ANDHRA KESARI UNIVERSITY ::ONGOLE

Programme: BCA-Hounours Artificial Intelligence & Data Science (Major)
(w.e.f. Academic Year 2025–26)

COURSESTRUCTURE

Year	Semester	Course	Title	Hr/ week	Credits
1	I	1	Computer Fundamentals and Office Automation	3	3
			Computer Fundamentals and Office Automation Lab	2	1
		2	Problem Solving Using C	3	3
			Problem Solving Using C Lab	2	1
	II	3	Python Programming and Data Structures	3	3
			Python Programming and Data Structures Lab	2	1
		4	Artificial & Computational Intelligence	3	3
			Artificial & Computational Intelligence Lab	2	1
2	III	5	Statistical Foundation of AI	3	3
			Statistical Foundation of AI Lab	2	1
		6	DBMS	3	3
			DBMS Lab	2	1
		7	Exploratory Data Analysis & Data Visualization	3	3
			Exploratory Data Analysis & Data Visualization Lab	2	1
	IV	8	Data Science with R	3	3
			Data Science with R Lab	2	1
		9	Foundation of ML & Supervised Learning	3	3
			Foundation of ML & Supervised Learning Lab	2	1
		10	Robotics Principles & Embedded systems	3	3
			Robotics Principles & Embedded systems Lab	2	1
3	V	11	Business Intelligence Tools	3	3
			Business Intelligence Tools Lab	2	1
		12 A	Big Data Technologies	3	3
			Big Data Technologies Lab	2	1
		(OR)			
		12 B	Natural Language Processing	3	3
			Natural Language Processing Lab	2	1
		13 A	Cloud Computing for Data Science	3	3
			Cloud Computing for Data Science Lab	2	1
		(OR)			
		13 B	Conversational AI	3	3
			Conversational AI Lab	2	1
	VI	14 A	Neural networks and Deep Learning	3	3
			Neural networks and Deep Learning Lab	2	1
		(OR)			
		14B	Time Series Analysis and Forecasting	3	3
			Time Series Analysis and Forecasting Lab	2	1
		15 A	Robotics Kinematics & Dynamics	3	3
			Robotics Kinematics & Dynamics Lab	2	1
		(OR)			
		15 B	Data Engineering & ML Ops	3	3
			Data Engineering & ML Ops Lab	2	1

U. Sarala
(U Sarala) DSGDCW, ongole

SEMESTER-V

COURSE 11: BUSINESS INTELLIGENCE TOOLS

Theory	Credits: 3	3 hrs/week

Course Objectives:

1. Introduce foundational concepts of Business Intelligence (BI) and Decision Support Systems (DSS), including their scope, evolution, and organizational relevance.
2. Familiarize students with leading BI tools such as Power BI and Tableau, highlighting their ecosystems, interfaces, and comparative strengths.
3. Develop skills in data preparation and transformation, using Power Query and Tableau's data connection features to clean and model datasets.
4. Enable effective data visualization and storytelling, leveraging charts, dashboards, and advanced features to communicate insights.
5. Equip learners with data modeling techniques, including dimensional modeling, relationships, joins, and governance principles for robust BI solutions.

Course Outcomes:

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

1. Differentiate between BI, Data Analytics, and Data Science, and explain the BI lifecycle and its applications across functional domains.
2. Use Power BI and Tableau to prepare, transform, and visualize data, applying basic DAX functions and calculated fields for analysis.
3. Design and implement dimensional data models, including star and snowflake schemas, and apply relationships and joins in BI tools.
4. Create interactive dashboards and visualizations, incorporating parameters, slicers, filters, and drilldowns to enhance decision-making.
5. Build and publish complete BI dashboards, and effectively communicate business insights through storytelling and visualization best practices.

Unit-I: Introduction to Business Intelligence and Decision Support Systems

Business Intelligence: Definition, Scope, and Evolution, Business IntelligenceI vs. Data Analytics vs. Data Science, BI Lifecycle, **Applications of BI in Functional Domains:** Finance, HR, Marketing, Retail, Education, Healthcare, etc., BI Maturity Models &

Organizational Readiness to BI adoption Decision Support Systems (DSS): Concepts, Components, and Architecture.

BI Tools Overview: Power BI, Tableau, and other tools, Comparison and suitability of BI Tools.

Case Study: Retail Chain's BI Strategy to Optimize Inventory

Unit-II: Data Preparation and Visualization with Power BI

Introduction to Power BI, Power BI Ecosystem: Desktop, Service, Mobile; Power BI Interface; Data Sources: Excel, CSV, SQL Server, Web APIs, Power Query: Data Preparation, Cleaning & Transformation - Connect, transform, and model a dataset; Basic DAX Functions: SUM, COUNT, AVERAGE, CALCULATE, IF; Creating Simple Visualizations: Charts, Tables, Cards; Sharing Reports via Power BI Service.

Case Study: Student performance analysis in Higher Education , Analyze Finance Dataset

Unit-III: Data Preparation, Visualization and Storytelling with Tableau

Introduction to Tableau; Characteristics of Tableau; Tableau Architecture and components - Tableau Public, Desktop, Reader, Online, Server; Tableau Interface: Shelves, Marks Card, Views; Tableau extensions, Data Connection and Preparation: Cleaning, Pivoting, Filtering; Calculated Fields and LOD Expressions; Basic Visualizations: Bar, Line, Tree, Geo Maps, Scatter Plots; Storytelling with Tableau, Creating a Tableau story.

Case Study: HR Analytics

Unit-IV: Data Modeling and Relationships in BI Tools

Dimensional Modeling: Dimension, Dimension table, fact, fact table, schema, Star and Snowflake Schemas

Power BI: Relationships, Cardinality, Cross-filtering; Tableau: Joins (Inner, Left, Full), Blending; Data Governance: Metadata, Hierarchies, Quality; Data Model Design Best Practices - Design and implement data model in Power BI and Tableau; HR or Retail data model design and insights

Case Study: Retail BI for Sales Optimization

Unit-V: Dashboard Design and Business Insights

Introduction to Dashboard, when to use dashboards, Dashboard components, Principles of Effective Visualization & Dashboarding, Advanced Visualizations: Parameters, Slicers, Filters, Drilldowns, Graphs and Maps, Dashboard Design: Layout, Alignment, Accessibility

Publishing Dashboards: Power BI Service, Tableau Public; Storytelling and Insight Communication, Build a complete BI dashboard using either tool.

Case Study: Business decision-making scenario (e.g., Sales Forecasting, Budgeting)

Text Books:

1. Decision Support and Business Intelligence Systems (9th ed.). Turban, E., Sharda, R., & Delen, D. (2014), Pearson Education.
2. Learning Tableau 2022: Create effective data visualizations, build interactive dashboards, and transform your data into insights (6th Edition), Milligan, J. N. (2022)., Packt Publishing.
3. Expert Data Modeling with Power BI: Enrich and optimize your data models for reporting and business needs (2nd Edition). Bakhshi, S. (2023), Packt Publishing.

Reference Books:

1. Visual Analytics with Tableau, Loth, A. (2019), Addison-Wesley Professional.
2. The Definitive Guide to DAX: Business intelligence for Microsoft Power BI, SQL Server Analysis Services, and Excel (2nd Edition), Russo, M., & Ferrari, A. (2020), Microsoft Press.

Online Resources:

1. Decision Support Systems Courses – Class Central
2. Advanced Business Decision Support Systems – NPTEL (IIT Kanpur)
3. Power BI Learning Paths – Microsoft Learn
4. Power BI Courses – Coursera
5. Tableau Learning Hub – Official Site
6. Free Tableau Course – Simplilearn
7. Dashboard Design Concepts – DataCamp
8. Business Intelligence with Power BI – Swayam

Activities:

1. Differentiate BI, Data Analytics, and Data Science + BI Lifecycle

Activity: Case Study Analysis: Provide students with a business scenario and ask them to identify how BI, analytics, and data science each contribute. Then, have them map the BI lifecycle stages to that scenario and explain its impact across departments (e.g., finance, marketing).

Evaluation Method: Evaluate on a 10-point scale based on Conceptual differentiation clarity (30%), BI lifecycle accuracy (30%), Application to domains (20%) and Written summary or presentation quality (20%)

2. Prepare, Transform & Visualize Data in Power BI/Tableau (with DAX & Calculated Fields)

Activity: Hands-on Data Challenge: Supply students with a raw dataset (e.g. sales or customer data). Task them with cleaning, transforming, and building visuals in Power BI/Tableau using basic DAX and calculated fields (e.g. profit margins, year-over-year growth).

Evaluation Method: Evaluate for 10-points based on Effective transformation workflow (25%), Correct DAX/formula usage (25%), Visualization relevance & clarity (25%) and Documentation of process (25%)

3. Dimensional Modelling with Star/Snowflake Schemas + Joins/Relationships

Activity: Model Building Lab: Students design a star or snowflake schema based on a retail or HR database. Then, implement the schema in Power BI or Tableau and create relationships between tables to ensure correct joins and data flow.

Evaluation Method: Evaluate on a 10-point scale based on Schema design accuracy (30%), Appropriate schema selection (star vs. snowflake) (20%), Implementation of joins/relationships (30%) and Functional model validation (20%)

4. Create Interactive Dashboards with Advanced Features

Activity: Dashboard Design Workshop: Students build an interactive dashboard using parameters, slicers, filters, and drilldowns to simulate real-time decision-making (e.g., tracking regional product sales or employee performance across departments).

Evaluation Method: Evaluate on a 10-point scale on the basis of Integration of interactive features (30%), Usability and navigation experience (30%), Data-driven insights extracted (20%) and Design polish and layout consistency (20%).

5. Build and Publish BI Dashboards + Business Storytelling

Activity: BI Capstone Project: Students design and publish a complete dashboard solving a real or simulated business problem (e.g. customer churn, supply chain bottlenecks). Include visual storytelling techniques-titles, annotations, color themes, and narratives.

Evaluation Method: Evaluate on a 10-point scale on the basis of Clarity of business insights communicated (30%), Storytelling elements (25%), Dashboard completeness and polish (25%) and Peer or instructor presentation (20%)

SEMESTER-V

COURSE 11: BUSINESS INTELLIGENCE TOOLS

Practical **Credits: 1** **2 hrs/week**

List of Experiments:

1. Exploring BI Tools - Power BI vs Tableau.
2. Create a simple retail dashboard using both tools.
3. **Connecting to Different Data Sources in Power BI**
    a. Learn to connect Excel, CSV, and Web data.
    b. Load a dataset from different formats.
    c. Demonstrate dataset loading and view schema.
4. **Data Cleaning and Transformation using Power Query**
    a. Apply Power Query for cleaning.
    b. Remove duplicates, fill nulls, filter rows.
    c. Submit Power Query steps and cleaned dataset.
5. **Prepare and clean a higher education student performance dataset using Power Query and visualize key academic metrics.**
    a. Connect to a dataset and perform data cleaning, transformation, and reshaping using Power Query.
    b. Visualize academic performance indicators such as GPA trends, pass rates, or subject-wise scores.
    c. Upload file containing the cleaned dataset and relevant academic metric visualizations.
6. **Implementing DAX Functions**
    a. Use DAX to perform basic calculations.
    b. Create calculated columns with SUM, AVERAGE, COUNT, CALCULATE, IF.
    c. Submit DAX expressions with visual output.
7. **Creating Basic Visualizations in Power BI**
    a. Develop simple charts and cards.
    b. Use sales or HR data to create bar, pie charts, and KPIs.
    c. Display Dashboard showing at least 3 chart types.
8. **Tableau Basics and Connecting to Data**
    a. Connect and explore data in Tableau Public.

   b. Load a student performance or any other dataset and preview data.
   c. Upload dataset and share working link.

9. **Visualize employee turnover patterns using Tableau and apply LOD expressions to uncover retention drivers.**
   a. Import and clean HR data for turnover analysis in Tableau.
   b. Apply LOD expressions to identify patterns across departments, roles, and tenure.
   c. Upload cleaned dataset and interactive visualizations highlighting retention insights.

10. **Data Cleaning, Pivoting & Filtering in Tableau**
    a. Prepare data inside Tableau.
    b. Pivot columns, apply filters, and rename headers.
    c. Upload cleaned worksheet.

11. **Creating Visualizations in Tableau**
    a. Use Marks Card, Shelves, and Views for visualization.
    b. Build bar chart, map view, scatter plot.
    c. One dashboard with 3 visuals.

12. **Creating a Tableau Story - (Eg HR or Student Performance or any other)**
    a. Build a narrative with visualizations.
    b. Combine charts into a Tableau story with captions.
    c. Publish and present story with link.

13. **Designing Data Models in Power BI**
    a. Create dimensional models (Star/Snowflake schema).
    b. Use a retail dataset with multiple tables and define relationships.
    c. Submit relationship diagram and schema explanation.

14. **Joins and Blending in Tableau**
    a. Implement data joins and blending techniques.
    b. Combine multiple data tables using inner/left/full joins.
    c. Demonstrate join types with visuals.

15. **Dashboard with Drill-downs, Filters, and Slicers**
    a. Use interactivity in dashboards.
    b. Build a multi-level dashboard in Power BI with drill-through.
    c. Submit .pbix file and link to published dashboard.

SEMESTER-V

COURSE 12 A: BIG DATA TECHNOLOGIES

Practical **Credits: 1** **2 hrs/week**

1. Installation & setup of Hadoop single-node cluster
2. Explore Hadoop directory structure and basic commands (hadoop fs operations)
3. Demonstration of Hadoop architecture components (HDFS, YARN, MapReduce) using sample logs
4. Store and retrieve large files from HDFS (block distribution, replication factor demo)
5. Simulate NameNode/DataNode failure and observe fault tolerance & recovery
6. Configure YARN and run sample applications, observe ResourceManager and NodeManager roles
7. Write a simple MapReduce program for word count
8. Develop a MapReduce job for inverted index creation
9. Perform data analysis using Pig Latin scripts
10. Execute Hive queries for structured data analysis (tables, partitions)
11. Import data from RDBMS into Hadoop using Sqoop
12. Capture and store log/streaming data using Flume
13. Serialize and store datasets in Avro and Parquet formats
14. Build an end-to-end ingestion workflow combining batch (Sqoop) and streaming (Flume)
15. Create and manage tables in HBase (CRUD operations)
16. Demonstrate coordination with ZooKeeper
17. Process HBase datasets using Spark integration with Hadoop

Unit 3. MapReduce & High-Level Tools

MapReduce programming model: map, shuffle, reduce phases, Writing MapReduce applications in Hadoop

High-level abstractions: Hive, Pig, Crunch, and introduction to Spark integration

Unit 4. Data Ingestion & Serialization

Data ingestion pipelines: Sqoop (for RDBMS), Flume (streaming), Data formats & serialization: Avro, Parquet, SequenceFile, Practical ingestion workflows-batch and streaming

Unit 5. NoSQL & Ecosystem Enhancements

Overview of NoSQL within Hadoop ecosystem: HBase, Configuration and usage of ZooKeeper for coordination, Hadoop integration with Spark for data processing

Textbooks

1. Hadoop: The Definitive Guide, Tom White, 4th Edition, O'Reilly
   Free Resource available at piazza-resources.s3.amazonaws.comO'Reilly Media
2. Learning Spark, 2nd Edition, Jules S. Damji, Brooke Wenig, Tathagata Das, Denny Lee, O'Reilly

Reference Books

3. BIG DATA, Black Book TM, DreamTech Press, 2016 Edition.
4. BIG DATA and ANALYTICS, Seema Acharya, SubhasniChellappan , Wiley publications, 2016

Activities:

Outcome: Explain Big Data concepts and Hadoop ecosystem

Activity: Short seminar / presentation on Big Data applications

Evaluation Method: Oral presentation + concept quiz

Outcome: Demonstrate HDFS and YARN architectures

Activity: Hands-on lab to configure HDFS & analyze NameNode/DataNode logs

Evaluation Method: Lab performance + viva

Outcome: Apply MapReduce and high-level tools

SEMESTER-V

COURSE 12 A: BIG DATA TECHNOLOGIES

Theory | **Credits: 3** | **3 hrs/week**

Course Objectives:

1. Introduce students to the concepts, characteristics, and challenges of Big Data.
2. Familiarize students with the Hadoop ecosystem and its core components (HDFS, YARN, MapReduce).
3. Develop practical knowledge of distributed storage and parallel processing in Hadoop.
4. Provide hands-on exposure to data ingestion tools (Sqoop, Flume) and serialization techniques.
5. Enable students to explore NoSQL databases (HBase), coordination services (ZooKeeper), and Hadoop–Spark integration for large-scale data analysis.

Course Outcomes:

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

1. Explain Big Data concepts and challenges along with the role of the Hadoop ecosystem.
2. Demonstrate understanding of HDFS and YARN architectures and their functions in distributed data management.
3. Apply MapReduce and high-level tools (Hive, Pig, Spark) to process and analyze large datasets.
4. Design and implement data ingestion workflows using Sqoop, Flume, and serialization formats like Avro and Parquet.
5. Utilize NoSQL databases and ecosystem enhancements such as HBase, ZooKeeper, and Hadoop–Spark integration for scalable big data solutions.

Unit 1. Foundations of Big Data & Hadoop Ecosystem

Introduction to Big Data: characteristics (volume, variety, velocity, veracity, value), Hadoop Ecosystem Overview: HDFS, MapReduce, YARN, Hadoop Common, Hadoop architecture and use cases

Unit 2. Hadoop Distributed File System (HDFS) & YARN:

Deep dive into HDFS architecture: blocks, NameNode, DataNodes, HDFS file operations, fault tolerance, replication

YARN architecture: ResourceManager, NodeManager, application scheduling

SEMESTER-V

COURSE 12 B: NATURAL LANGUAGE PROCESSING

Theory	Credits: 3	3 hrs/week

Course Objectives:

1. Introduce the foundations of Natural Language Processing and its applications in real-world tasks.
2. Familiarize students with text preprocessing, linguistic analysis, and parsing techniques.
3. Equip learners with methods for information extraction, word representations, and sentiment classification.
4. Explore deep learning techniques for NLP, including RNNs, LSTMs, GRUs, and Transformers.
5. Provide hands-on experience with modern NLP tools (NLTK, spaCy, Hugging Face) for implementing applications such as chatbots, summarization, and document classification.

Course Outcomes:

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

1. Explain the principles, challenges, and applications of NLP and use basic text processing tools.
2. Apply preprocessing techniques (tokenization, stemming, lemmatization) and parsing methods to analyze language structures.
3. Implement information extraction and text representation methods (NER, embeddings, classification pipelines).
4. Build and evaluate deep learning models (RNN, LSTM, GRU, Transformer) for NLP tasks.
5. Utilize pre-trained transformer models (BERT, GPT) with Hugging Face for advanced NLP applications such as summarization, chatbots, and document classification.

Unit 1. Introduction to NLP and Language Fundamentals:

Definition, Goals, and Scope of NLP, Real-world Applications (Assistants, Chatbots, Translation, Summarization, QA, Spam Detection), Fundamentals of Language Processing, Ambiguities in NLP (Lexical, Structural, Contextual)

Installations: Python setup, NLTK, spaCy basics

Regular Expressions (Essential patterns, findall, split, sub, matching tokens)

Unit 2. Text Preprocessing and Linguistic Analysis:

Key NLP Terminologies: Morphology, Lexicon, Orthographic Rules

Finite State Transducers

Text Preprocessing Techniques: Tokenization, Stopword Removal, Stemming, Lemmatization

Grammar and Context-Free Grammar

Parsing Techniques: Top-down, Bottom-up, CYK Algorithm

Semantic Analysis: Elements, Meaning Representation

Unit 3. Information Extraction and Representation:

Named Entity Recognition (NER): Concepts, Examples, Using spaCy & NLTK

Word Embeddings: Word2Vec (Skip-Gram, CBOW), Comparison, Implementations, Bag of Words and N-grams, Text Classification Pipeline, Sentiment Analysis Applications

Ethical considerations in preprocessing & classification

Unit 4. Deep Learning for NLP:

Recurrent Neural Networks (RNN): Basics, RNN vs CNN/Feedforward NN, LSTM and GRU for Sequence Modeling, Transformer Models: Introduction, Pretrained Models (BERT, GPT), Hugging Face Ecosystem

Unit 5. Transformers and Modern NLP:

Transformer architecture basics (self-attention, encoder-decoder), BERT: Pretraining, Fine-tuning,

GPT and Generative NLP, Hugging Face Ecosystem (using pre-trained models)

Text Summarization: Extractive, Abstractive, Hybrid Approaches

Applications: Document Classification, Chatbots, Virtual Assistants

Textbook:

1. Natural Language Processing, Sini Raj Pulari, Umadevi Maramreddy, Shriram k. Vasudevan, Oxford University Press
2. Speech and Language Processing, An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Daniel Jurafsky, James H. Martin, Pearson Education, 2023.

Reference Book:

1. Natural Language Processing and Information Retrieval, Tanveer Siddiqui, U.S. Tiwary , Oxford University Press.
2. 2. Natural Language Processing Recipes - Unlocking Text Data with Machine Learning and Deep Learning using Python, Akshay Kulkarni, Adarsha Shivananda, Apress, 2019.

Activities:

Outcome: Explain NLP fundamentals and basic text processing tools.
Activity: Quiz/ Assignment on Analyze ambiguities in Indian language sentences.
Evaluation Method: Quiz Score

Outcome: Apply preprocessing and parsing techniques to analyze language.
Activity: Case study: Building a simple grammar-based sentence parser.
Evaluation Method: Application of text preprocessing to raw text corpus, parsing

Outcome: Implement information extraction and text representation methods.
Activity: Hands-on NER using spaCy and NLTK.
Evaluation Method: Lab report submission on embeddings & NER.

Outcome: Build and evaluate deep learning models for NLP.
Activity: Group Discussion on Discussion: Compare RNN vs Transformers.
Evaluation Method: Depth of Understanding, Participation, Explanation

Outcome: Utilize pre-trained transformer models for advanced NLP applications.
Activity: Lab: Implement chatbot using GPT model.
Evaluation Method: Practical exam using Hugging Face models – Accuracy, Effectiveness

SEMESTER-V

COURSE 12 B: NATURAL LANGUAGE PROCESSING

Practical **Credits: 1** **2 hrs/week**

1. Install Python, NLTK, and spaCy. Write a sample program to print available NLP corpora and models.
2. Write regex patterns for extracting emails, phone numbers, hashtags, and dates from a text file.
3. Demonstrate lexical and structural ambiguity with example sentences. Use NLTK parse trees to visualize.
4. Implement sentence and word tokenization using NLTK and spaCy. Compare outputs.
5. Write a program to remove stopwords and analyze text length reduction.
6. Apply Stemming and Lemmatization on a dataset and compare differences.
7. Use NLTK to demonstrate top-down and bottom-up parsing of a simple grammar.
8. Use spaCy to extract entities (e.g., names, locations, organizations) from news text.
9. Implement text representation and calculate similarity between documents.(Bag of Words and N-grams)
10. Build a sentiment classifier using Scikit-learn (Naive Bayes / Logistic Regression).
11. Implement a simple RNN to generate sentences character by character.
12. Hugging Face to load a pretrained BERT model and perform masked word prediction.
13. Implement extractive and abstractive summarization using Hugging Face pipelines.
14. Build a simple FAQ-based chatbot using Transformer-based embeddings.

SEMESTER-V

COURSE 13 A: CLOUD COMPUTING FOR DATA SCIENCE

Theory **Credits: 3** **3 hrs/week**

Course Objectives

1. Introduce the fundamentals of cloud computing and its role in data science.
2. Provide understanding of virtualization, service, and deployment models.
3. Familiarize students with cloud storage, data management, and databases.
4. Expose students to cloud-based big data and machine learning platforms.
5. Train students in building, deploying, and monitoring ML pipelines on the cloud.

Course Outcomes

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

1. Explain cloud computing concepts including service models, deployment models, and virtualization.
2. Demonstrate cloud storage and database services for managing large-scale data.
3. Apply cloud-based platforms to run data science and machine learning workflows.
4. Build and deploy ML models on cloud services using AutoML and managed ML platforms.
5. Evaluate and monitor cloud-deployed solutions with respect to scalability, performance, and cost.

Unit 1. Introduction to Cloud Computing

Definition & Evolution of Cloud Computing, Service-Oriented Architecture (SOA) & Web Services, Utility & Grid Computing concepts, Characteristics of Cloud Computing

Cloud Computing Architecture: Front-end, Back-end, Networking, Delivery Models

Cloud Service Models: SaaS, PaaS, IaaS, Continuous Delivery using PaaS

Unit 2. Virtualization & Deployment Models

Concept & importance of Virtualization, Types of Virtualizations: Application, Network, Desktop, Storage, Server, Data Virtualization

Cloud Deployment Models: Public, Private, Community, Hybrid

Role of Cloud Computing in Data Science, Advantages of Cloud in Machine Learning

Unit 3. Cloud Storage & Data Management

Cloud Storage: Introduction, Benefits, Use Cases (Backup, Archiving, DR, Content Delivery)

Cloud Storage Systems: Block-based, File-based, Object-based storages

Key-Value Databases: Features & limitations

Batch vs. Streaming data for ML pipelines

Cloud Data Warehouses: AWS Redshift, Google BigQuery

Unit 4. Cloud Platforms for Data Science & ML

Machine Learning in the Cloud: Benefits & Limitations, Cloud-based ML Services: AIaaS, GPUaaS

Managed ML Platforms: Overview & advantages, Cloud ML Platforms: AWS SageMaker, Azure ML Studio, Google Cloud AutoML

Unit 5. Training & Deployment of ML on Cloud

Factors for selecting Cloud ML Platforms: ETL/ELT pipeline support, Scale-up/out training, ML frameworks, Pre-tuned services

Steps for Training ML Models in Cloud: Data source identification, Feature engineering, Training, Validation, Deployment, Monitoring, Monitoring & improving cloud-deployed ML models

Case studies & industry applications

Text / Reference Books

1. Handbook of Cloud Computing, Dr. Anand Nayyar, BPB Publications (2019)
2. Cloud Computing: A Practical Approach, Toby Velte, Anthony Velte, Robert C., McGraw Hill
3. Cloud Computing for Data Analysis, Noah Gift, Alfredo Deza, Pragmatic AI Labs
4. Data Science on the Google Cloud Platform, Valliappa Lakshmanan, O'Reilly
5. Machine Learning in the AWS Cloud: Amazon SageMaker, Abhishek Mishra, Wiley

Activities:

Outcome: Explain cloud computing concepts including service models, deployment models, and virtualization.

Activity: Prepare a comparative chart/report on cloud service and deployment models with real-world examples (AWS, Azure, GCP).

Evaluation Method: Report submission & viva (assess clarity, accuracy, and examples used).

Outcome: Demonstrate cloud storage and database services for managing large-scale data.
Activity: Perform lab experiments on block/file/object storage and execute queries in BigQuery / RDS.
Evaluation Method: Lab performance & practical exam (students demonstrate CRUD operations and explain storage use cases).

Outcome: Apply cloud-based platforms to run data science and machine learning workflows.
Activity: Implement a cloud-based ETL pipeline (e.g., using AWS Glue / Dataflow) for preparing a dataset.
Evaluation Method: Lab report + demo evaluation (workflow completeness, correctness of execution).

Outcome: Build and deploy ML models on cloud services using AutoML and managed ML platforms.
Activity: Train and deploy a classification/regression model using AWS SageMaker / Azure ML / Google AutoML.
Evaluation Method: Practical demo + oral viva (assess deployment success, prediction results, and understanding of pipeline).

Outcome: Evaluate and monitor cloud-deployed solutions with respect to scalability, performance, and cost.
Activity: Configure monitoring (e.g., CloudWatch, Stackdriver) for a deployed ML service and analyze resource usage.
Evaluation Method: Mini-project report & presentation (grading based on monitoring setup, analysis quality, cost optimization suggestions).

O. Jal

SEMESTER-V

COURSE 13 A: CLOUD COMPUTING FOR DATA SCIENCE

Practical	Credits: 1	2 hrs/week

1. Create Virtual Machine using VMware workstation for Windows/Linux
2. Install & configure WAMP/XAMPP/Apache on the VM and host a sample page
3. Install and configure a cloud account (AWS/Azure/GCP free tier).
4. Create and manage storage buckets; upload and access datasets.
5. Launch an instance and configure Block-based storage (EBS)
6. Create & configure File-based storage on cloud VM (EFS/Network FS).
7. Set up Jupyter Notebook/Colab on cloud VM.
8. Connect to cloud-hosted database services (AWS RDS, BigQuery, Cosmos DB).
9. Implement a batch ETL pipeline in the cloud.
10. Launch a SageMaker notebook, attach IAM role and S3 bucket, run sample notebook
11. Build a classification/regression model using AWS SageMaker / Azure ML Studio / GCP AI Platform.
12. Implement a simple ETL job: extract (RDS / CSV), transform, load into cloud DW (e.g., Redshift / BigQuery).
13. Use CloudWatch / Stackdriver to monitor endpoints, set alarms and auto-scale rules.
14. Use cloud AutoML services for dataset prediction tasks.
15. Deploy a trained ML model as a REST API endpoint in the cloud.

SEMESTER-VI

COURSE 13 B: CONVERSATIONAL AI

Theory	Credits: 3	3 hrs/week

Course Objectives

1. Understand the fundamentals and significance of conversational AI and chatbots.
2. Learn to design conversation flows and manage dialogs effectively.
3. Gain hands-on experience in building and training Natural Language Understanding (NLU) models.
4. Develop skills for creating multilingual and voice-enabled chatbots.
5. Testing, deployment, and maintenance of conversational AI systems.

Course Outcomes

Upon completion, students will be able to:

1. Describe core components and architecture of conversational AI systems.
2. Design and implement dialog flows and effective conversation management strategies.
3. Build and evaluate NLU models for intent classification and entity extraction.
4. Create multilingual and voice-enabled chatbots integrating speech technologies.
5. Conduct chatbot testing, deploy on multiple platforms, and implement monitoring for continuous improvement.

Unit 1: Introduction to Conversational AI and Chatbots

What is Conversational AI and its importance, Overview of chatbot use cases and domains, Key components: NLU, dialog management, NLG, Dialog systems and architecture, Introduction to chatbot platforms and tools, Differences between rule-based and AI chatbots, Challenges in conversational AI development

Unit 2: Designing Conversations and Dialog Management

Understanding conversation flows and user intents, Dialog management strategies: state machines, frame-based systems, Slot filling techniques for context maintenance, Handling multi-turn conversations and context switching, Managing conversation errors and fallback strategies, Role of session and user context in dialog personalization, Best practices in conversation design for UX

Unit 3: Natural Language Understanding (NLU) Fundamentals

Preparing training annotation and labeling, intent classification techniques and algorithms, Entity recognition and extraction methods, Using pre-trained models and transfer learning in NLU, Handling ambiguous and overlapping intents, Supporting multiple languages with NLU components, Evaluating NLU model performance

Unit 4: Building Multilingual and Voice-Enabled Chatbots

Techniques to support multilingual conversations, Speech recognition and synthesis basics, Integrating voice assistants with chatbots, text-to-speech (TTS) and speech-to-text (STT), Designing voice user interfaces (VUIs), Testing voice chatbots with different accents and dialects, Privacy and ethical considerations in voice-enabled AI

Unit 5: Testing, Deployment, and Maintenance of Chatbots

Testing chatbots: functional, user acceptance, load testing, Metrics for chatbot performance and user satisfaction, Continuous learning, retraining, and updating models, Handling failures and graceful degradation, Deploying chatbots across platforms and integration with backend systems, Analytics and monitoring tools for conversational AI, Future trends in conversational AI and emerging technologies

Recommended Textbooks & References

- "Reinforcement Learning: An Introduction" by Richard S. Sutton and Andrew G. Barto, MIT Press
- "Conversational AI: Chatbots that work" by Andrew Freed, Packt Publishing
- "Handbook of Natural Language Processing" edited by Nitin Indurkhya and Fred J. Damerau
- "Conversational Interfaces: Principles and Practice" by Michael McTear
- "Deep Learning for Natural Language Processing" by Palash Goyal, Sumit Pandey, Karan Jain
- "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig
- Rasa Open Source Documentation (https://rasa.com/docs/rasa)
- "Building Chatbots with Python" by Sumit Raj, Apress
- Online tutorials and practical guides from Rasa community and GitHub repositories

Activities

Unit 1: Introduction to Conversational AI and Chatbots

Activity: Explore chatbot examples and dissect major components

Outcome: Understand conversational AI scope, architecture, and difference between rule-based and AI chatbots

Evaluation Method: Quiz on chatbot concepts and architecture; assignment on case studies of chatbot use cases

Unit 2: Designing Conversations and Dialog Management

Activity: Map conversation flows, build intents and slots, design fallback strategies using Rasa

Outcome: Ability to model user intents, manage context, and design resilient dialogs

Evaluation Method: Lab exercises designing conversation flows; peer review of chatbot dialogues

Unit 3: Natural Language Understanding (NLU) Fundamentals

Activity: Annotate training data, train NLU pipeline in Rasa, experiment with entity extraction and intent classification

Outcome: Develop skills in preparing data and training NLU models; evaluate model performance

Evaluation Method: Practical NLU model accuracy tests; report on training data preparation and challenges

Unit 4: Building Multilingual and Voice-Enabled Chatbots

Activity: Implement multilingual support using separate pipelines; integrate speech-to-text and text-to-speech APIs

Outcome: Create chatbots supporting multiple languages and voice interfaces

Evaluation Method: Demonstration of multilingual chatbot proficiency; voice bot interaction testing

Unit 5: Testing, Deployment, and Maintenance of Chatbots

Activity: Use Rasa X for interactive testing, deploy chatbot on cloud servers, monitor conversation analytics

Outcome: Master chatbot evaluation, deployment techniques, error handling, and continuous improvement

Evaluation Method: Final project deployment; analysis report of chatbot performance and user feedback

SEMESTER-V

COURSE 13 B: CONVERSATIONAL AI

Practical | **Credits: 1** | **2 hrs/week**

Tools: Rasa Open Source

Experiment 1: Setting Up Rasa

- Install Rasa and prerequisite tools
- Understand folder structure and key files (domain.yml, nlu.yml, stories.yml)
- Create a basic rule-based chatbot and test on Rasa shell

Experiment 2: Defining Intents and Entities

- Create intents and entities in `nlu.yml`
- Annotate training data for intent classification and entity extraction
- Train and test the NLU model with sample queries

Experiment 3: Dialogue Management with Stories and Rules

- Write stories for conversation flows (user intents + bot responses)
- Define rules for slot filling and fallback handling
- Test multi-turn conversations and context handling

Experiment 4: Implementing Slots and Form Actions

- Use slots to capture and remember user inputs
- Create forms to fill multiple slots during conversations
- Manage slot validity and resetting

Experiment 5: Custom Actions for Dynamic Responses

- Write custom Python actions (in `actions.py`) for dynamic content
- Use APIs or simple logic to enhance chatbot responses
- Test bot with custom actions integrated

Experiment 6: Multilingual Chatbot Development

- Extend chatbot to support multiple languages via separate NLU pipelines
- Handle intents and responses per language
- Test language switching in conversations

Experiment 7: Integrating Voice with Rasa

- Connect Rasa chatbot with speech-to-text and text-to-speech services
- Build a simple voice-enabled chatbot demo
- Evaluate voice input accuracy and bot response

Experiment 8: Error Handling, Fallbacks, and Recovery

- Implement fallback policies for unclear user inputs
- Log fallback incidents for model improvement
- Design conversation recovery strategies

Experiment 9: Testing, Evaluation, and Debugging

- Use Rasa X for interactive learning and conversation reviews
- Evaluate intent classification, entity extraction, and dialogue success
- Debug and retrain models iteratively

Experiment 10: Deploying Rasa Chatbot

- Deploy Rasa chatbot locally and on cloud platforms (Heroku, Railway)
- Expose bot via chatbot UI or API endpoints
- Test chatbot in real-world web interface

SEMESTER-VI

COURSE 14 A: NEURAL NETWORKS AND DEEP LEARNING

Theory **Credits: 3** **3 hrs/week**

Course Objectives:

1. Introduce the fundamental concepts of Artificial Neural Networks and Deep Learning, along with their historical and biological inspirations.
2. Provide an in-depth understanding of different neural network architectures including Perceptron, DNN, CNN, RNN, and advanced models.
3. Develop hands-on skills to design, train, and evaluate deep learning models using popular frameworks such as TensorFlow and Keras.
4. Expose students to applications of deep learning in computer vision, natural language processing, and generative modeling.
5. Enable students to critically analyze challenges in deep learning such as overfitting, bias, and ethical concerns.

Course Outcomes:

After successful completion of this course, students will be able to:

1. Explain the principles of neural networks, perceptrons, activation functions, and the evolution of deep learning.
2. Apply concepts of forward/backward propagation, weight initialization, and optimization techniques to train deep neural networks.
3. Design and implement convolutional neural networks (LeNet, AlexNet, VGG) for image classification tasks.
4. Build and analyze recurrent neural networks (RNN, LSTM, GRU) for sequential data and natural language processing applications.
5. Experiment with advanced deep learning concepts such as transfer learning, generative models, and transformers using pre-trained models.

Unit 1. Foundations of Deep Learning:

What is Artificial Intelligence, Machine Learning, and Deep Learning? History and applications of deep learning, Biological vs. Artificial Neurons

Introduction to Neural Networks, Perceptron and activation functions (Linear, ReLU, Sigmoid, Tanh, Softmax), Types of Neural Networks (shallow vs. deep, feedforward vs. recurrent), Gradient descent and backpropagation (conceptual only), Concept of loss functions (MSE, cross-entropy) at intuitive level

Unit 2. Deep Neural Networks:

Forward and backward propagation, Weight initialization, learning rate, and optimization algorithms (SGD, Adam, RMSProp), Overfitting & underfitting: Regularization, Dropout, Batch normalization, Activation functions in deep networks, Loss functions in detail (MSE, cross-entropy, hinge loss) Introduction to Keras/TensorFlow framework

Unit 3. Convolutional Neural Networks (CNNs):

Introduction to images and pixels, Filters/kernels, padding, and pooling, CNN architecture and layers (Conv, Pooling, Fully Connected, Softmax), Classical CNN architectures: LeNet-5 (digit recognition - first CNN model), AlexNet (ImageNet breakthrough - deeper CNN), VGG (concept of depth, simplicity)

Applications in image classification, object detection, facial recognition

Unit 4. Recurrent Neural Networks (RNNs) and NLP

Sequences and time series data, Introduction to RNNs: vanishing/exploding gradient issue LSTM and GRU (intuitive and architectural view), Word embeddings: Word2Vec, GloVe, introduction to contextual embeddings (BERT at high level)

Applications: Sentiment analysis, text generation, simple time-series forecasting

Unit 5. Advanced & Emerging Topics:

Generative models: GANs (Generator & Discriminator intuition), VAEs (introduction only), Transformers: attention mechanism (intuitive), BERT, GPT family (overview), Transfer learning & fine-tuning pre-trained models (vision & NLP), AI ethics: Bias, fairness, privacy, safety, explainability

Textbooks:

1. Chollet, F. (2018). *Deep Learning with Python* (1st ed.). Manning Publications.
2. Nielsen, M. A. (2015). *Neural Networks and Deep Learning*. Determination Press. *(Available free online: http://neuralnetworksanddeeplearning.com)*

Reference Books:

1. Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.

2. Howard, J., & Gugger, S. (2020). *Practical Deep Learning for Coders*. O'Reilly Media. *(Based on the fast.ai course)*
3. Shane, J. (2019). *You Look Like a Thing and I Love You: How AI Works and Why It's Making the World a Weirder Place.* Voracious / Hachette Book Group.

Activities:

Outcome: Explain the principles of neural networks, perceptrons, activation functions, and the evolution of deep learning.

Activity: *Concept Mapping Exercise* - Students work in small groups to draw a timeline-based concept map linking biological neurons → perceptron → multilayer perceptron → activation functions → deep learning.

Evaluation Method: Rubric-based evaluation of concept maps (clarity, correctness, and depth of connections).

Outcome: Apply concepts of forward/backward propagation, weight initialization, and optimization techniques to train deep neural networks.

Activity: *Hands-on Coding Task* - Students implement a simple 3-layer neural network from scratch in Python/NumPy (without Keras/TensorFlow) to observe propagation and optimization effects.

Evaluation Method: Code demonstration + oral viva where students explain how changing initialization/learning rate affects training.

Outcome: Design and implement convolutional neural networks (LeNet, AlexNet, VGG) for image classification tasks.

Activity: *Mini Project (Image Classification)* - Students form teams to train CNNs (LeNet, AlexNet, VGG) on a dataset like MNIST/CIFAR-10 and compare results.

Evaluation Method: Project report + presentation with accuracy comparison, confusion matrix, and discussion of design choices.

Outcome: Build and analyze recurrent neural networks (RNN, LSTM, GRU) for sequential data and natural language processing applications.

Activity: *Case Study on Sentiment Analysis* - Students use IMDB reviews dataset to build RNN/LSTM/GRU models for sentiment classification and analyze performance.

Evaluation Method: Submission of case study report with experimental setup, evaluation metrics (accuracy/F1-score), and reflection on differences among models.

Outcome: Experiment with advanced deep learning concepts such as transfer learning, generative models, and transformers using pre-trained models.

Activity: *Model Exploration & Demonstration* - Students choose one advanced technique (Transfer Learning on ResNet, GAN for image generation, or Transformer for text classification) and prepare a live demo in class.

Evaluation Method: Evaluation of demo + short reflective note (1–2 pages) on challenges, benefits, and application potential of the chosen technique.

SEMESTER-VI

COURSE 14 A: NEURAL NETWORKS AND DEEP LEARNING

Practical | **Credits: 1** | **2 hrs/week**

1. Build a perceptron from scratch in Python
2. Use Google Teachable Machine or Tensor Flow Playground
3. Visualize various Activation Functions and their Gradients
4. Build and train a deep neural network for classification (e.g., MNIST digits)
5. Experiment with dropout, batch normalization, and different activations
6. Train a CNN to classify fashion images (Fashion-MNIST)
7. Visualize filters and feature maps
8. Fine-tune a pre-trained CNN (Mobile Net, VGG) for a small dataset
9. Build an LSTM model for movie review sentiment analysis (IMDb dataset)
10. Generate text using a simple character-level RNN
11. Use a pre-trained model (like MobileNet or BERT) for a simple task
12. Use Huggingface to deploy a Sentiment Analysis App for Swiggy Reviews

SEMESTER-VI

COURSE 14 B: TIME SERIES ANALYSIS AND FORECASTING

Theory **Credits: 3** **3 hrs/week**

Course Objectives

The course aims to:

1. Provide fundamental understanding of time series data, components, and characteristics.
2. Train students in identifying, modeling, and forecasting using ARMA/ARIMA/SARIMA models.
3. Introduce state-space and multivariate approaches for complex data.
4. Familiarize students with modern forecasting methods, including spectral and evaluation techniques.
5. Enable hands-on practice with real-world datasets using R/Python statistical libraries.

Course Outcomes

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

1. Explain the concepts of time series, stationarity, and autocorrelation functions.
2. Apply ARMA/ARIMA/SARIMA models to real-world time series data.
3. Analyze multivariate and state-space time series using appropriate tools.
4. Implement forecasting workflows using R/Python for financial, business, and scientific datasets.
5. Evaluate forecast accuracy and select appropriate models using statistical criteria.

Unit 1. Fundamentals & Stationary Processes

Introduction to time series: types, components, forecasting process. Stationary processes: definitions, autocovariance, autocorrelation functions (ACF/PACF). Model evaluation metrics. ACF/PACF example analyses.

Unit 2. ARMA & Forecasting with ARMA

ARMA(p,q) models: definition, estimation, forecasting approaches. Model identification: AIC, PACF/ACF, diagnostic checks. Practical examples of fitting ARMA and generating forecasts.

Unit 3. Non-Stationary & Seasonal Models

Non-stationary time series: differencing, unit roots. Seasonal models: SARIMA and multiplicative seasonal ARIMA. Identification, estimation, and diagnostic checks for seasonal models.

Unit 4. State-Space & Multivariate Time Series

Multivariate time series: Vector ARMA models (VARMA), estimation, forecasting. State-space representation: formulation, Kalman filter basics, forecasting in state-space models.

Unit 5. Advanced Topics & Forecast Evaluation

Spectral analysis: frequency-domain representation, spectral density. Forecast performance: measures, monitoring, choosing models.

Textbook:

1. Introduction to Time Series and Forecasting, Peter J. Brockwell & Richard A. Davis, 2nd Edition, Springer

Reference Books

1. Time Series Analysis: Forecasting and Control, George E. P. Box, Gwilym M. Jenkins & Gregory C. Reinsel
2. Introduction to Time Series Analysis and Forecasting, Douglas C. Montgomery, Cheryl L. Jennings, Murat Kulahci , (Wiley)
3. Time Series Analysis and Its Applications: With R Examples, R. H. Shumway & D. S. Stoffer

Activities:

Outcome: Explain the concepts of time series, stationarity, and autocorrelation functions

Activity: Students will prepare a seminar or short presentation explaining stationarity, ACF, PACF with a simple dataset example (like sales data).

Evaluation Method: Evaluated through presentation quality, understanding during viva, and a short concept quiz.

Outcome: Apply ARMA/ARIMA/SARIMA models to real-world time series data

Activity: Students will conduct a hands-on lab task to fit ARIMA and SARIMA models on stock price or rainfall data using Python/R.

Evaluation Method: Lab record submission, correctness of implementation, and a practical exam.

Outcome: Analyze multivariate and state-space time series using appropriate tools
Activity: Students will carry out a case study on macroeconomic datasets (like GDP, inflation, unemployment) using VAR or state-space modeling.
Evaluation Method: Case study report, results interpretation, and oral viva.

Outcome: Implement forecasting workflows using R/Python for financial, business, and scientific datasets
Activity: Students will design an end-to-end forecasting pipeline (data preprocessing → model building → forecasting → visualization). Example: forecasting COVID-19 daily cases or retail sales.
Evaluation Method: Project demo, code submission, and project report.

Outcome: Evaluate forecast accuracy and select appropriate models using statistical criteria
Activity: Students will compare multiple forecasting methods (e.g., ARIMA vs. Exponential Smoothing) on the same dataset and analyze performance using RMSE, MAE, and MAPE.
Evaluation Method: Written assignment, interpretation of metrics, and justification of chosen model.

SEMESTER-VI

COURSE 14 B: TIME SERIES ANALYSIS AND FORECASTING

Practical **Credits: 1** **2 hrs/week**

(Using R/Python statsmodels, pandas, forecast, or equivalent)

1. Import and visualize time series datasets (stock, weather, sales).
2. Perform decomposition of time series into trend, seasonal, residual components.
3. Compute and plot Autocorrelation Function (ACF) & Partial ACF (PACF).
4. Test stationarity using Augmented Dickey-Fuller (ADF) test.
5. Fit ARMA models and validate residuals.
6. Implement ARIMA and SARIMA models for seasonal data.
7. Perform model selection using AIC/BIC and cross-validation.
8. Forecast with ARIMA/SARIMA and plot prediction intervals.
9. Apply multivariate time series (VAR) to macroeconomic datasets.
10. Explore state-space models using Kalman filtering..
11. Conduct spectral analysis of a time series.
12. Compare forecasting methods: ARIMA vs. Exponential Smoothing vs. ML models.
13. Evaluate forecast performance with RMSE, MAPE, etc.

SEMESTER-VI

COURSE 15 A: ROBOTICS KINEMATICS & DYNAMICS

Theory | **Credits: 3** | **3 hrs/week**

Course Objectives

1. Introduce fundamental concepts of robot kinematics and dynamics.
2. Develop ability to analyze robot motion through forward and inverse kinematics.
3. Understand velocity and acceleration relations via Jacobian matrices.
4. Derive robot dynamics equations using Newton-Euler and Lagrangian methods.
5. Learn trajectory planning and control techniques for robotic manipulators.

Course Outcomes

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

1. Identify robot components, degrees of freedom, and use Denavit-Hartenberg convention for kinematic modeling.
2. Solve forward and inverse kinematics problems for serial manipulators.
3. Compute Jacobians and analyze velocity kinematics and singularities.
4. Apply Newton-Euler and Lagrangian formulations to derive dynamic equations of robot manipulators.
5. Develop trajectory plans and implement basic control strategies for robot motion.

Unit 1: Introduction to Robot Kinematics

Introduction to robotics: types and applications, Robot components and coordinate frames, Degrees of freedom and workspace, Homogeneous transformations and spatial transformations, Denavit-Hartenberg (D-H) convention for link representation

Unit 2: Forward and Inverse Kinematics

Forward kinematics: computing end-effector position and orientation

Inverse kinematics: analytical and numerical solutions

Workspace analysis and tool configurations, Redundancy and multiple solutions in inverse kinematics

Unit 3: Velocity and Jacobian Analysis

Linear and angular velocities of rigid bodies, Differential motions and velocity propagation through robot links, Jacobian matrix: formulation and interpretation, Singularity analysis and dexterity measures, Kinematic duality and manipulator velocity control

Unit 4: Robot Dynamics

Basics of kinetics: work-energy, momentum principles, Newton-Euler formulation of robot dynamics, Lagrangian mechanics and derivation of equations of motion, Dynamic modeling of serial manipulators, Inclusion of actuator dynamics and nonrigid effects

Unit 5: Trajectory Planning and Control

Trajectory planning in joint and Cartesian spaces, Path interpolation and smooth motion generation, Control strategies: PD control, computed torque, inverse dynamics control, Force and hybrid control, Introduction to robot programming languages and simulation

Suggested Textbooks & References

1. John J. Craig, Introduction to Robotics: Mechanics and Control, Pearson
2. Mark W. Spong, Seth Hutchinson, M. Vidyasagar, Robot Modeling and Control, Wiley
3. Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, Giuseppe Oriolo, Robotics: Modelling, Planning and Control, Springer
4. Robert J. Schilling, Fundamentals of Robotics: Analysis and Control, Prentice Hall
5. Haruhisa Kawasaki, Robot Kinematics and Dynamics, Encyclopedia of Life Support Systems
6. MATLAB Robotics Toolbox Documentation and Tutorials for practical work

Activities:

Unit 1: Introduction to Robot Kinematics

Activities:

- Study robot types and components
- Setup coordinate frames and assign Degrees of Freedom
- Practice homogeneous and spatial transformations
- Apply Denavit-Hartenberg parameters modeling on simple mechanisms

Outcomes: Ability to model robots in terms of coordinate frames and D-H parameters.

Evaluation: Quiz and assignment on D-H parameter assignments and transformations.

Unit 2: Forward and Inverse Kinematics

Activities:

- Compute end-effector position and orientation forward kinematics
- Solve inverse kinematics analytically and numerically for basic robots
- Analyze robot workspace and tool configurations

Outcomes: Skill in solving forward/inverse kinematic problems and workspace analysis.

Evaluation: Problem-solving assignment and class test.

Unit 3: Velocity and Jacobian Analysis

Activities:

- Calculate linear and angular velocities of robot links
- Derive Jacobian matrices and interpret their significance
- Study singularity and robot dexterity measures

Outcomes: Understanding of velocity propagation and kinematic influence on manipulator control.

Evaluation: Lab exercises on Jacobian calculation and singularity analysis.

Unit 4: Robot Dynamics

Activities:

- Learn basic kinetics principles relevant to robotics
- Derive equations of motion with Newton-Euler and Lagrangian methods
- Model dynamics of serial manipulators including actuator effects

Outcomes: Ability to formulate dynamic models for robot motion analysis.

Evaluation: Assignment on dynamics derivation and mid-term written exam.

Unit 5: Trajectory Planning and Control

Activities:

- Generate and simulate robot trajectories in joint and Cartesian spaces
- Explore control schemes: PD, computed torque, inverse dynamics control
- Understand force and hybrid control approaches; introduction to programming and simulation tools

Outcomes: Skills in planning trajectories and designing control strategies for robot manipulators.

Evaluation: Project work and final exam practicals.

SEMESTER-VI

COURSE 15 A: ROBOTICS KINEMATICS & DYNAMICS

Practical	Credits: 1	2 hrs/week

Practical 1: MATLAB Basics for Robotics

- Learn basic MATLAB operations and plotting
- Visualize points, vectors, and coordinate frames in 2D and 3D

Practical 2: Coordinate Transformations

- Create rotation and translation matrices
- Build homogeneous transformation matrices
- Apply transformations to points and visualize the results

Practical 3: Denavit-Hartenberg Parameters and Forward Kinematics

- Assign DH parameters for a simple 2-DOF planar arm
- Compute forward kinematics using DH transformation
- Plot end-effector positions for given joint angles

Practical 4: Inverse Kinematics of Planar Manipulator

- Solve analytical inverse kinematics for 2-DOF arm
- Verify solutions by forward kinematics plotting

Practical 5: Jacobian and Velocity Kinematics

- Calculate Jacobian matrix for 2-DOF arm
- Compute end-effector velocity from joint velocities
- Analyze singular configurations

Practical 6: Basic Trajectory Planning

- Generate joint trajectories with linear interpolation
- Plot joint angle, velocity, acceleration profiles
- Simulate end-effector path

Practical 7: Simple Dynamic Calculations

- Calculate torque requirements for joint movements
- Plot torque vs. time for simple motion profiles

Practical 8: Robot Motion Animation

- Animate arm movement following generated trajectories
- Visualize workspace and reachable points

SEMESTER-VI

COURSE 15 B: DATA ENGINEERING & MLOPS

Theory **Credits: 3** **3 hrs/week**

Course Objectives

1. To introduce the lifecycle and roles in Data Engineering.
2. To explore data architecture principles, distributed systems, and technology choices.
3. To analyze MLOps features, risks, and challenges in developing ML systems.
4. To design CI/CD pipelines and deployment strategies for ML models.
5. To understand monitoring, governance, and Responsible AI compliance in production ML.

Course Outcomes

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

1. Explain Data Engineering and its organizational roles.
2. Analyze major concepts in data architecture and distributed systems.
3. Apply MLOps features and evaluate challenges in ML model development.
4. Design and implement CI/CD pipelines for ML deployment.
5. Evaluate governance and Responsible AI practices in MLOps.

Unit 1. Foundations of Data Engineering

Data Engineering: definition, lifecycle, skills, activities. Evolution and roles of Data Engineers: technical vs business responsibilities, internal vs external roles. Relationship between Data Engineering and Data Science. Data lifecycle vs Data Engineering lifecycle.

Unit 2. Data Architecture & Distributed Systems

Enterprise and Data Architecture definitions. Principles of good data architecture. Scalability, failure design, tiers, microservices, monolith vs modular. Event-driven architecture, hybrid cloud, multicloud, edge computing. Technology selection criteria: team size, interoperability, cost, TCO.

Unit 3. MLOps Fundamentals

MLOps challenges and risk mitigation. Responsible AI and scaling ML solutions. Key MLOps features: EDA, feature engineering, model training & evaluation, reproducibility. Deployment requirements, monitoring basics. Model versioning and experimentation tracking.

Unit 4. Model Deployment & CI/CD Pipelines

Preparing models for production. Runtime environments: dev to production adaptation.
CI/CD pipelines: building ML artifacts, testing pipelines. Deployment strategies: batch, online, A/B testing, canary releases. Containerization & scaling (Docker, Kubernetes).

Unit 5. Monitoring, Feedback Loops & Governance

Monitoring models in production: drift detection, ground truth evaluation.
Feedback loops: retraining workflows, online evaluation. Logging, monitoring frameworks.
Governance: regulations (GDPR, CCPA, GxP), Responsible AI principles.
Templates for governance, compliance, and model risk management.

Text/ Reference books

1. Fundamentals of Data Engineering, Joe Reis & Matt Housley, O'Reilly, 2022.
2. Introducing MLOps, Mark Treveil & Dataiku Team, O'Reilly, 2020

Web Resources:

https://www.ibm.com/think/topics/data-engineering

https://martinfowler.com/articles/microservices.html

https://towardsdatascience.com/a-gentle-introduction-to-mlops-7d64a3e890ff/

Activities

Outcome: Explain Data Engineering and roles

Activity: Prepare a concept map showing different Data Engineer roles in an organization.

Evaluation Method: Short presentation + written quiz.

Outcome: Analyze data architecture concepts

Activity: Case study on choosing between monolith, microservices, and event-driven architectures.

Evaluation Method: Case study report + viva.

Outcome: Apply MLOps features in ML development
Activity: Lab exercise on feature engineering & reproducibility using MLflow.
Evaluation Method: Lab record submission + demo.

Outcome: Design CI/CD pipelines for ML
Activity: Mini-project: build a simple ML CI/CD pipeline with GitHub Actions/Docker.
Evaluation Method: Project demo + evaluation rubric.

Outcome: Evaluate governance and Responsible AI practices
Activity: Group discussion & policy brief on GDPR/Responsible AI practices.
Evaluation Method: Written assignment + peer review.

SEMESTER-VI

COURSE 15 B: DATA ENGINEERING & MLOPS

Practical | **Credits: 1** | **2 hrs/week**

1. Install and configure a modern data engineering environment (Python, Jupyter, VSCode).
2. Explore and visualize the data lifecycle on a sample dataset (sales/weather).
3. ETL basics: extract data from CSV/JSON -> transform -> load into relational database.
4. Compare performance of batch vs event-driven ingestion using Apache Kafka or RabbitMQ.
5. Deploy a small dataset on Hadoop Distributed File System (HDFS) and perform simple operations.
6. Case study lab: design a microservices vs monolithic workflow for a mock business problem.
7. Perform Exploratory Data Analysis (EDA) and track experiments using **MLflow.**
8. Build a simple ML model (regression/classification) and enable **reproducibility** with version control.
9. Manage datasets and model versions using **DVC (Data Version Control).**
10. Containerize an ML model with **Docker.**
11. Automate training and deployment with a **GitHub Actions CI/CD pipeline.**
12. Deploy an ML model as a REST API using **FastAPI / Flask.**
13. Implement model drift detection: monitor incoming data and compare with training data distribution.
14. Build a simple feedback loop: retrain a model automatically when drift exceeds a threshold.
15. Configure logging and monitoring using **Prometheus/Grafana** for a deployed ML model.
16. Case study: analyze GDPR/Responsible AI implications on a real dataset (e.g., facial recognition).

Activity: Mini-project implementing MapReduce job and Hive queries
Evaluation Method: Project report + execution demo

Outcome: Design and implement data ingestion workflows
Activity: Lab task to ingest data using Sqoop/Flume and serialize with Avro/Parquet
Evaluation Method: Lab record + output validation

Outcome: Utilize NoSQL and ecosystem enhancements
Activity: Case study on HBase–Spark integration with example dataset
Evaluation Method: Case study report + written test