python ML library classes in the program.
6 Write a Python program to load (or create) a dataset with missing values, count missing values in each column, and handle them by replacing with NaN, the column mean, and the most frequent value. Finally, display the dataset before and after performing the replacements.
7 Write a Python program to load (or create) a dataset, split it into training and testing sets, and apply feature scaling to the data. Finally, display the dataset before and after scaling to observe the changes.
8 Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Python ML library classes can be used for this problem.
9 Implement the working principle of Artificial Neural Network with feed forward and feed backward principle.
10 Implement and demonstrate the working of SVM algorithm for classification.

References:

1. Kevin P. Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press, 2012.
2. C. M. Bishop. Pattern Recognition and Machine Learning. First Edition. Springer, 2006. (Second Indian Reprint, 2015).
3. Tom M. Mitchell, “Machine Learning”, McGraw-Hill, 2010
4. P. Flach. Machine Learning: The Art and Science of Algorithms that Make Sense of Data. First Edition, Cambridge University Press, 2012.

Course Code	Course Name	L	T	P	Cr
DOAI250004	Artificial Intelligence	3	1	0	4

Course Outcomes:

CO1: Understand AI foundations, intelligent agents, and problem-solving formulations.
CO2: Apply uninformed and heuristic search strategies, and adversarial game-playing techniques.
CO3: Represent knowledge using logic, rules, and reasoning methods.
CO4: Analyze different learning approaches including neural networks and connectionist models.
CO5: Design and evaluate expert systems for domain-specific knowledge representation and reasoning.

Module 1:

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

Module 2:

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

Module 3:

Knowledge Representation: Using Predicate logic, representing facts in logic, functions and predicates, Conversion to clause form, Resolution in propositional logic, Resolution in predicate logic, Unification.
Representing Knowledge Using Rules: Procedural Versus Declarative knowledge, Logic Programming, Forward versus Backward Reasoning

Module 4:

Learning: What is learning, Rote learning, Learning by Taking Advice, Learning in Problem-solving,
Learning from example: induction, Explanation-based learning.
Connectionist Models: Hopfield Networks, Learning in Neural Networks, Applications of Neural Networks, Recurrent Networks. Connectionist AI and Symbolic AI.

Module 5:

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

Labs / Practicals:

NA

References:

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

Course Code	Course Name	L	T	P	Cr
DOAI250024	Deep Learning	2	0	4	4

Course Outcomes:

CO1: Understand the key concepts and training techniques of neural networks.
CO2: Apply deep learning models like CNNs and RNNs to solve real-world problems.
CO3: Gain expertise in advanced models like transformers and generative models for diverse applications.

Module 1:

Introduction to Neural Networks: Basics of Artificial Intelligence and Machine Learning, Introduction to Neural Networks and Biological Inspiration, Perceptron Model and Multilayer Perceptrons (MLP), Activation Functions (Sigmoid, ReLU, Tanh, Softmax), Gradient Descent and Backpropagation

Module 2:

Deep Feedforward Networks: Deep Learning vs. Traditional Machine Learning, Architecture of Feedforward Neural Networks (FNN), Weight Initialization Techniques, Optimization Techniques (SGD, Adam, RMSprop), Vanishing and Exploding Gradient Problems
Convolutional Neural Networks (CNNs): Introduction to Convolutional Neural Networks, Convolutional Layers, Pooling, and Padding, Popular Architectures (LeNet, AlexNet), Applications of CNNs in Image Processing

Module 3:

Recurrent Neural Networks (RNNs) and LSTMs: Introduction to Sequential Data and RNNs, Backpropagation Through Time (BPTT), Vanishing Gradient Problem in RNNs, Long Short-Term Memory (LSTMs) and Gated Recurrent Units (GRUs).

Module 4:

Generative Models and Autoencoders: Introduction to Generative Models, Autoencoders and Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), Applications in Image Generation and Data Augmentation

Module 5:

Generative Models and Autoencoders: Introduction to Attention Mechanisms, Transformer Architecture, Self-Attention and Multi-Head Attention, BERT, GPT, and Large Language Models, Applications in NLP and Computer Vision

Labs / Practicals:

Preparing

References:

1. Charu C. Aggarwal, “ Neural Networks and Deep Learning”, Springer
2. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", MIT Press
3. Rowel Atienza , “Advanced Deep Learning with TensorFlow 2 and Keras”, Packt Publishing
4. Lewis Tunstall, Leandro von Werra, Thomas Wolf, "Natural Language Processing with Transformers"
5. "Deep Learning for Natural Language Processing" – Palash Goyal, Sumit Pandey, Karan Jain
6. "Fundamentals of Deep Learning" – Nikhil Buduma, Nicholas Locascio

Course Code	Course Name	L	T	P	Cr
DCSE250001	Advanced Algorithms and Analysis	3	1	0	4

Course Outcomes:

CO1: Learn core algorithmic techniques: greedy, dynamic programming, backtracking, and branch-and-bound.
CO2: Understand computational complexity (P, NP, NP-hard, NP-complete) and heuristic/randomized methods.
CO3: Apply analytical tools like probabilistic inequalities, amortized analysis, and competitive analysis.
CO4: Explore advanced applications in geometry, graph algorithms, network flows, and string processing.
CO5: Gain exposure to approximation and parallel algorithms, including linear programming, heuristics, and PRAMs.

Module 1:

Defining Key Terms: Algorithm complexity, Greedy method, Dynamic Programming, Backtracking, Branch-and-bound Techniques; Examples for understanding above techniques; Memory model, linked lists and basic programming skills.

Module 2:

Overview - Class P - Class NP - NP Hardness - NP Completeness - Cook Levine Theorem - Important NP Complete Problems. Heuristic and Randomized algorithms.

Module 3:

Use of probabilistic inequalities in analysis, Amortized Analysis - Aggregate Method - Accounting Method - Potential Method, competitive analysis, applications using examples.

Module 4:

Point location, Convex hulls and Voronoi diagrams, Arrangements, graph connectivity, Network Flow and Matching: Flow Algorithms - Maximum Flow – Cuts - Maximum Bipartite Matching - Graph partitioning via multi-commodity flow, Karger'r Min Cut Algorithm, String matching and document processing algorithms.

Module 5:

Approximation algorithms for known NP hard problems - Analysis of Approximation Algorithms Use of Linear programming and primal dual; local search heuristics; Parallel algorithms: Basic techniques for sorting, searching, merging, list ranking in PRAMs and Interconnection.

Labs / Practicals:

NA

References:

1. Michael T Goodric and Roberto Tamassia, “Algorithm Design: Foundations, Analysis and Internet Examples”, John Wiley and Sons, 2002.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, “Introduction to Algorithms”, Third Edition, The MIT Press, 2009.
3. Sanjoy Dasgupta, Christos Papadimitriou and Umesh Vazirani, “Algorithms”, Tata McGraw-Hill, 2009.
4. Cormen, Leiserson, Rivest, and Stein. Introduction to Algorithms. 2nd ed. Cambridge, MA: MIT Press, 2001.David P. Williamson and David B. Shmoys, The Design of Approximation Algorithms

Course Code	Course Name	L	T	P	Cr
DASH250095	Research Methodology and IPR	3	1	0	4

Course Outcomes:

CO1: Understand the principles and process of research methodology.
CO2: Apply appropriate research design and data collection methods in computing projects.
CO3: Analyze and interpret qualitative and quantitative data using statistical tools.
CO4: Develop ethically sound, well-documented research or project reports conforming to academic and industrial standards.
CO5: Understand various forms of intellectual property and apply processes for protection and patent filing.

Module 1:

Introduction to Research and Research Ethics
Definition and objectives of research, types of research – fundamental, applied, exploratory, empirical, significance of research in computer applications, research ethics and integrity, plagiarism, citation styles (APA, IEEE), tools for plagiarism check.

Module 2:

Research Design and Data Collection
Research problem identification and formulation, hypothesis generation, literature review techniques, types of research design: experimental, descriptive, analytical, sampling techniques, primary and secondary data sources, survey design, case study design.

Module 3:

Data Analysis and Interpretation
Quantitative and qualitative data analysis, measures of central tendency and dispersion, t-test, ANOVA, correlation and regression, chi-square test, software tools: Excel, R, SPSS (demo-level), data visualization and result interpretation.

Module 4:

Technical Writing and Project Report Preparation
Structure of research papers and reports, abstract, introduction, literature review, methodology, results, and conclusions, referencing tools (Mendeley/Zotero), IEEE/ACM/SCI journal guidelines, thesis and dissertation formatting, proposal writing basics.

Module 5:

Intellectual Property Rights and Patent Process
Introduction to IPR: patents, copyrights, trademarks, industrial design, Indian and international patent systems (WIPO, TRIPS), criteria for patentability, patent filing process in India (IPINDIA portal), open-source licenses, patent drafting basics, IPR in software and AI innovations.

Labs / Practicals:

N/A

References:

1. C.R. Kothari & Gaurav Garg – Research Methodology: Methods and Techniques, New Age International.
2. Ranjit Kumar – Research Methodology – A Step-by-Step Guide, Sage Publications India.
3. P. Narayan – Intellectual Property Law, Eastern Book Company.
4. Neeraj Pandey & Khushdeep Dharni – Intellectual Property Rights, PHI Learning.
5. Yogesh Kumar Singh – Fundamentals of Research Methodology and Statistics, New Age International.
6. Bharat Bhushan – Research Methodology in Computer Science, Himalaya Publishing.
7. WIPO and IPIndia Manuals – Guidelines for Filing Patents in India.

Course Code	Course Name	L	T	P	Cr
DOAI250009	Big Data Analytics	3	0	2	4

Course Outcomes:

CO1: Describe big data characteristics, challenges, and applications across industries and domains.

CO2: Evaluate and apply NoSQL databases and distributed storage architectures for big data management.

CO3: Install, configure, and operate Hadoop and HDFS for batch-oriented big data processing.

CO4: Develop and implement data analysis workflows using Hadoop ecosystem tools including Hive, Pig, and HBase.

CO5: Design and implement real-time data processing and analytics using Spark Streaming and PySpark.

CO6: Apply visualization and regression techniques for analytical insights from large-scale data.

Module 1:

Introduction to Big Data Ecosystem

Big Data Characteristics: Volume, Variety, Velocity, Veracity, and Value, Business applications and case studies of big data analytics, Challenges in conventional systems, Big Data Analytics Lifecycle, Tools and Technologies in Big Data, Hadoop vs. Traditional Systems, NoSQL Overview (Key-Value, Document, Columnar, Graph databases), Basics of MongoDB and Cassandra for big data storage.

Module 2:

Hadoop Framework for Big Data

History and Evolution of Hadoop, Hadoop Distributed File System (HDFS): Architecture, Internals, and Operations, Block-level Storage, Fault Tolerance, MapReduce Paradigm: Concepts, Anatomy of a MapReduce Job, Failures, Scheduling, Shuffle, Sort, and Task Execution, Input / Output Formats in MapReduce, Hadoop Streaming and Integrations, Developing MapReduce Applications with Java and Python APIs.

Module 3:

Data Analysis with Hadoop Ecosystem

Hive: Data Modeling, Data Types, File Formats, HiveQL (DDL, DML, DQL), Querying Structured Data, Hive Services and Optimization.
Pig: Data Flow Language, Pig Latin Programming, Grunt Shell, Data Models, Operators, Joins, Developing and Testing Pig Scripts.
HBase: Architecture, Data Model, Integration with Hadoop, Use Cases.
Introduction to Data Lakes using Hadoop.
Hands-on with Hadoop Ecosystem for ETL pipelines.

Module 4:

Real-Time Big Data Analytics and Streaming

Stream Processing Fundamentals, Stream Data Models, Stream Architecture, Concepts of Sampling, Filtering, Counting in Streams, Real-time Sentiment Analysis, Stock Market Analysis using Streams, Introduction to Apache Kafka for Streaming Data Ingestion, Spark Basics: RDD, DataFrames, Spark SQL, In-Memory Computing, PySpark APIs, Spark Streaming for Real-Time Processing, Real-Time Analytics Applications, Case

Studies.

Module 5:

Analytical Models and Visualization for Big Data

Regression Techniques: Simple Linear Regression, Multiple Linear Regression, Model Interpretation for Large-Scale Data, Feature Engineering for Big Data, Visualization Techniques: Interactive Dashboards, Graphical Tools for Big Data (Tableau / PowerBI concepts), Visual Analytics: Trends, Patterns, Correlations, Interaction Models, Real-World Applications of Big Data Visualizations.

Labs / Practicals:

1. Setting up Hadoop, HDFS, and YARN on local and cloud-based environments (AWS / Azure / GCP).

2. Developing basic MapReduce jobs using Java and Python.

3. Implementing ETL pipelines using Hive and Pig.

4. Performing analytics with HBase and Hive.

5. Real-time data streaming with Apache Kafka and Spark Streaming (using PySpark).

6. Implementing basic ML pipelines over Spark (MLlib).

7. Visualizing Big Data using interactive tools (Matplotlib, Plotly, Tableau / PowerBI).

8. Group Project: Building and deploying an end-to-end Big Data Analytics pipeline on a real-world dataset.

References:

1. Tom White, Hadoop: The Definitive Guide, 4th Edition, O'Reilly Media, 2015.

2. Anand Rajaraman, Jeffrey David Ullman, Mining of Massive Datasets, Cambridge University Press, 3rd Edition, 2020.

3. Bill Franks, Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics, John Wiley & Sons, 2012.

4. Jiawei Han, Micheline Kamber, Jian Pei, Data Mining: Concepts and Techniques, 3rd Edition, Morgan Kaufmann / Elsevier, 2011.

5. Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia, Learning Spark: Lightning-Fast Big Data Analysis, 2nd Edition, O'Reilly Media, 2020.

6. Paul Zikopoulos, Dirk deRoos, Krishnan Parasuraman, Thomas Deutsch, James Giles, David Corrigan, Harness the Power of Big Data – The IBM Big Data Platform, Tata McGraw Hill, 2012.

7. Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer, 2007.

8. Da Ruan, Guoqing Chen, Etienne E. Kerre, Geert Wets, Intelligent Data Mining, Springer, 2007.

Official Documentation (Highly Recommended for Labs / Practical Work):

9. Official documentation for Hadoop: https://hadoop.apache.org/

10. Official documentation for Spark: https://spark.apache.org/docs/latest/

11. Official documentation for Hive: https://cwiki.apache.org/confluence/display/Hive/Home

12. Official documentation for Pig: https://pig.apache.org/

13. Official documentation for HBase: https://hbase.apache.org/

14. Official documentation for Kafka: https://kafka.apache.org/documentation/

15. Official documentation for PySpark: https://spark.apache.org/docs/latest/api/python/

Course Code	Course Name	L	T	P	Cr
DOAI250033	Machine Learning for Cyber Security	3	1	0	4

Course Outcomes:

CO1: Understand malware concepts, feature generation, and apply classification techniques for malware analysis.
CO2: Apply anomaly detection methods for intrusion detection, feature engineering, and system design considerations.
CO3: Analyze network traffic using machine learning to build predictive models for attack classification.
CO4: Apply supervised and unsupervised learning techniques to detect and mitigate consumer web abuse.
CO5: Evaluate adversarial machine learning techniques, security vulnerabilities, and attack models such as poisoning and evasion.

Module 1:

MALWARE ANALYSIS: Review to basic security concepts and cryptography primitives
Review to basic machine learning concepts, Understanding Malware, Feature Generation, From Features to Classification

Module 2:

ANOMALY DETECTION: When to Use Anomaly Detection Versus Supervised Learning, Intrusion Detection with Heuristics, Data-Driven Methods, Feature Engineering for Anomaly Detection, Anomaly Detection with Data and Algorithms, Challenges of Using Machine Learning in Anomaly Detection, Response and Mitigation, Practical System Design Concerns

Module 3:

NETWORK TRAFFIC ANALYSIS: Theory of Network Defense, Machine Learning and Network Security, Building a Predictive Model to Classify Network Attacks

Module 4:

PROTECTING THE CONSUMER WEB: Monetizing the Consumer Web, Types of Abuse and the Data That Can Stop Them, Supervised Learning for Abuse Problems, Clustering Abuse, Further Directions in Clustering

Module 5:

ADVERSARIAL MACHINE LEARNING: Terminology, The Importance of Adversarial ML, Security Vulnerabilities in Machine Learning, Algorithms, Attack Technique: Model Poisoning, Attack Technique: Evasion Attack

Labs / Practicals:

NA

References:

1. Clarence Chio, David Freeman "Machine Learning and Security", 1st Edition (2018), O'Reilly Media, Inc.
2. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, Handbook of Applied Cryptography, CRC Press.
3. William Stallings, Cryptography and Network Security: Principles and Practice, 7th Ed. (2017), Prentice Hall of India.

Course Code	Course Name	L	T	P	Cr
DOAI250043	Pattern Recognition	3	0	2	4

Course Outcomes:

CO1: Understand the fundamentals of pattern recognition, probability, discriminant functions, and supervised learning methods.
CO2: Apply clustering techniques for unsupervised learning and validate cluster quality.
CO3: Perform feature extraction and selection using statistical, transform-based, and dimensionality reduction methods.
CO4: Implement classification techniques using HMMs, SVMs, and feature selection methods.
CO5: Apply fuzzy logic and genetic algorithms for pattern classification in real-world case studies.

Module 1:

Overview of Pattern recognition – Basics of Probability and Statistics, Linear Algebra, Linear Transformations, Components of Pattern Recognition System, Learning and adaptation Discriminant functions – Supervised learning – Parametric estimation – Maximum Likelihood Estimation – Bayesian parameter Estimation – Problems with Bayes approach– Pattern classification by distance functions – Minimum distance pattern classifier.

Module 2:

Clustering for unsupervised learning and classification–Clustering concept – C Means algorithm – Hierarchical clustering – Graph theoretic approach to pattern Clustering – Validity of Clusters.

Module 3:

Feature Extraction and Feature Selection: Feature extraction – discrete cosine and sine transform, Discrete Fourier transform, Principal Component analysis, Kernel Principal Component Analysis. Feature selection – class separability measures, Feature Selection Algorithms - Branch and bound algorithm, sequential forward / backward selection algorithms. Principle component analysis, Independent component analysis, Linear discriminant analysis, Feature selection through functional approximation – Elements of formal grammars.

Module 4:

State Machines – Hidden Markov Models – Training – Classification – Support vector Machine – Feature Selection.

Module 5:

Fuzzy logic – Fuzzy Pattern Classifiers – Pattern Classification using Genetic Algorithms – Case Study Using Fuzzy Pattern Classifiers and Perception

Labs / Practicals:

1. Implementation of Minimum Distance Classifier
o Classify patterns using Euclidean distance-based decision rules.
2. Supervised Learning using Bayesian Classifier with Gaussian Assumptions
o Implement Bayesian classification with maximum likelihood parameter estimation.
3. Unsupervised Learning using C-Means Clustering Algorithm
o Perform clustering on multivariate data using the C-Means algorithm.
4. Hierarchical Clustering and Dendrogram Visualization
o Apply hierarchical clustering and plot dendrograms for visual analysis.
5. Feature Extraction using PCA and Kernel PCA
o Extract features and reduce dimensionality using PCA and Kernel PCA.
6. Sequential Forward Feature Selection Algorithm
o Implement and evaluate sequential forward feature selection technique.
7. Design and Training of Hidden Markov Model (HMM)
o Train an HMM and use it for classification of sequential data.

8. Support Vector Machine Classification using scikit-learn
o Train an SVM with linear and nonlinear kernels for binary/multi-class classification.
9. Pattern Classification using Fuzzy Inference Systems
o Implement a fuzzy rule-based classifier using membership functions.
10. Feature Selection using Genetic Algorithms
• Apply genetic algorithm for optimal feature subset selection in classification problems.

References:

1. Andrew Webb, “Statistical Pattern Recognition”, Arnold publishers, London,1999.
2. C.M.Bishop, “Pattern Recognition and Machine Learning”, Springer, 2006.
3. M. Narasimha Murthy and V. Susheela Devi, “Pattern Recognition”, Springer 2011.
4. Menahem Friedman, Abraham Kandel, “Introduction to Pattern Recognition Statistical, Structural, Neural and Fuzzy Logic Approaches”, World Scientific publishing Co. Ltd, 2000.
5. Robe

Course Code	Course Name	L	T	P	Cr
DCSE250009	Programming Framework for Machine Learning	2	0	4	4

Course Outcomes:

CO1. Demonstrate an understanding of machine learning concepts and techniques.
CO2. Perform data preprocessing and exploratory data analysis using Python
CO3. Develop and evaluate machine learning models using Python libraries.
CO4. Apply machine learning algorithms to real-world data problems.
CO5. Integrate data analytics and machine learning models into practical applications.

Module 1:

Introduction to Machine Learning: Definition and importance of machine learning, Types of machine learning: Supervised, unsupervised, and reinforcement learning, Applications of machine learning in various domains.
Python for Data Analysis: Introduction to Python programming, Python libraries for data analysis: NumPy, Pandas, Matplotlib, Data manipulation and visualization using Pandas and Matplotlib.
Data Preprocessing: Data cleaning and transformation, Handling missing values and outliers, Feature scaling and normalization.

Module 2:

Regression: Linear regression, Polynomial regression, Model evaluation metrics: MAE, MSE, RMSE. Classification: Logistic regression, K-Nearest Neighbors (KNN), Decision Trees and Random Forests, Model evaluation metrics: Accuracy, precision, recall, F1-score, ROC-AUC. Model Training and Evaluation: Train-test split and cross-validation, Hyper parameter tuning using GridSearchCV, Overfitting and underfitting.

Module 3:

Clustering: K-Means clustering, Hierarchical clustering, Evaluation of clustering results. Dimensionality Reduction: Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) t-Distributed Stochastic Neighbor Embedding (t-SNE).
Association Rule Learning: Apriori algorithm, Market Basket Analysis, Evaluation metrics for association rules

Module 4:

Ensemble Methods: Bagging and Boosting, Gradient Boosting Machines (GBM), Extreme Gradient Boosting (XGBoost).
Support Vector Machines (SVM): Linear and non-linear SVM, Kernel trick, Model evaluation and tuning.
Neural Networks and Deep Learning: Introduction to neural networks, Building and training neural networks using TensorFlow and Keras, Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN).

Module 5:

Exploratory Data Analysis (EDA): Data visualization techniques, Statistical analysis and hypothesis testing, Identifying patterns and insights from data.
Time Series Analysis: Introduction to time series data, Time series forecasting using ARIMA and Prophet, Evaluating time series models.
Integrating Machine Learning Models: Deployment of machine learning models, Building web applications with Flask and Django, Case studies on real-world applications of machine learning.

Labs / Practicals:

1. Implement and demonstratethe FIND-Salgorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a CSV file.
2. For a given set of training data examples stored in a CSV file, implement and demonstrate the CandidateElimination algorithmto output a description of the set of all hypotheses consistent with the training examples
3. Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data

set for building the decision tree and apply this knowledge to classify a new sample
4. Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
5. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions.
6. Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.
7. Write a program to demonstrate Regression analysis with residual plots on a given data set.
8. Write a program to compute summary statistics such as mean, median, mode, standard deviationand variance of the given different types of data.
9. Write a program to implement k-Means clustering algorithm to cluster the set of data stored in .CSV file.

References:

1. "Python Machine Learning: Machine Learning and Deep Learning with Python, scikitlearn, and Tensor Flow " by Sebastian Raschka and Vahid Mirjalili.
2. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron.
3. "Introduction to Machine Learning with Python", Andreas C. Müller & Sarah Guido

Course Code	Course Name	L	T	P	Cr
DOAI250019	Data Mining and Warehousing	3	0	2	4

Course Outcomes:

CO1: Understand and apply data warehousing concepts and architectures.

CO2: Perform clustering, classification, and prediction for pattern discovery.

CO3: Address issues related to data streams and implement stream mining techniques.

CO4: Analyze and apply web mining methodologies for real-world applications.

CO5: Explore current trends and advancements in distributed warehousing and pattern mining.

Module 1:

Data warehousing components, Building a data warehouse, Data warehouse architecture, DBMS schemas for decision support, Data extraction, cleanup, and transformation tools, Metadata, Reporting, Query tools and applications, Online Analytical Processing (OLAP), OLAP and multidimensional data analysis.

Module 2:

Data mining functionalities, Data preprocessing: cleaning, integration, transformation, reduction, discretization, Concept hierarchy generation, Architecture of typical data mining systems, Classification of data mining systems, Efficient and scalable frequent itemset mining methods, Mining various kinds of association rules, From association mining to correlation analysis, Constraint-based association mining.

Module 3:

Issues in classification and prediction, Classification methods: Decision Trees, Bayesian classification, Rule-based classification, Neural networks: backpropagation, Support Vector Machines (SVM), Associative classification, Lazy learners, Other classification methods, Prediction techniques, Accuracy and error measures, Evaluating classifiers/predictors, Ensemble methods, Model selection.

Module 4:

Types of data in clustering, Categorization of major clustering methods, Partitioning methods, Hierarchical methods, Density-based methods, Grid-based methods, Model-based clustering methods, Clustering high-dimensional data, Constraint-based clustering, Outlier analysis.

Module 5:

Mining objects, Spatial data mining, Multimedia data mining, Text mining, Mining the World Wide Web, Multidimensional analysis of complex data, Descriptive mining of complex data objects.

Labs / Practicals:

1. Design a Data Warehouse Schema

Create Star Schema and Snowflake Schema for business data (sales, e-commerce, etc.).

2. ETL Process Implementation

Perform Extract, Transform, Load (ETL) operations on structured data using SQL-based tools (MySQL / PostgreSQL).

3. Association Rule Mining

Apply the Apriori Algorithm for mining frequent itemsets and generating association rules from transactional datasets.

4. Classification Using Decision Trees

Build, visualize, and evaluate Decision Tree Classifiers on datasets like Iris or Titanic.

5. Clustering Using K-Means

Apply K-Means Clustering on multi-dimensional datasets and visualize the results.

6. Data Preprocessing & Dimensionality Reduction

Handle missing values, normalize data, and apply Principal Component Analysis (PCA).

7. OLAP Operations and Cube Construction

Perform OLAP operations (Roll-up, Drill-down, Slice, Dice) using sample datasets.

8. Web Mining Basics

Analyze web log files for pattern discovery (frequent user paths, sessions).

References:

1. Jiawei Han, Micheline Kamber, Jian Pei, Data Mining Concepts and Techniques, Third Edition, Elsevier, 2011.

2. Alex Berson, Stephen J. Smith, Data Warehousing, Data Mining & OLAP, Tata McGraw-Hill, 10th Reprint, 2007.

3. K.P. Soman, Shyam Diwakar, V. Ajay, Insight into Data Mining: Theory and Practice, PHI, 2006.

4. G.K. Gupta, Introduction to Data Mining with Case Studies, PHI, 2006.

5. Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Introduction to Data Mining, Pearson Education, 2007.

Course Code	Course Name	L	T	P	Cr
DOAI250023	Data Visualization	2	0	4	4

Course Outcomes:

CO1: To introduce the fundamental concepts and methodologies of data visualization.
CO2: To understand various techniques for visualizing different types of data, including time series, networks, and hierarchies.
CO3: To explore the process of acquiring, parsing, and handling data for visualization.
CO4: To develop interactive data visualizations using frameworks like D3.js, Tableau, and Matplotlib.
CO5: To analyze and interpret security-related data through visualization techniques.

Module 1:

INTRODUCTION
Context of data visualization – Definition, Methodology, Visualization design objectives. Key Factors – Purpose, visualization function and tone, visualization design options – Data representation, Data Presentation, Seven stages of data visualization, widgets, data visualization tools.

Module 2:

VISUALIZING DATA METHODS
Mapping - Time series - Connections and correlations - Scatterplot maps - Trees, Hierarchies and Recursion - Networks and Graphs, Info graphics

Module 3:

VISUALIZING DATA PROCESS
Acquiring data, - Where to Find Data, Tools for Acquiring Data from the Internet, Locating Files for Use with Processing, Loading Text Data, Dealing with Files and Folders, Listing Files in a Folder, Asynchronous Image Downloads, Advanced Web Techniques, Using a Database, Dealing with a Large Number of Files. Parsing data - Levels of Effort, Tools for Gathering Clues, Text Is Best, Text Markup Languages, Regular Expressions (regexps), Grammars and BNF Notation, Compressed Data, Vectors and Geometry, Binary Data Formats, Advanced Detective Work.

Module 4:

INTERACTIVE DATA VISUALIZATION
Drawing with data – Scales – Axes – Updates, Transition and Motion – Interactivity - Layouts – Geomapping – Exporting, Framework – D3.js, tableau, Matplotlibsns.

Module 5:

SECURITY DATA VISUALIZATION
Port scan visualization - Vulnerability assessment and exploitation - Firewall log visualization -Intrusion detection log visualization -Attacking and defending visualization systems - Creating security visualization system.- Visual Data Analysis Techniques, Interaction Techniques - Systems and Applications.

Labs / Practicals:

1. Graphical Integrity & Perception
Create accurate vs misleading visuals; understand scale, axis distortion.

2. Data-Ink Ratio & Redesign
Apply Tufte's principles to redesign charts by reducing non-data ink.

3. Bar & Dot Plots
Visualize amounts using grouped bars, stacked bars, dot plots, and heatmaps.

4. Distribution Visualization
Use histograms, density plots, Q-Q plots to explore data distribution.

5. Pie & Proportional Charts
Visualize proportions using pie charts and donut charts; compare with bar plots.

6. Time Series Visualization
Plot individual and multiple time series with different response variables.

7. Geospatial Visualization
Create scatterplot maps, choropleth maps using geographic data.

8. Multivariate Data Visualization
Use scatterplot matrices, bubble charts, parallel coordinates for multidimensional data.

9. Network & Hierarchical Visualization
Build tree maps, sunburst charts, and network graphs for relationships.

10. Visual Analytics Dashboard
Create interactive dashboards in Tableau with filters, KPIs, and insights.

11. Python Library Comparison
Compare matplotlib, seaborn, plotly, and bokeh using the same dataset.

References:

1. Edward R. Tufte, The Visual Display of Quantitative Information, 2nd edition, Graphics Press, 2007
2. Claus O. Wilke, Fundamentals of Data Visualization, O'Reilly, 2019
3. Ben Fry, Visualizing Data: Exploring and Explaining Data with the Processing Environment, O'reilly, 1st edition, 2008
4. A. Kerren, J. T. Stasko, J. Fekete, C. North, Information Visualization, Springer, 2008
5. Alex Campbell, Data Visualization: Ultimate Guide to Data Mining and Visualization, 2020
6. A. Loth, Visual Analytics with Tableau, Willey, 2019
7. Kieran Healy, Data Visualization: A Practical Introduction, Princeton University Press, 2018
8. Alexandru C. Telea, Data Visualization: Principles and Practice, Second Edition, CRC Press, 2015

Course Code	Course Name	L	T	P	Cr
DOAI250041	Natural Language Processing	3	0	2	4

Course Outcomes:

CO1: Understand the history, stages, challenges, and applications of NLP in real-world systems.
CO2: Apply morphological analysis, finite state methods, stemming, and n-gram models for language processing.
CO3: Implement POS tagging, CFG parsing, and sentence-level syntactic analysis using statistical and ML-based methods.
CO4: Analyze word sense disambiguation techniques using machine learning and dictionary-based approaches.
CO5: Apply discourse analysis techniques including reference resolution, pronoun interpretation, and coherence modeling.

Module 1:

History of NLP; Generic NLP system; Levels of NLP; Knowledge in language processing problem; Ambiguity in natural language; Stages in NLP; Challenges of NLP; Role of machine learning; Brief history of the field; Applications of NLP: Machine translation, Question answering system, Information retrieval, Text categorization, text summarization & Sentiment analysis

Module 2:

Morphology analysis survey of English morphology, inflectional morphology & derivational morphology; Regular expressions; Finite automata; Finite state transducers (FST); Morphological parsing with FST; Lexicon free FST, Porter stemmer, N-Grams, N-gram language model, N-gram for spelling correction.

Module 3:

Part-of-Speech tagging (POS); Lexical syntax tag set for English (Penn Treebank); Rule based POS tagging; Stochastic POS tagging; Issues: Multiple tags & words, unknown words, class-based n-grams, HM Model ME, SVM, CRF; Context Free Grammar; Constituency; Context free rules & trees; Sentence level construction; Noun Phrase; Coordination; Agreement; Verb phrase & sub categorization

Module 4:

Attachment for fragment of English sentences, noun phrases, verb phrases, prepositional phrases; Relations among lexemes & their senses; Homonymy, Polysemy based disambiguation & limitations, Robust WSD; Machine learning approach and dictionary-based approach.

Module 5:

Discourse reference resolution; Reference phenomenon; Syntactic & semantic constraints on co reference; Preferences in pronoun interpretation; Algorithm for pronoun resolution; Text coherence; Discourse structure

Labs / Practicals:

Module 1: Introduction to NLP & Applications

1. Experiment 1: Demonstration of NLP Applications

o Implement basic NLP applications like sentiment analysis, text summarization, question answering, and information retrieval using pre-trained models or APIs.

2. Experiment 2: Text Preprocessing Techniques

o Perform tokenization, stop-word removal, stemming, and lemmatization on sample English texts.

Module 2: Morphology and Language Models

3. Experiment 3: Morphological Analysis using Stemming and Lemmatization

o Analyze inflectional and derivational forms of English words using stemmers and lemmatizers.

4. Experiment 4: Implementation of N-Gram Language Models

o Build unigram, bigram, and trigram models from a text corpus and compute probabilities of word sequences.
5. Experiment 5: Spelling Correction using N-Gram Probabilities
o Implement a simple spelling corrector using edit distance and N-gram based word probability estimation.

Module 3: POS Tagging and Grammar Parsing
6. Experiment 6: POS Tagging using Rule-Based and Statistical Methods
o Perform part-of-speech tagging using both rule-based and statistical approaches; evaluate accuracy on tagged corpora.
7. Experiment 7: Constituency Parsing using Context-Free Grammar (CFG)
o Define a CFG and parse given sentences to generate parse trees showing sentence structure.

Module 4: Word Sense Disambiguation
8. Experiment 8: Word Sense Disambiguation using Dictionary-Based Methods
o Use the Lesk algorithm to identify correct word senses in different sentence contexts.
9. Experiment 9: Word Sense Disambiguation using Supervised Learning
o Implement a supervised machine learning model to classify the correct sense of an ambiguous word using labeled data.

Module 5: Discourse and Reference Resolution
10. Experiment 10: Co-reference Resolution and Discourse Analysis
• Resolve pronouns and noun phrase references in a passage and analyze coherence and discourse structure.

References:

1. James Allen. Natural Language Understanding. The Benajmins/Cummings Publishing Company Inc. 1994. ISBN 0-8053-0334-0.
2. Tom Mitchell. Machine Learning. McGraw Hill, 1997. ISBN 0070428077.
3. Cover, T. M. and J. A. Thomas: Elements of Information Theory. Wiley. 1991. ISBN 0-471-06259-6.

Course Code	Course Name	L	T	P	Cr
DOAI250049	Social Media Analytics	3	1	0	4

Course Outcomes:

CO1: Understand foundations of analytics, social media data sources, and methods of data gathering.
CO2: Apply visualization and data mining techniques in social media analytics.
CO3: Implement keyword search, classification, clustering, and transfer learning methods for social media data.
CO4: Analyze network centrality, similarity, and structural measures in social media networks.
CO5: Model and predict individual and collective behavior using social media data.

Module 1:

The foundation for analytics, Social media data sources, Defining social media data, data sources in social media channels, Estimated Data sources and Factual Data Sources, Public and Private data, data gathering in social media analytics.

Module 2:

Introduction, A Taxonomy of Visualization, The convergence of Visualization, Interaction and Analytics. Data mining in Social Media: Introduction, Motivations for Data mining in Social Media, Data mining methods for Social Media, Related Efforts

Module 3:

Introduction, Keyword search, Classification Algorithms, Clustering Algorithms Greedy Clustering, Hierarchical clustering, k-means clustering, Transfer Learning in heterogeneous Networks, Sampling of online social networks, Comparison of different algorithms used for mining, tools for text mining.

Module 4:

Centrality: Degree Centrality , Eigenvector Centrality, Katz Centrality , Page Rank, Betweenness Centrality, Closeness Centrality ,Group Centrality ,Transitivity and Reciprocity, Balance and Status, Similarity: Structural Equivalence, Regular Equivalence

Module 5:

Individual Behavior: Individual Behavior Analysis, Individual Behavior Modeling, Individual Behavior Prediction Collective Behavior: Collective Behavior Analysis, Collective Behavior Modeling, Collective Behavior Prediction

Labs / Practicals:

NA

References:

1. Reza Zafarani Mohammad Ali Abbasi Huan Liu, Social Media Mining, Cambridge University Press,ISBN: 10: 1107018854.
2. Charu C. Aggarwal, Social Network Data Analytics, Springer, ISBN: 978-1-4419-8461-6.
3. Marshall Sponder, Social Media Analytics: Effective Tools for Building, Interpreting, and Using Metrics, McGraw Hill Education, 978-0-07-176829-0.
4. Matthew A. Russell, Mining the Social Web, O'Reilly, 2nd Edition, ISBN:10: 1449367615.
5. Jiawei Han University of Illinois at Urbana-Champaign Micheline Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2nd Edition, ISBN: 13: 978-1-55860-901-3 ISBN: 10: 1-55860- 901-6.
6. Bing Liu, Web Data Mining : Exploring Hyperlinks, Contents and Usage Data, Springer, 2nd Editio

Course Code	Course Name	L	T	P	Cr
DOEE250121	Internet of Things	3	0	2	4

Course Outcomes:

CO 1 Explain the concept of IoT
CO 2 Networking basics for IoT application development
CO 3 Analyze various protocols for IoT
CO 4 Design a PoC of an IoT system using various hardware platforms
CO 5 Apply data analytics and use cloud offerings related to IoT
CO 6 Analyze applications of IoT in real time scenario

Module 1:

Overview of IoT

Introduction to the Internet of Things, IoT system architecture and standards, Networking Basics - TCP/IP, Networking Basics - IP addressing basics (IPV4 and IPV6), Optimizing IP for IoT: From 6LoWPAN to 6Lo, Routing over Low Power and Lossy Networks.

Module 2:

IoT hardware Platforms

IoT Design Methodology – Embedded computing logic – Microcontroller-System on Chips, Hardware platforms for prototyping IoT node- Arduino, Raspberry Pi, NodeMCU, ESP32, ARM Cortex Microcontrollers, IoT mote hardware platforms, Swadeshi RISC V based solutions, Interfacing sensors and actuators with hardware platforms, Developing IoT applications using Raspberry Pi with Python Programming.

Module 3:

IoT connectivity & Protocols

IoT Access Technologies: WiFi, Zigbee, Zwave, Bluetooth,UWB, sub1GHz, LoRaWAN, Sigfox and NB-IoT, Topology and Security of IEEE 802.15.4, 802.15.4g, 802.15.4e, 1901.2a, 802.11ah and LoRaWAN, IoT application level protocols: MQTT, CoAP, XMPP, HTTP/Rest Services, WebSockets.

Module 4:

Data analytics, Cloud and IoT Security

No SQL Databases Vs SQL Databases, Apache web servers, JSON, Open and commercial Cloud solutions for IoT, Python Web Application Frameworks for IoT, IoT data visualisation tools, IoT Security - Need for encryption, standard encryption protocol, lightweight cryptography, Trust models for IoT, ARM Cortex Microcontroller Security, Root Security Services (RSS).

Module 5:

IoT Case studies

Smart Lighting, Smart home, Smart Agriculture, Smart farming, IoT for health care & patient monitoring, Smart and Connected Cities, Building end-to-end smart applications with TinyML.

Labs / Practicals:

Hands-on Sessions on IoT Application Development using IoT Hardware Kits (ESP32/STM32/Rpi etc.)

- IPv4 and IPv6 Implementation
- 6LoWPAN Network Implementation
- Interfacing multiple sensors and actuators with Processor Hardware using different communication protocols and develop an integrated monitoring system.
- Implementation of different IoT connectivity technologies including WiFi, Bluetooth, and BLE for various application scenarios.
- Implementation of MQTT protocol for IoT device communication
- Implementation of CoAP for resource-constrained IoT devices
- Integration of IoT devices with cloud platforms for data analytics

References:

Reference Books

1. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton and Jerome Henry, —IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things, Cisco Press, 2017
2. Alessandro Bassi, Martin Bauer, Martin Fiedler, Thorsten Kramp, Rob van Kranenburg, Sebastian Lange, Stefan Meissner, “Enabling things to talk – Designing IoT solutions with the IoT Architecture Reference Model”, Springer Open, 2016
3. VijayMadisetti , ArshdeepBahga, Adrian McEwen (Author), Hakim Cassimally “Internet of Things: A Hands-on-Approach” ArshdeepBahga& Vijay Madisetti, 2014.
4. Gian Marco Iodice, TinyML Cookbook: Combine artificial intelligence and ultralow-power embedded devices to make the world smarter
5. Olivier Hersent, David Boswarthick, Omar Elloumi , —The Internet of Things – Key applications and Protocols, Wiley, 2012

Course Code	Course Name	L	T	P	Cr
DOAI250027	Generative AI	3	0	2	4

Course Outcomes:

CO1: Understand the scope, applications, and ethical challenges of Generative AI across domains.
CO2: Explain traditional and deep learning-based approaches to language modeling and LLM architectures.
CO3: Analyze transformer models, GPT architectures, and their real-world use cases.
CO4: Apply prompt engineering techniques to design effective prompts for generative models.
CO5: Evaluate ethical implications, address bias, and explore real-world applications and future trends in Generative AI.

Module 1:

- Definition and Scope of Generative AI
- Applications of Generative AI in Various Domains
- Importance of Generative AI in Business, Healthcare, Art, and Science
- Ethical Considerations and Challenges in Generative AI
- Introduction to Language Models and Their Role in AI

Module 2:

- Traditional Approaches to Language Modeling
- Deep Learning-Based Approaches in NLP
- Overview of Popular LLM Architectures: Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) Networks, Autoencoders & Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs)

Module 3:

- Introduction to Transformers and Self-Attention Mechanism
- Understanding GPT and Its Significance in AI
- Pre-training and Fine-Tuning Processes in GPT
- Architecture and Working of GPT Models
- Overview of GPT Variants and Their Use Cases (GPT-3, GPT-4, ChatGPT, etc.)

Module 4:

- Understanding the Concept and Significance of Prompt Engineering
- Strategies for Designing Effective Prompts
- Best Practices for Prompt Engineering in Generative AI
- Experimenting with Different Prompting Techniques
- Real-World Examples and Applications of Prompt Engineering

Module 5:

- Ethical Implications of Generative Models
- Addressing Bias and Fairness in AI Systems
- Ensuring Responsible Use and Deployment of Generative Models
- Use Cases in NLP, Content Generation, and Creative Applications
- Case Studies Highlighting Successful Implementations
- Potential Future Applications and Emerging Trends in Generative AI

Labs / Practicals:

Practical 1: Exploring Generative AI Use Cases
In this practical, students will explore the diverse applications of Generative AI across domains such as healthcare, business, art, and science. They will select one domain and research real-world use cases where Generative AI has made significant contributions. Students will summarize their findings in a short report or

presentation, focusing on the problem solved, the type of generative model used, and the impact of the solution.

Practical 2: Write a Python program to implement a basic Recurrent Neural Network (RNN) using TensorFlow or PyTorch. Train the model on a small text dataset and generate simple text sequences to demonstrate how the model learns patterns in the input.

Practical 3: Use a Python-based tool like TensorFlow Hub or PyTorch to load a pre-trained GAN model and generate images from random noise. Display multiple generated images and experiment by modifying the latent input vectors.

Practical 4: Write a Python program to build an Autoencoder model for text reconstruction. Provide the model with noisy or incomplete input sentences and train it to reconstruct the original version, demonstrating its application in denoising.

Practical 5: Write a Python program to implement a Variational Autoencoder (VAE) for generating synthetic text or image data. Visualize how sampling from different parts of the latent space produces different outputs.

Practical 6: Write a Python program to demonstrate the self-attention mechanism used in Transformer models. Simulate a simple attention block and visualize the attention scores across different input tokens.

Practical 7: Using OpenAI GPT Models (GPT-3 / GPT-4 / ChatGPT)
In this practical, students will interact with large language models such as GPT-3 or GPT-4 via OpenAI Playground or API. They will experiment with generating content such as poems, emails, or stories by modifying parameters like temperature, top-p, and prompt structure. The goal is to understand how GPT models behave and how output quality can be influenced by prompt design.

Practical 8: Designing Effective Prompts for LLMs
Students will learn and apply the concepts of prompt engineering by crafting prompts for different tasks such as summarization, translation, storytelling, and question-answering. They will compare the results of various prompt designs and document the impact of changes in wording, context, and formatting on the model's response.

References:

1. Generative Deep Learning by David Foster, 2nd Edition, O'Reilly.
2. Neural Network Methods for Natural Language Processing by Goldberg, Morgan & Claypool Publishers.

Course Code	Course Name	L	T	P	Cr
DOAI250042	Optimisation Techniques	3	1	0	4

Course Outcomes:

CO1: Understand mathematical foundations and unconstrained optimization methods for solving optimization problems.
CO2: Apply gradient-based optimization techniques and analyze their applications in machine learning and neural networks.
CO3: Implement constrained optimization methods using Lagrange multipliers, KKT conditions, and duality theory.
CO4: Analyze and apply nature-inspired metaheuristic algorithms for AI and ML optimization tasks.
CO5: Utilize optimization techniques in supervised and unsupervised ML applications, including PCA, regression, and SVMs.

Module 1:

Mathematical Foundations and Unconstrained Optimization:
Mathematical Preliminaries: Vector Spaces and Matrices, Calculus Concepts- Gradients, Hessians, Taylor series.
Optimization Basics: Definitions, problem formulation, Necessary and sufficient conditions for local optima, Convex sets and convex functions.
One-Dimensional Optimization: Golden Section Search, Fibonacci Method, Newton's and Secant Methods.

Module 2:

Gradient-Based Optimization and Applications in ML:
Multidimensional Optimization: Steepest Descent Method and convergence analysis, Newton's and Quasi-Newton Methods- DFP, BFGS.
Conjugate Gradient Methods for Quadratic and Non-Quadratic functions.
Solving Systems of Equations via Optimization.
Application in Neural Networks- Backpropagation as gradient descent.

Module 3:

Constrained Optimization Techniques:
Equality and Inequality Constraints: Lagrange Multipliers, Karush-Kuhn-Tucker (KKT) conditions.
Convex Optimization: Duality Theory: Weak and Strong Duality.
Algorithms for Constrained Problems: Projected Gradient Methods, Penalty Methods

Module 4:

Nature-Inspired Metaheuristics for AI & ML:
Introduction to Metaheuristics, Simulated Annealing, Genetic Algorithms, Differential Evolution.
Swarm Intelligence: Particle Swarm Optimization, Ant Colony Optimization, Bee Colony Optimization, Firefly and Bat Algorithms, Comparison with classical methods

Module 5:

Optimization in Machine Learning Applications: Optimization in Supervised and Unsupervised ML, Feature Engineering: PCA and Linear Autoencoder, Stochastic Gradient Descent and Variants (SVRG), Regularized Linear and Logistic Regression, Support Vector Machines (SVM) and Hinge Loss, Kernel Methods and Optimization

Labs / Practicals:

NA

References:

1. Edwin K. P. Chong, Wu-Sheng Lu, Stanislaw H. Zak. – An Introduction to Optimization with Applications to

Machine Learning, 5th Ed., Wiley, 2024.
2. Xin-She Yang – Nature-Inspired Metaheuristic Algorithms, 2nd Ed., Luniver Press, 2010
3. Hamdy A. Taha, Operations Research, Prentice Hall, Pearso.
4. J. S Arora, Introduction to optimum design, 2nd edition, Elsevier India Pvt. Ltd.,
5. S. S Rao, Optimization: theory and application, Wiley Eastern Ltd., New Delhi.

Course Code	Course Name	L	T	P	Cr
DOAI250048	Reinforcement Learning	3	0	2	4

Course Outcomes:

CO1: Understand the fundamentals of reinforcement learning, its elements, scope, and historical development.
CO2: Apply action-value methods and strategies for solving multi-armed bandit problems.
CO3: Model real-world tasks as Markov Decision Processes and derive value-based solutions.
CO4: Implement Monte Carlo methods for prediction and control, including off-policy learning techniques.
CO5: Analyze and evaluate reinforcement learning applications through case studies in games, control, and scheduling.

Module 1:

The Reinforcement Learning Problem: Reinforcement Learning, Examples, Elements of Reinforcement Learning, Limitations and Scope, An Extended Example: Tic-Tac-Toe, Summary, History of Reinforcement Learning.

Module 2:

Multi-arm Bandits: An n-Armed Bandit Problem, Action-Value Methods, Incremental Implementation, Tracking a Nonstationary Problem, Optimistic Initial Values, Upper- Confidence-Bound Action Selection, Gradient Bandits, Associative Search (Contextual Bandits)

Module 3:

Finite Markov Decision Processes: The Agent–Environment Interface, Goals and Rewards, Returns, Unified Notation for Episodic and Continuing Tasks, The Markov Property, Markov Decision Processes, Value Functions, Optimal Value Functions, Optimality and Approximation.

Module 4:

Monte Carlo Methods: Monte Carlo Prediction, Monte Carlo Estimation of Action Values, Monte Carlo Control, Monte Carlo Control without Exploring Starts, Off-policy Prediction via Importance Sampling, Incremental Implementation, Off-Policy Monte Carlo Control, Importance Sampling on Truncated Returns

Module 5:

Applications and Case Studies: TD-Gammon, Samuel’s Checkers Player, TheAcrobot, Elevator Dispatching, Dynamic Channel Allocation, Job-Shop Scheduling.

Labs / Practicals:

Practical 1: Write a Python program to implement a basic Grid World environment, where an agent can move up, down, left, or right to reach a goal. The environment should respond with the new state and a reward after each action.

Practical 2: Write a Python program to simulate the n-Armed Bandit problem with random rewards. Implement a random action-selection strategy and calculate the average reward over multiple steps.

Practical 3: Write a Python program to implement the ε-greedy strategy in the n-Armed Bandit problem. Compare the performance with different ε values and plot the average reward over time.

Practical 4: Write a Python program to estimate action-values using sample averages in a multi-armed bandit setting. Track the convergence of action-value estimates over multiple episodes.

Practical 5: Write a Python program to implement optimistic initial values in the n-Armed Bandit problem. Show how optimism in the face of uncertainty affects the agent’s exploration behavior.

Practical 6: Write a Python program to model a simple Finite Markov Decision Process (MDP). Define a small number of states and actions and simulate transitions and rewards for different policies.

Practical 7: Write a Python program to implement Monte Carlo prediction for estimating the value of states in an episodic environment. Use sample episodes to compute returns and update value estimates.

Practical 8: Write a Python program to implement Monte Carlo control for learning optimal policy in a Grid World environment. Use exploring starts or ε-greedy behavior policy for action selection.

Practical 9: Write a Python program to implement off-policy Monte Carlo control using importance sampling. Evaluate a target policy while following a different behavior policy and analyze convergence.

Practical 10: Write a Python program to simulate Tic-Tac-Toe as a reinforcement learning problem. Define the state, action, and reward structure and simulate random or greedy agent play to learn the winning strategy.

References:

1. Richard S. Sutton and Andrew G. Barto, “Reinforcement Learning-An Introduction”,2nd Edition,The MIT Press,2018
2. Marco Wiering , Martijn van Otterlo Reinforcement Learning: State-of-the-Art (Adaptation, Learning, and Optimization (12)) 2012th Edition.
3. Vincent François-Lavet , Peter Henderson , Riashat Islam, An Introduction to Deep Reinforcement Learning (Foundations and Trends(r) in Machine Learning) , 2019