AING 364 - Large Scale Computing and Big Data

Department: Computer Engineering

Instructor Information

Name: Tansel Sarıhan **E-mail:** tansel.sarihan@emu.edu.tr **Office:** CMPE 202 **Group:** 01

Assistant Information

Name: Amirhossein Mokabberi **E-mail:** 24500508@emu.edu.tr **Office:** CMPE 226 **Group:** 01 (Coordinator)

Name: Hamidreza Rahimi **E-mail:** hamidreza.rahimi@emu.edu.tr **Office:** CMPE 125

Program Name: Artificial Intelligence Engineering	**Program Code:** 2L

Course Code	**Credits**	**Year / Semester**
AING 364	4	2025 – 2026 Spring

☒ Required Course ☐ Elective Course

Prerequisite(s)

AING 353 – Database Management Systems

Catalog Description

Introduction to big data with databases and data formats (such as JSON, HDF5, XML, and Graph). Distributed file systems (such as Hadoop). Introduction to data analytics (such as clustering with Spark) and dimensionality deduction (e.g. with Spark). Distributed computation models (such as MapReduce); resilient distributed datasets (such as Spark RDDs); structured querying over large datasets (such as Spark Data frames, Hive and SQL); graph data processing systems (such as Spark GraphX and Neo4); stream data processing systems (such as Kafka and MongoDB). Scalable machine learning models (such as Spark MLlib and TensorFlow), distributed and federated machine learning models (such as Spark MLlib and TensorFlow Federated Learning). Optimization, concurrency, recovery and an overview of ethical questions regarding large-scale data.

Course Web Page

https://staff.emu.edu.tr/tanselsarihan/en/teaching/aing-364-large-scale-computing-and-big-data

Textbook(s)

- Kleppmann, M. (2017). Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems. O'Reilly Media, Inc.
- C Müller, A. (2017). Introduction to Machine Learning with Python.
- VanderPlas, J. (2016). Python data science handbook: Essential tools for working with data. O'Reilly Media, Inc.

Schedule and Classrooms

Wednesday	10:30 – 12:20	CMPE 236
Tuesday	12:30 – 14:20	CMPE 236 (LAB: Subgroup 1)
Wednesday	12:30 – 14:20	CMPE 236 (LAB: Subgroup 2)
Friday	10:30 – 12:20	CMPE 236

Topics Covered and Class Schedule (4 hours of lectures per week)	
Week	**Topic**
February 23 – 28	Big Data Fundamentals: Definition of Big Data and 3Vs/5Vs, History (GFS/MapReduce), Scale-up vs Scale-out, compute vs data locality
March 2 – 7	Data Formats & Performance: Columnar (Parquet) vs Row-based (CSV/JSON) data, HDF5, Graph data, Serialization cost (Pickle/Avro), I/O bottlenecks, Benchmarking.
March 9 – 14	Single-Machine Limits & Text Analytics: RAM saturation, Python GIL, Swapping, Pandas, TF-IDF, Cosine Similarity, Memory-efficient processing.
March 16 – 21	Distributed File Systems: HDFS architecture, NameNode/DataNode, Block storage, Replication, Fault tolerance.
March 23 – 28	MapReduce: Paradigm (Map-Shuffle-Reduce), Data flow, limitations, performance bottlenecks.
March 30 – April 4	Apache Spark: RDDs vs DataFrames, Lazy Evaluation, DAG, Spark Execution Model.
April 6 – 11	Structured Querying at Scale: Spark SQL, logical/physical plans, optimizers
Midterm Exams (April 10 - 25)	
April 27 – May 2	Dimensionality Reduction & Clustering: PCA, K-Means on distributed data, MLlib integration.
May 4 – 9	Graph Processing: Graph models, PageRank algorithm, GraphX fundamentals.
May 11 – 16	Stream Processing: Spark Streaming, event time semantics
May 18 – May 23	No Lecture
May 25 – 30	CPU vs GPU & PyTorch Distributed: GPU architecture, PyTorch on single-GPU, Scalable ML pipelines, DDP logic.
Jun 1 – 6	Federated Learning & Scalable NLP: HuggingFace ecosystem, LLM fine-tuning, Tokenization at scale, Privacy-aware ML.
Jun 8 – 13	Optimization, Recovery & Ethics: Caching, partitioning, executor failures, data ownership, ethical considerations
Final Exams (Jun 15 – 27)	
Lab Schedule (Tentative)	
Week	**Topic**
March 23 – 28	Infrastructure & Environment: Setting up Docker, Testing Python GIL and Memory Limits with Local Datasets
March 30 April – 04	NLP and Text Analytics: Pandas-based TF-IDF and Cosine Similarity on Text Dataset
April 6 – 11	Storage and Spark: Working with HDF5, HDFS simulation and Spark RDD fundamentals
May 4 – 9	Machine Learning at Scale: Distributed PCA and Clustering using Spark MLlib.
Jun 1 – 6	Graph Analytics: GraphX operations
Jun 8 – 13	Modern AI: LLM Fine-tuning and inference using HuggingFace on local GPU

Course Learning Outcomes

Upon successful completion of the course, students are expected to have the following competencies:

1. Explain Big Data characteristics and the evolution of distributed data systems
2. Analyze performance limits of single-machine data processing
3. Use Apache Spark for large-scale text analytics
4. Apply dimensionality reduction and clustering on distributed datasets
5. Build basic NLP pipelines using PyTorch and HuggingFace
6. Design stream data workflows
7. Perform graph analytics
8. Explain optimization, fault tolerance, and ethical issues in large-scale systems

	Method	Number	Percentage
Assesment	Midterm Exam	1	40%
	Final Exam	1	45%
	Lab Works	6	15%
	Attendance	Every lecture	0%

Policy on Makeup

For eligibility to take a makeup exam, the student should bring (submit) a doctor's report ***within 3 working days*** of the missed exam. You will have only one make-up for midterm or final exams only. Make-up will be organized after final exam period and will cover all the materials covered during the semester.

Policy on the NG Grade

NG grade will be given in case of missing both midterm and final exams without official excuse.

Policy on Attendance

Attendance will be taken in every lecture but will not be graded.

Policy on Missed Labs

There will be ***no makeup*** for missed labs. If you cannot attend a lab for some reason, you should contact the assistant beforehand so that you can present your work in advance.

Policy on Cheating and Plagiarism

Any student caught cheating in exams or in any other graded course work will automatically fail from the course and may be sent to the disciplinary committee at the discretion of the instructor.

Relationship of the course to ABET Student Outcomes

The course has been designed to contribute to the following student outcomes:
1. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics

Prepared by: Tansel Sarıhan	**Date:** 3 February 2026 / Updated: 11.03.2026