Department of Computer Science and Engineering (DS)

Honors in Data Science
MLRS-R24 Regulations

Course Structure and Syllabus
Applicable From 2024-25 Admitted Batch

III Year I Semester

Course Code	Course Name	L	T	P	Credits
245HDS6701	Data wrangling	3	0	0	3
245HDS6702	AR and VR	3	0	0	3
245HDS6771	AR and VR Lab	0	0	2	1

III Year II Semester

Course Code	Course Name	L	T	P	Credits
246HDS6703	Reinforcement Learning	3	0	0	3
246HDS6772	Reinforcement Learning Lab	0	0	2	1

IV Year I Semester

Course Code	Course Name	L	T	P	Credits
247HDS6704	Big Data	3	0	0	3
247HDS6705	Topological Data Analysis	3	0	0	3
247HDS6706	Gen AI and Agentic AI				
247HDS6707	Time Series Analysis				
247HDS6708	Knowledge Representation and Reasoning				
247HDS6709	Text and speech Analysis				

IV Year II Semester

Course Code	Course Name	L	T	P	Credits
248HDS6773	Mini-Project	0	0	6	3

III-I

245HDS6701:DATA WRANGLING

III Year B.Tech. DS Honors I Sem **L T P C**
3 0 0 3

Course Overview:
The course Emphasizes practical skills in handling messy datasets, including cleaning missing values, removing duplicates, standardizing formats, transforming and joining datasets, working with semi-structured data (JSON), and optimizing queries**.** Students will learn how to build efficient wrangling pipelines that prepare real-world datasets for analytics, machine learning, and business intelligence applications.

Prerequisites: **Nil**

Course Objectives: **The students will try to learn**
- Importance of **data wrangling** in engineering and data science workflows.
- Provide students with the ability to use **SQL for cleaning, transforming, and preparing data** for analysis and decision-making.
- Develop skills to handle **missing, inconsistent, and duplicate data** using SQL functions and operations.
- Equip students with knowledge of **advanced SQL techniques** such as joins, subqueries, window functions, and views for complex data wrangling tasks.
- Enable students to work with **date/time and semi-structured data (JSON)** for real-world applications.

Course Outcomes: After completion of the course, , students will be able to
- Understand the role of data wrangling in data science and engineering workflows.
- Use SQL to clean, preprocess, and transform messy datasets.
- Apply advanced SQL techniques (joins, subqueries, window functions, views) for wrangling.
- Handle date/time data and semi-structured data (e.g., JSON) effectively.
- Build end-to-end wrangling workflows and optimize queries for large datasets.

UNIT –I **[09 Periods]**
SQL fundamentals for data retrieval: Introduction to relational databases and SQL. Overview of database, importance of SQL used for data wrangling, and basic structure of tables, rows, and columns.

UNIT –II **[10 Periods]**
Basic data retrieval, ORDER BY clauses, Filtering and limiting data, Master the use of logical operators and comparison operators to filter results. Handling unknown data - IS NULL.
Aggregate functions, Grouping and summarizing, Joins. Learn Merging datasets with Set operators.

UNIT –III **[10 Periods]**

Data cleaning and manipulation:

Data type conversion. Use CAST() and CONVERT() functions, Standardizing data with string functions. Handling and imputing missing values - handling NULL values using conditional logic (CASE statements) and functions like COALESCE(), Removing duplicate data.

UNIT –IV **[09 Periods]**

Advanced wrangling techniques: Subqueries and Common Table Expressions (CTEs)- nested queries and use CTEs to simplify and manage complex, multi-step queries, Window functions.

Pivoting and reshaping data. Transform rows into columns and vice-versa.

UNIT –V **[10 Periods]**

Real-world application and query optimization:

End-to-end data wrangling project. Apply all learned skills to a complete, realistic dataset. This includes discovery, cleaning, combining, and preparing data for a final report.

Optimizing query performance. Understand how to write efficient SQL queries by reducing table size, improving joins, and using the EXPLAIN command to analyze execution plans.

TEXT BOOKS:

1. Ben Forta, *SQL in 10 Minutes, Sams Teach Yourself* (Any latest edition), Sams Publishing.
2. Alan Beaulieu, *Learning SQL: Generate, Manipulate, and Retrieve Data* (2nd Edition), O'Reilly Media.
3. Joel Murach, *Murach's MySQL* (3rd Edition), Mike Murach & Associates.

REFERENCE BOOKS:

1. Markus Winand**,***SQL Performance Explained*, Markus Winand Publishing.
1. Anthony Molinaro**,***SQL Cookbook* (2nd Edition), O'Reilly Media.
2. Renee M. P. Teate**,***SQL for Data Scientists: A Beginner's Guide for Building Datasets for Analysis*, Wiley.4
3. Joe Celko**,***SQL for Smarties: Advanced SQL Programming* (5th Edition), Morgan Kaufmann.

245HDS6702: AUGMENTED REALITY AND VIRTUAL REALITY

III Year B.Tech. DS Honors I Sem **L T P C**

3 0 0 3

OVERVIEW

The principles of 3D graphics, Augmented Reality, and Virtual Reality bridge theory with practical development. Students will leverage Unity 3D and C# scripting to create immersive, interactive applications.

PREREQUISITE

Foundational 3D graphics fundamentals with Unity 3D and C# scripting, integrating mathematical concepts and AR/VR exposure to design interactive applications.

COURSE OBJECTIVES

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

- Understand fundamentals of 3D graphics, AR and VR concepts
- Use Unity 3D as a development environment for creating interactive AR and VR applications.
- Improve Programming and scripting skills in C# for creating interactions, animations, and logic in AR/VR applications.
- Apply design considerations, physics, and user interface principles.
- Can develop Industry applications of AR and VR across healthcare, manufacturing, retail, and other domains.

COURSE OUTCOMES: After completion of the course, students should be able to

- Understand the basic concepts of 3D graphics, modeling, lighting, rendering, and AR/VR/MR technologies.
- Use Unity 3D interface, import/create assets, and build AR/VR user interfaces for mobile and headset applications.
- Apply C# programs and scripts for implementing interactivity, animations, and advanced behaviors in AR/VR models.
- Design and develop VR applications considering VR design principles, user interface requirements, and deploy them on devices such as Oculus Quest.
- Implement AR/VR applications in different industry domains and develop a domain-specific prototype application.

UNIT I -Introduction to 3D graphics and AR VR fundamentals [09 Periods]

Introduction to 3D graphics concepts – modeling, file formats – texturing – lighting and rendering – animation – AR / VR and MR (Mixed Reality) concepts – mobile phones and headset applications – basic introduction to ARCore and ARKit.

UNIT II -Exploring the Unity 3D programming toolkit [10 Periods]

Introduction to Unity 3D Interface – Using Unity 3D to create AR and VR applications – Importing and creating assets – User Interfaces for AR and VR – Combining assets into complex models – Adding physics to the interactions - Creating an AR Application and publishing to a mobile device.

UNIT III -C# programming and scripting for AR and VR [09 Periods]

C# programming introduction – data types and classes – programming logic – using C# to write scripts for Unity 3D – Using C# to animate and add advanced interactions to AR and VR models.

UNIT IV -Virtual Reality Application essentials [10 Periods]

Virtual Reality fundamentals – VR design considerations – Using Unity 3D and C# programming to create VR applications – Oculus Quest VR headset fundamentals – User interface considerations - Creating a VR application and publishing to the Oculus VR headsets.

UNIT V -Industry applications of AR and VR [10 Periods]

Industry domains where AR and VR are applied – healthcare, manufacturing, field service, retail and branding – Design considerations while creating domain specific applications – Example industry applications – Writing an industry application from a domain of your choice

TEXTBOOKS

1. Beginning 3D Game Development with Unity: All-in-One, Multi-Platform Game Development
2. Learning C# by Developing Games with Unity
3. Augmented Reality: Where We Will All Live

REFERENCE BOOKS

1. Learning Virtual Reality: Developing Immersive Experiences and Applications for Desktop, Web, and Mobile
2. Programming 3D Applications with HTML5 and WebGL: 3D Animation and Visualization for Web Pages
3. Augmented Reality with Unity AR Foundation

245HDS6771: AUGMENTED REALITY AND VIRTUAL REALITY LAB

IIIYearB.Tech.DS Honors I–Sem **L T P C**

0 0 2 1

Course Overview: This course equipped with the hardware and software necessary to explore and create augmented reality (AR) and virtual reality (VR) experiences, blending digital information with the real world (AR) or simulating entirely digital environments (VR).

Prerequisites: Foundational knowledge in programming (preferably JavaScript), basic 3D concepts

COURSE OBJECTIVES: Using this Course students able to

- Set up and explore Unity and Visual Studio environments for AR/VR development
- Create and manipulate 2D/3D objects with transformations, materials, lighting, and physics interactions.
- Develop interactive AR/VR applications using C# scripting, UI, and game mechanics.
- Implement AR and VR experiences using tools like Vuforia, ARKit, ARCore, and Oculus.
- Build, test, and deploy AR/VR applications for real-world scenarios, ensuring usability and performance.

COURSE OUTCOMES: After completion of the course, students should be able to

- Understand unity environment and interface for creating 2D/3D applications.
- Develop interactive games with physics and animation using unity.
- Implement UI elements and score systems in game environments.
- Explore AR/VR platforms and develop basic immersive applications.
- Design, build, and deploy complete game applications using unity.
- Build and test AR application on mobile device or simulator.

LIST OF EXPERIMENTS

1. Study of tools like Unity, Maya, 3DS MAX, AR toolkit, Vuflora and Blender.
2. Use the primitive objects and apply various projection types by handling camera.
3. Install unity and Visual Studio. Explore unity 2D/3D templates. Overview of unity Editor: Scene, Game, Hierarchy, Project, Inspector, Game Objects, Components.
4. Create simple 3D objects. Apply transformations. Add basic lighting, materials, and shaders. Understand how to use prefabs and assets.
5. Add Rigid body, Colliders, Physics Materials. Apply forces and detect collisions using On Collision Enter. Demonstrate physics joints and triggers.
6. Set up project structure with folders (Materials, Prefabs, Scripts, Scenes). Create environment and moving player. Write movement scripts.

7. Move the camera, build play area boundaries. Create and collect pick-up objects (gems), update score, and display on screen using UI Text.
8. Add an enemy to follow the player using NavMesh or scripting. Create a health bar and display it. Show “Game Over” and “You Win” conditions.
9. Create a complete UI with canvas, buttons, health bar, score, and pop-ups. Script UI interactions using C#.
10. Build the Infinite Runner game into a Windows executable. Package resources. Test for debugging and playability.
11. Define AR & VR with real-time examples. Introduction to tools like Oculus, Vuforia, Kudan, Wikitude, ARKit, and ARCore.
12. Explore VR environment using Oculus. Experience basic interactions and movement in virtual environments.
13. Install Vuforia in unity, generate license key, create and configure database. Place 3D models on image targets.
14. Interact with augmented objects in the real world. Build and test AR application on mobile device or simulator.

TEXTBOOKS

1. Steven M. LaValle, Virtual Reality, Cambridge University Press, 2016.
2. William R. Sherman & Alan B. Craig, Understanding Virtual Reality, Morgan Kaufmann, 2002.
3. Alan B. Craig et al., Developing Virtual Reality Applications, Morgan Kaufmann, 2009.
4. Allan Fowler, AR Game Development, Apress, 2018. ISBN: 978-1484236178.
5. Schmalstieg & Hollerer, Augmented Reality: Principles & Practice, Pearson, 2016. ISBN-10: 933257849.

REFERENCE BOOKS

1. Gerard Jounghyun Kim, Designing Virtual Systems, 2005.
2. Doug A. Bowman et al., 3D User Interfaces: Theory and Practice, Addison Wesley, 2005.
3. Oliver Bimber & Ramesh Raskar, Spatial Augmented Reality, 2005.
4. Grigore C. Burdea & Philippe Coiffet, Virtual Reality Technology, Wiley, 2003.
5. Kharis O’Connell, Designing for Mixed Reality, O’Reilly Media, 2016.
6. Sanni Siltanen, Theory and Applications of Marker-Based AR, 2012. ISBN: 978-951-38-7449-0.

III-II

245HDS6703: REINFORCEMENT LEARNING

III Year B.Tech. DS Honors II Sem **L T P C**
3 0 0 3

Course Overview: Reinforcement Learning (RL) is a branch of Artificial Intelligence and Machine Learning that focuses on how agents learn to make decisions through trial-and-error interactions with an environment. Unlike supervised learning, RL does not rely on labeled data but instead optimizes actions to maximize cumulative rewards.

Prerequisites: Machine Learning

Course Objectives:

- Understand the principles of reinforcement learning and its relation to AI.
- Analyze problems in terms of agents, environments, states, actions, and rewards.
- Apply Markov Decision Processes (MDPs) to model sequential decision-making.
- Implement RL algorithms such as Dynamic Programming, Monte Carlo, Temporal-Difference (TD) Learning, Q-Learning, and Policy Gradient methods.
- Explore advanced methods like Deep Reinforcement Learning (DRL).

Course Outcomes: After completion of the course, students should be able to

- Understand basics of RL.
- Apply RL Framework and Markov Decision Process.
- Analyze Ning through the use of Dynamic Programming and Monte Carlo.
- Implement TD(0)algorithm, TD(λ)algorithm.
- Demonstrate the ability for designing and implementing RL algorithms.

UNIT–I **[10 Periods]**

Basics of probability and linear algebra, Definition of a stochastic multi-armed bandit, Definition of regret, Achieving sub linear regret, UCB algorithm, KL-UCB, Thompson Sampling.

UNIT–II **[10 Periods]**

Markov Decision Problem, policy, and value function, Reward models (infinite discounted, total, finite horizon, and average), Episodic & continuing tasks, Bellman's optimality operator, and Value iteration & policy iteration

UNIT–III **[09 Periods]**

The Reinforcement Learning problem, prediction and control problems, Model-based algorithm, Monte Carlo methods for prediction, and Online implementation of Monte Carlo policy evaluation

UNIT–IV **[09 Periods]**

Bootstrapping; TD(0) algorithm; Convergence of Monte Carlo and batch TD(0) algorithms; Model-free control: Q-learning, Sarsa, Expected Sarsa.

UNIT–V **[10 Periods]**

n-step returns, TD(λ)algorithm, Need for generalization in practice, Linear unction approximation and geometric view, Linear TD(λ). Tile coding, Control with function approximation, Policy search, Policy gradient methods, Experience replay, Fitted Q Iteration, Case studies.

TEXTBOOKS:

1. Reinforcement learning: An introduction, First Edition,Sutton,Richard S., and Andrew G. Barto, MIT press 2020.
2. Statisticalreinforcementlearning:modernmachinelearningapproaches,FirstEditi on,Sugiyama, Masashi. CRC Press 2015.

REFERENCEBOOKS:

1. Bandit algorithms, First Edition, Lattimore, T.andC. Szepesvári. Cambridge University Press. 2020.
2. Reinforcement Learning Algorithms: Analysis and Applications," Boris Belousov, Hany Abdu lsamad, Pascal Klink, Simone Parisi, and Jan Peters First Edition, Springer 2021.
3. Alexander Zaiand Brandon Brown "Deep Reinforcement Learning in Action," First Edition, Manning Publications 2020.

246HDS6772: REINFORCEMENT LEARNING LAB

III Year B.Tech. DS Honors II Sem **L T P C**

0 0 2 1

Course Overview:

Reinforcement Learning (RL) Laboratory focuses on practical implementation of core RL concepts and algorithms. Students will learn to build RL agents to solve decision-making problems in various environments.

Prerequisites:

- A course on Data Structures and Algorithms
- Basics of Probability and Linear Algebra
- Familiarity with Python programming language

Course Objectives: The students will try to learn

- Implementation of Markov Decision Processes and RL algorithms.
- Applying Monte Carlo, TD learning, SARSA, and Q-learning methods.
- Using function approximation for value estimation.
- Developing policy gradient and Actor-Critic methods.
- Experimenting with exploration strategies and analyzing results.

Course Outcomes: After completion of the course, students should be able to

- Develop RL agents for classical and grid-world problems.
- Implement and compare model-free RL algorithms.
- Use function approximation in RL problems.
- Apply policy gradient methods for continuous action spaces.
- Analyze and tune RL algorithms for better performance.

List of Experiments:

Environment Setup & Basic Programming

1. Setup Python environment with necessary RL libraries (NumPy, OpenAI Gym).
2. Implement a simple MDP and simulate agent-environment interaction.

Monte Carlo Methods

3. Implement Monte Carlo Prediction to estimate value functions.
4. Implement Monte Carlo Control with Exploring Starts to find optimal policies.

Temporal Difference Learning

5. Implement TD(0) prediction algorithm and observe learning progress.
6. Implement SARSA (on-policy TD control) for a grid-world environment.
7. Implement Q-Learning (off-policy TD control) and compare with SARSA.

Function Approximation

8. Implement linear function approximation for value estimation.
9. Implement tile coding or feature-based state representation.

Policy Gradient Methods

10. Implement the REINFORCE algorithm for a simple environment.
11. Implement Actor-Critic method combining value and policy updates.

Exploration Strategies

12. Implement ε-greedy exploration and vary ε to observe impact.
13. Implement Upper Confidence Bound (UCB) for multi-armed bandit problem.

Real World Problems and Applications

14. Build an RL agent to solve the Multi-Armed Bandit problem.
15. Train an RL agent to navigate a Grid World with obstacles.
16. Apply Q-Learning to a simple OpenAI Gym environment (e.g., FrozenLake).

Advanced Topics (Optional / Bonus)

17. Implement Eligibility Traces (TD(λ)) and compare with TD(0).
18. Implement Deep Q-Network (DQN) basics using PyTorch or TensorFlow.

Tools and Platforms:

- Python (latest version)
- Libraries: NumPy, OpenAI Gym, Matplotlib, optionally TensorFlow or PyTorch
- Jupyter Notebook or any Python IDE (PyCharm, VSCode, etc.)

References:

1. Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction*, 2nd Edition, MIT Press, 2018.
2. OpenAI Gym Documentation: https://gym.openai.com/docs/
3. Csaba Szepesvári, *Algorithms for Reinforcement Learning*, Morgan & Claypool, 2010.
4. Python RL tutorials and examples on GitHub (various sources).

IV-I

247HDS6704: BIG DATA

(Professional Elective)

IV Year B.Tech DS Honors I Sem **L T P C**

3 0 0 3

Course Overview:

This course introduces students to the concepts, technologies, and practical applications of **Big Data** in modern computing. Students will learn to manage, process, and analyze massive datasets using industry-standard frameworks and tools, gaining both theoretical knowledge and hands-on experience.

Course Objectives:

The students will try to learn

- Basic concepts and significance of Big Data.
- Various tools, technologies, and frameworks for handling Big Data.
- Hadoop architecture, HDFS, and MapReduce programming model.
- Ecosystem components: Pig, Hive, HBase, Spark.
- Applications of Big Data in real-world domains.

Prerequisites: PYTHON, SQL

Course Outcomes:

After completion of the course, students should be able to

- Understand the key concepts, challenges, and applications of Big Data.
- Apply Hadoop ecosystem components for storage and processing of large datasets.
- Use tools like Hive, Pig, and Spark to perform Big Data analytics.
- Develop simple MapReduce and Spark programs for large-scale data processing.
- Analyze case studies for understanding Big Data solutions in industry.

UNIT-I **[10 Periods]**

Introduction to Big Data: Introduction to Big Data – Definition, Characteristics, Challenges, Traditional vs. Big Data approaches, Types of Big Data.

Big Data Technologies: Big Data analytics applications in various domains (healthcare, finance, social media, IoT, e-commerce).

UNIT-II **[10 Periods]**

Hadoop Framework: Hadoop architecture, Hadoop Distributed File System (HDFS), Data replication, Fault tolerance, Hadoop cluster setup.

MapReduce: Programming model, Phases of MapReduce, Examples of MapReduce applications.

UNIT-III **[09 Periods]**

Hadoop Ecosystem:

- Pig – Data flow language, Use cases.

- Hive – Data warehousing concepts, HiveQL.
- HBase – NoSQL database, CAP theorem, HBase architecture and use cases.

UNIT-IV **[10 Periods]**

Big Data Analytics with Spark: Introduction to Apache Spark, RDDs, DataFrames, Spark SQL.

Machine Learning with Spark: Basics of MLlib, Example use cases.

UNIT-V **[09 Periods]**

Big Data Applications and Case Studies:

- Real-time analytics with Kafka and Spark Streaming.
- Case studies in healthcare, social media analytics, recommendation systems, fraud detection.
- Future trends in Big Data

TEXT BOOKS:

1. Tom White, Hadoop: The Definitive Guide, O'Reilly Media, 4th Edition.
2. Rajkumar Buyya, Rodrigo N. Calheiros, Amir Vahid Dastjerdi, Big Data: Principles and Paradigms, Elsevier, 2016.

REFERENCE BOOKS:

1. Seema Acharya, Subhashini Chellappan, Big Data Analytics, Wiley India, 2015.
2. Chris Eaton, Dirk DeRoos, Tom Deutsch, George Lapis, Paul Zikopoulos, Understanding Big Data, McGraw Hill, 2012.
3. Vignesh Prajapati, Big Data Analytics with R and Hadoop, Packt Publishing, 2013.

247HDS6705: TOPOLOGICAL DATA ANALYSIS

(Professional Elective)

IV Year B.Tech DS Honors I Sem **L T P C**
3 0 0 3

Course Overview:

This course explores the intersection of topology—a branch of mathematics focused on the properties of space that are preserved under continuous transformations—and data analysis.

Prerequisites: Basic Mathematics, Machine Learning

Course Objectives:

- Introduce fundamental concepts of topology relevant to data analysis, including simplicial complexes and homology.
- Develop an understanding of persistent homology and its role in capturing multi-scale features of data.
- Equip students with computational tools and software to apply topological methods to real-world datasets.
- Illustrate how topological data analysis (TDA) integrates with machine learning and other data science techniques.
- Encourage critical thinking about the interpretation and limitations of topological summaries in various application domains.

Course Outcomes: After completion of the course, students should be able to

- Explain key topological concepts and their significance in analyzing data structures.
- Construct and analyze simplicial complexes from raw data sets.
- Interpret persistence diagrams and barcodes for identifying meaningful data features.
- Apply topological data analysis methods on practical problems in areas such as biology, image processing, and sensor networks.
- Critically evaluate the advantages and limitations of TDA in different data-driven contexts.

UNIT 1: Introduction to Topology **[10 Periods]**

Mathematical Foundations: Basic concepts of topological spaces, metric spaces, and maps between spaces.

Continuity and Homotopy: Understanding continuous deformations and homotopy equivalence to compare spaces.

UNIT 2: Data-Driven Topology **[09 Periods]**

Distance and Filtration: How to construct topological objects from data points, often using a notion of distance.

Computational Aspects: Methods and algorithms for generating these complexes from data.

UNIT 3: Persistent Homology **[10 Periods]**

Filtration and Persistence: The core idea of tracking topological features (like connected components, holes, and voids) as the filtration level changes.

Persistent Homology Barcodes and Diagrams: Visual representations of persistent homology, showing the "birth" and "death" of features at different scales.

UNIT 4: Mapper and Visualization **[09 Periods]**

The Mapper Algorithm - Clustering and Visualization: How Mapper uses filters and clustering to reveal the underlying structure of data, making it easier to visualize.

Feature Extraction: Using the Mapper graph to identify key regions and features within complex datasets.

UNIT 5: Applications in Engineering **[10 Periods]**

Engineering Data Analysis: Applying TDA techniques to analyze complex, high-dimensional data from various engineering domains.

Feature Analysis: Using persistent homology and Mapper to discover hidden patterns, anomalies, and structural properties in engineering data.

.

Text Books

1. Computational Topology for Data Analysis, Tamal Krishna Dey, Yusu Wang

2. Topological Data Analysis with Applications, Gunnar Carlsson and Mikael Vejdemo-Johansson

References Books

1.Scholarly articles for References for topology data analysis

2. An Introduction to Topological Data Analysis: Fundamental and Practical Aspects for Data Scientists, Frédéric Chazal, Bertrand Michel

247HDS6706:GEN AI AND AGENTIC AI

(Professional Elective)

IV Year B.Tech DS Honors I Sem **L T P C**
3 0 0 3

Course Overview:

This course explores the foundations and applications of Generative and Agentic AI, with a focus on developing autonomous, goal-directed AI systems. Students will learn core concepts, including Large Language Models (LLMs), Retrieval Augmented Generation (RAG), and multi-modal systems, as they progress from content generation to developing intelligent agents that can make decisions and take action.

Prerequisites:Artificial Intelligence

Course Objectives: The students will try to learn

- Grasp the core concepts of both Generative AI and Agentic AI.
- Acquire practical skills in leveraging and integrating Generative AI components
- Master the development and deployment of autonomous AI agents
- Gain proficiency in orchestrating multi-agent systems
- Understand the crucial ethical considerations, governance challenges, and responsible implementation strategies

Course Outcomes: After completion of the course, students should be able to

- Understand foundational concepts of Generative AI
- Analyze the architecture and functioning of Large Language Models (LLMs)
- Apply Generative AI techniques on real-world data
- Implement the concept of Agentic AI and its difference from traditional AI systems
- Design and simulate simple AI agents capable of autonomous decision-making

UNIT1: **[10 Periods]**

Foundations of Generative AI

Generative AI vs. Discriminative Models: Understanding the fundamental differences in model types.

Core Generative AI Techniques: Exploration of Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and Diffusion Models.

UNIT2: Large Language Models (LLMs) and Prompt Engineering [10 Periods]

LLM Architecture: In-depth look at Transformer networks and architectures.

LLM Development and Optimization: Methods for training and fine-tuning LLMs for specific tasks.

LLM Integration: Using APIs like OpenAI and libraries like Hugging Face and LangChain to build applications.

UNIT3: **[09 Periods]**

Retrieval-Augmented Generation (RAG)

RAG Fundamentals: Combining retrieval with generation to enhance AI responses with external, factual data.

Text Processing: Techniques for preparing documents, including text splitting and indexing.

Building RAG Applications: Practical experience developing RAG-enabled models for tasks like chatbots or data retrieval.

UNIT4: **[09 Periods]**

Agentic AI Foundations

Introduction to AI Agents: Defining autonomous agents and exploring their components: Sense, Plan, Act, and Learn.

Agentic Frameworks: Hands-on experience with multi-agent frameworks like LangGraph, CrewAI, and AutoGen.

Single and Multi-Agent Systems: Developing both individual agents and complex, collaborative systems.

UNIT5: **[10 Periods]**

Advanced Agentic AI and Deployment

Agentic RAG: Building sophisticated, knowledge-aware agents by implementing advanced RAG techniques.

Multimodal Agents: Creating agents that can process and generate text, images, and audio.

Ethical AI and Guardrails: Understanding and mitigating risks such as bias, security, and hallucinations in agents.

TEXT BOOKS:

1. Agentic Artificial Intelligence: Harnessing AI Agents to Reinvent Business, Work and Life by Pascal Bornet et al. (2025)
2. AI Agents in Action by Michael Lanham (2025)

REFERENCE BOOKS:

1. Natural Language Processing with Transformers by Lewis Tunstall, Leandro von Werra, and Thomas Wolf
2. Generative AI: Navigating the Course to the Artificial General Intelligence Future by Martin Musiol

247HDS6707: TIME SERIES ANALYSIS

(Professional Elective)

IV Year B.Tech.DS Honors I Sem **L T P C**
3 0 0 3

Course Overview:

This course provides students with the concepts and techniques of Time Series Analysis, Index Numbers, Correlation, and Forecasting, along with stochastic modeling approaches for data-driven decision-making.

Prerequisites: Basic knowledge of probability, statistics, and regression analysis.

Course Objectives:

- Introduce the fundamental concepts of time series data, its characteristics, and components.
- Familiarize students with methods to identify, measure, and analyze trends, seasonality, cycles, and random variations.
- Explain correlation, covariance, and techniques for analyzing seasonal variations.
- Develop skills to construct and evaluate index numbers for economic and business applications.
- Introduceforecasting strategies and stochastic models for modeling and predicting future values.
- Applyregression, exploratory data analysis, and seasonal models to real-world time series problems.

Course Outcomes: After completion of the course, students should be able to

- Explain the fundamental concepts of time series data, its components, properties, and applications in real-world problems.
- Analyze correlation, covariance, and seasonal variations using different statistical methods for interpreting patterns in time series data.
- Construct and evaluate various index numbers (Laspeyres, Paasche's, Fisher's, Marshall-Edgeworth) and apply them for measuring economic and business performance.
- Apply stochastic models such as White Noise, Random Walk, and Autoregressive models for time series modeling and forecasting.
- Develop suitable forecasting strategies using regression, exponential smoothing, Holt-Winters, and seasonal models for supporting decision-making in uncertain environments.

UNIT-I: INTRODUCTION **[10 Periods]**

Time Series Data: Purpose, Time series: Plots, Trends, and seasonal variation, Decomposition of series, Characteristics of Time Series.

Components of Time Series:

Trends, Seasonality, Cycles, Noise, Mathematical Models: Additive and Multiplicative models. Resolving components of a Time Series.

UNIT-II: Correlation **[10 Periods]**

Expectation and the ensemble, correlo gram, covariance of sum of random variables, Measuring Seasonal Variation: Method of Simple Averages, Ratio-to-Trend Method, Ratio-to-Moving Average Method and Link Relative Method, Cyclical and Random Fluctuations, Variate Difference Method.

UNIT-III: Index Numbers and their Definitions **[09 Periods]**

Construction and Uses of Fixed and Chain based Index Numbers, Simple and Weighted Index Numbers, Laspeyres, Paasche's, Fisher's, and Marshall - Edgeworth Index Numbers, Optimum Tests for Index Numbers, Cost of Living Index Numbers.

UNIT-IV: Basic Stochastic Models: **[10 Periods]**

White Noise, Random Walks, Fitted models & Diagnostic plots, **Autoregressive models:** Stationary and non-stationary Auto regressive process Time series Regression and Exploratory Data Analysis.

UNIT-V: Linear Models: **[09 Periods]**

Moving Average models, Fitted MA Models, ARIMA Models: Auto regressive Moving Average Models, Differential Equations, Auto correlation and Partial Correlation, Forecasting & Estimation, Non-stationary Models.

TEXT BOOKS:

1. Introductory Time series with R Paul S.P. Cowpert wait, Andrew V. Metcalfe 1st Edition Springer 2010
2. Time Series Analysis and its Applications with R Examples Robert H Shumway, David S Stoffer 4th Edition O'Reilly 2020

REFERENCEBOOKS:

1. Introduction to Time Series and Forecasting Peter J Broke well, Richard A Davis Third Springer 2016
2. The Analysis of Time Series – An Introduction Chris Chatfield First Chapman & Hall / CRC 1996

247HDS6708:KNOWLEDGE REPRESENTATION AND REASONING (PROFESSIONAL ELECTIVE)

IV Year B.Tech DS Honors I Sem. **L T P C**
3 0 0 3

Course Overview:
This course introduces the foundational concepts and techniques used to represent knowledge in computer systems and perform reasoning to infer new information.

Prerequisites: Artificial Intelligence, Basic programming

Course Objectives:
The students will try to learn
- Fundamental concepts and importance of knowledge representation in AI.
- Logical systems for knowledge representation: propositional and first-order logic.
- Inference and reasoning techniques for drawing conclusions from knowledge bases.
- Representation schemes like semantic networks, frames, ontologies, and description logics.
- Methods to handle uncertainty and reasoning about actions and change.

Course Outcomes:
After successful completion of the course, students should be able to
- Understand and apply formal logic for representing knowledge.
- Perform automated reasoning using various inference techniques.
- Design and implement knowledge representation schemes for real-world problems.
- Apply probabilistic and non-monotonic reasoning on uncertain knowledge.
- Use ontologies and description logics in semantic web and AI applications.

UNIT – I **[10 Periods]**

Introduction and Logical Foundations:
Introduction to Knowledge Representation and Reasoning in AI, Propositional Logic: Syntax, semantics, truth tables, logical equivalences, normal forms, First-Order Logic (FOL): Syntax, semantics, quantifiers, interpretations, Inference in Propositional and First-Order Logic: Resolution, unification, soundness, completeness.

UNIT – II **[09 Periods]**

Inference and Reasoning Techniques:
Deductive reasoning and proof procedures, Inductive and abductive reasoning, On-monotonic reasoning: Default logic, circumscription, auto epistemic logic, Reasoning algorithms and complexity considerations.

UNIT – III **[10 Periods]**

Knowledge Representation Schemes:

Semantic Networks and their limitations, Frames and frame-based systems, Scripts and stereotypical knowledge representation, Conceptual Dependency Theory.

UNIT – IV **[09 Periods]**

Ontologies and Description Logics:

Introduction to Ontologies and their role in Redescription Logics (DL): Syntax, semantics, inference services reasoning tasks: Subsumption, satisfiability, instance checking, Web Ontology Language (OWL) overview.

UNIT – V **[10 Periods]**

Uncertainty and Reasoning about Actions:

Probabilistic Reasoning: Bayesian networks, Markov models, Fuzzy Logic and Dempster-Shafer theory, Reasoning about Actions and Change: Situation calculus, event calculus, The Frame Problem and solutions.

TEXT BOOKS:

1. Ronald Brachman and Hector Levesque, *Knowledge Representation and Reasoning*, Morgan Kaufmann, 2004.
2. Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 4th Edition, Pearson, 2020.

REFERENCE BOOKS:

1. Michael Genesereth and Nils Nilsson, *Logical Foundations of Artificial Intelligence*, Morgan Kaufmann, 1987.
2. Franz Baader et al., *The Description Logic Handbook*, Cambridge University Press, 2003.
3. Judea Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, 1988

247HDS6709:TEXT AND SPEECH ANALYSIS
(Professional Elective)

IV Year B.Tech DS Honors I Sem **L T P C**
3 0 0 3

Course Overview:
This course covers the fundamental concepts and techniques used in the analysis, processing, and testing of speech signals.

Prerequisites:Machine learning, Basic programming

Course Objectives: The students will try to learn
- Fundamentals of text and speech processing.
- Various techniques for text pre-processing, tokenization, and linguistic analysis.
- Speech production and acoustic features of speech signals.
- Methods for speech recognition, synthesis, and analysis.
- Applications of text and speech analysis in real-world systems.

Course Outcomes: After completion of the course, students should be able to
- Understand the basics of natural language text analysis and speech signal processing.
- Apply pre-processing and feature extraction techniques on text and speech data.
- Perform text classification, sentiment analysis, and speech recognition tasks.
- Use probabilistic and machine learning models for text and speech analytics.
- Design small-scale systems that integrate text and speech processing methods.

UNIT – I :Introduction to Text and Speech Processing [10 Periods]
Basics of text analysis: preprocessing, stemming, lemmatization, stop-word removal, Tokenization and n-gram models, Speech production mechanism, anatomy of speech organs.

UNIT – II :Text Analysis Techniques [10 Periods]
Morphological, syntactic, and semantic analysis, Part-of-speech tagging, Named Entity Recognition (NER), Sentiment analysis: lexicon-based and machine learning approaches, Topic modelling: LDA, Latent Semantic Analysis.

UNIT - III :Speech Features and Processing [09 Periods]
Digital representation of speech, Time domain features: energy, zero-crossing rate, Frequency domain features: Fourier analysis, spectrogram, Mel-frequency cepstral coefficients (MFCCs), Linear Predictive Coding (LPC).

UNIT-IV :Speech and Text Modelling [10 Periods]
Hidden Markov Models (HMM) for speech recognition, N-gram language models, smoothing techniques, Word embeddings (Word2Vec, GloVe), Introduction to deep learning approaches in NLP and speech (RNN, LSTM, Transformer).

UNIT-V :Applications of Text and Speech Analysis **[09 Periods]**

- Text-to-speech (TTS) synthesis.
- Chatbots and virtual assistants.

Case studies: sentiment-aware speech interfaces, speech emotion recognition, multilingual systems.

TEXT BOOKS:

1. Daniel Jurafsky and James H. Martin, Speech and Language Processing, Pearson, 3rd Edition (Draft).
2. Chris Manning and Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT Press.

REFERENCE BOOKS:

1. Lawrence Rabiner and Biing-Hwang Juang, Fundamentals of Speech Recognition, Pearson.
2. Steven Bird, Ewan Klein, Edward Loper, *Natural Language Processing with Python*, O'Reilly.
3. Thomas N. Sainath et al., *Deep Learning for Speech and Language*, Springer.

IV-II