Muhammad-Mahdi Amirpour

09927178533 — muhammadmahdiamirpour@gmail.com — linkedin.com/in/muhammadmahdi-amirpour
github.com/MuhammadMahdiAmirpour

Summary

A Computer Engineering undergraduate with practical experience in data science and machine learning. Familiar with Python, C/C++, Go, and Docker, and well-grounded in algorithms, data structures, and linear algebra. Served as a teaching assistant in formal languages and signals and systems, and contributed to research on anomaly detection architecture. Motivated to refine technical expertise through hands-on projects and collaborative learning.

Skills

- **Languages:** Python, C/C++, Go, Java, Assembly
- **OS:** Linux (LPIC1)
- **Database:** MySQL, Oracle, PostgreSQL
- **Virtualization:** Docker, VMware
- **Frameworks:** Numpy, Pandas, Matplotlib, River, Pytorch
- **Theory:** Algorithm, Data Structures, Linear Algebra, Operating Systems, Signals and Systems

Experience

KNTU **March 2024 – Present**
Research Assistant

- **Assistant Professor:** Dr. B. Pishgoo
- Designed system architecture for anomaly detection

KNTU **March 2024 – February 2025**
Teaching Assistant

- **Assistant Professor:** Dr. H. Khasteh
- Grading quizzes
- Answering students questions
- Designed questions for Exercise Sessions
- Holding Exercise Sessions

KNTU **March 2024 – February 2025**
Teaching Assistant

- **Assistant Professor:** Dr. M. Mordian
- Designed homework questions
- Grading quizzes
- Answering students questions
- Designed questions for Exercise Sessions
- Holding Exercise Sessions
- Designed the final project of the Signals and Systems Course

Education

Khajeh Nasir Toosi University of Technology **Sept 2021 – Present**
Bachelor of Science in Computer Engineering
GPA: (3.5/4) = 17.21/20

Projects

NLP/Vision/ML Project

- **Objective:** Analyze a dataset of book summaries using Python, combining techniques from Natural Language Processing (NLP), Computer Vision, and Machine Learning (ML).
- **Key Steps:**
  - **Data Preprocessing:**
    - Cleaned missing values and standardized text data for analysis.
    - Explored summary lengths and identified common themes in the dataset.
  - **NLP Component:**
    - Summarized book descriptions into shorter, concise versions.
    - Processed text using basic tokenization and stop-word removal techniques.
  - **Computer Vision Component:**
    - Converted condensed text summaries into simple visual representations.
    - Experimented with basic methods to generate images from text.
  - **Machine Learning Component:**
    - Built a simple model to classify book genres based on their summaries.
    - Evaluated the model using standard metrics like accuracy and F1-score.
- **Outcome:** Created a small-scale project that demonstrates the integration of NLP, Computer Vision, and ML techniques. The project provides condensed summaries, simple visualizations, and basic genre classification.

Spectral Clustering Project (Linear Algebra Course)

- **Objective:** Implement spectral clustering algorithms to perform unsupervised learning on datasets, leveraging linear algebra concepts such as eigenvalues and eigenvectors.
- **Key Steps:**
  - **Algorithm Implementation:**
    - Implemented Radial Basis Function (RBF) Clustering, K-Means, and K-Nearest Neighbors (K-NN) Clustering using NumPy for matrix operations.
    - Constructed similarity matrices and computed Laplacian matrices to perform spectral decomposition.
  - **Dataset Analysis:**
    - Worked with the MNIST dataset to classify handwritten digits using spectral clustering techniques.
    - Visualized clustering results to evaluate algorithm performance and identify patterns in the data.
  - **Optimization:**
    - Optimized computational efficiency by vectorizing operations and reducing redundant calculations.
    - Compared clustering results with traditional methods (e.g., K-Means) to assess the advantages of spectral clustering.
- **Outcome:** Developed a robust implementation of spectral clustering algorithms, demonstrating strong understanding of both theoretical concepts and practical applications in machine learning and linear algebra.

Dockerized Full Stack Hotel Booking App

- **Overview:** Developed a full-stack hotel booking application using modern web technologies, containerized with Docker for scalability and portability.
- **Features:**
  - User authentication and role-based access control using Spring Security.
  - RESTful APIs built with Spring Boot for backend services.
  - Frontend developed using React JS for a responsive and dynamic user interface.
  - Database management with JPA Hibernate for seamless integration with relational databases.
- **Outcome:** A fully functional, containerized hotel booking system that demonstrates proficiency in full-stack development and DevOps practices.

Dockerized Distributed File System

- **Overview:** Designed and implemented a distributed file system using Golang, with Docker for containerization to ensure modularity and scalability.
- **Key Features:**
  - Distributed storage architecture allowing multiple nodes to store and retrieve files.
  - Fault tolerance through replication and redundancy mechanisms.
  - Efficient file chunking and metadata management for improved performance.
- **Outcome:** A robust, scalable file system capable of handling large-scale distributed storage requirements.

Chat App Go

- **Overview:** Built a real-time chat application using Go for the backend and JavaScript for the frontend, enabling seamless communication between users.
- **Key Features:**
  - WebSocket-based communication for real-time messaging.
  - User authentication and session management for secure access.
  - Scalable architecture supporting multiple concurrent users.
- **Outcome:** A lightweight, efficient chat application demonstrating expertise in real-time communication systems.

Go-Back-N and CRC Simulation Implemented with Socket Programming

- **Overview:** Simulated the Go-Back-N protocol and Cyclic Redundancy Check (CRC) error detection mechanism using Django (Python) for backend logic and JavaScript for frontend visualization.
- **Key Features:**
  - Backend implemented in Django to handle core functionality, including reliable data transmission, error detection, retransmission, and sliding window protocol management.
  - Frontend developed in JavaScript to provide interactive visualizations of packet flow, error correction, and protocol behavior.
  - Real-time simulation of packet transmission and error handling to demonstrate protocol efficiency.
- **Outcome:** A hands-on demonstration of networking protocols and error detection techniques, showcasing proficiency in backend development and frontend visualization.

Death Arena Game

- **Overview:** Developed a simple command-line interface (CLI) game in Java, where players engage in automated combat against AI-controlled opponents.
- **Key Features:**
  - Automated gameplay with AI-driven opponent behavior.
  - Turn-based combat mechanics with random events and outcomes.
  - Simple scoring system to track player progress.
- **Outcome:** A lightweight CLI game that demonstrates foundational programming skills and object-oriented design principles.

Video Merger

- **Overview:** Created a tool to merge two video files into a single output file, leveraging low-level programming in C and Assembly.
- **Key Features:**
  - Support for various video formats and resolutions.
  - Optimized performance using Assembly for critical operations.
  - Integration with CMake and Makefile for streamlined builds.
- **Outcome:** A high-performance video merging utility demonstrating proficiency in systems programming.