NIRMA UNIVERSITY

Institute:	School of Technology
Name of Programme:	M Tech CSE, MTech CSE (Cyber Security), MTech CSE (Data Science)
Course Code:	6CS377ME25
Course Title:	Distributed Data Analytics
Course Type:	Department Elective-II
Year of Introduction:	2025-26

L	T	Practical Component				C
		LPW	PW	W	S	
2	0	2	-	-	-	3

Course Learning Outcomes (CLO):

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

1. demonstrate the implementation of scalable solutions for mining massive datasets using Spark's MLlib and other core libraries (BL2)
2. analyse the architectural components and distributed computing paradigms of Apache Spark for efficient big data processing (BL3)
3. experiment with querying, graphs, and streaming data in Spark (BL3)
4. evaluate the performance of Spark applications. (BL5)

Unit	Contents	Teaching Hours (Total 30)
Unit-I	**Analyzing Big Data:** Challenges of Data Science, Evolution of Big Data Processing (Hadoop vs. Spark), Introduction MapReduce Programming, MapReduce Based Machine Learning, Introduction of Apache Spark, The Spark Programming Model	06
Unit-II	**Spark Framework:** Spark Ecosystem and Components, Advantages of Spark, Spark Driver, Executors, and Cluster Managers, Resilient Distributed Dataset (RDD), Execution Flow (DAGs, Stages, and Tasks), Cluster Deployment Modes, Writing Spark Application - Spark Programming in Scala, Python, R, Java - Application Execution	06
Unit-III	**Spark Streaming, Spark SQL and GraphX:** Overview – Errors and Recovery – Streaming Source – Streaming live data with spark, SQL Context – Importing and Saving data – Data frames – using SQL – GraphX overview – Creating Graph – Graph Algorithms	06
Unit-IV	**Classification, Regression, and Clustering with Spark MLLib:** Linear support vector machines - Naive Bayes model- Decision Trees - Least square regression- Decision trees for regression, Hierarchical Clustering in a Euclidean and Non-Euclidean Space – The Algorithm of Bradley, Fayyad, and Reina - A variant of K-means algorithm	06
Unit-V	**Recent Trends in Big Data Analytics using Spark:** Recommendation Systems, Predictive Modelling, Geospatial and Temporal Analysis, Financial Risk Estimation.	06

Self-Study:

The self-study contents will be declared at the commencement of the semester. Around 10% of the questions will be asked from self-study content.

Suggested Readings/ References:

1. Parsian, M., Data algorithms with Spark: Recipes and design patterns for scaling up using PySpark, O'Reilly
2. Chambers, B., & Zaharia, M., Spark: The definitive guide: Big data processing made simple, O'Reilly
3. Frampton, M., Mastering Apache Spark. Packt Publishing
4. Guller, M., Big data analytics with Spark: A practitioner's guide to using Spark for large scale data analysis. Apress
5. Ryza, S., Laserson, U., Owen, S., & Wills, J., Advanced Analytics with Spark: Patterns for learning from data at scale, O'Reilly
6. Bekkerman, R., Bilenko, M., & Langford, J., Scaling up machine learning: Parallel and distributed approaches, Cambridge University Press
7. Miner, D., & Shook, A., MapReduce design patterns: Building effective algorithms and analytics for Hadoop and other systems, O'Reilly
8. Lin, J., & Dyer, C, Data-intensive text processing with MapReduce, Morgan & Claypool Publishers.

Suggested List of Experiments:

Sr. No.	Name of Experiments/Exercises	Hours
1	**Hadoop vs. Spark Comparison** Objective: To compare Hadoop and Spark frameworks by processing a sample dataset (e.g., word count or log file analysis). Procedure: Process the dataset using Hadoop MapReduce and Spark RDD. Measure the execution time and resource usage for both frameworks. Outcome: Understand the performance differences between Hadoop and Spark	04
2	**MapReduce Programming Concepts using Spark** Objective: To write a MapReduce program for counting word frequencies in a large text dataset. Procedure: Create a Java or Python program using the MapReduce framework. Execute the program on a Hadoop cluster. Outcome: Learn the fundamentals of MapReduce programming and its execution	04
3	**RDD Operations and Transformation in Spark** Objective: To perform basic RDD operations such as map, filter, reduce, and join using Spark. Procedure: Load a sample dataset into an RDD. Apply transformations (map, filter) and actions (reduce, collect) on the RDD. Observe and interpret the results. Outcome: Learn to work with Resilient Distributed Datasets (RDDs) in Spark	06
4	**Machine Learning using Spark** Objective: To implement a Naive Bayes classifier using MapReduce for text classification.	04

	Procedure: Create a labeled dataset for classification. Write the MapReduce code to calculate probabilities and classify the data. Test the classifier's performance on the dataset. Outcome: Understand how to implement machine learning models using MapReduce	
5	**Spark Application Development** Objective: To write and execute a Spark application in Scala, Python, and Java to compute statistical measures of a dataset. Procedure: Develop a Spark application to calculate mean, median, and variance. Run the application in local and cluster deployment modes. Outcome: Understand the basics of Spark application development and execution	06
6	**Cluster Deployment Modes in Spark** Objective: To configure and deploy a Spark application in standalone and cluster modes. Procedure: Configure Spark on a local machine and a cluster. Deploy the application and analyse its execution flow using DAGs, stages, and tasks. Outcome: Learn about Spark's deployment modes and execution mechanisms.	06