Madhumitha Murthy

+65 8319 1000 | madhumit007@e.ntu.edu.sg | LinkedIn | Portfolio | GitHub | Google Scholar | Singapore

Summary

AI Engineer with 5 peer-reviewed publications across ACL, EACL, IEEE, and Nature Portfolio. Experienced in building agentic AI systems, LLM fine-tuning (LoRA/PEFT), RAG pipelines, drift detection, and production deployment with FastAPI, Docker, and Kubernetes. Applies linear programming, constrained optimisation, and discrete event simulation (SimPy) to real-world ML pipelines. Implements responsible AI practices including hallucination risk scoring and grounding checks on LLM outputs. MSc candidate at NTU with research experience at A*STAR I2R. Available for full-time roles in Singapore from July 2026. Open to AI Engineer, ML Engineer, NLP Engineer, and Data Scientist roles.

Education

MSc in Signal Processing and Machine Learning *Aug 2024 – Jun 2026 (Expected)*
Nanyang Technological University (NTU), School of EEE, Singapore

- Dissertation with I2R, A*STAR on ISAC using 5G for V2V/Infrastructure Communication and Localisation
- Coursework: Genetic Algorithms & ML, NLP, AI & Data Mining, Pattern Recognition & Deep Learning, Real-Time DSP, Machine Vision

Bachelor of Engineering in Computer Science and Engineering *Sept 2020 – June 2024*
Meenakshi Sundararajan Engineering College, Anna University, Chennai, India

Experience

Research Attachment *Oct 2024 – Dec 2025*
Institute for Infocomm Research (I2R), A*STAR, Singapore *NVIDIA Sionna, Python, Blender, OpenStreetMap, OFDM Radar, Kalman Filter, MATLAB*

- Built a high-fidelity 3D ISAC simulation platform using NVIDIA Sionna ray tracing, modelling multipath propagation, Doppler effects, and dynamic occlusions for vehicle sensing and communication in a realistic urban environment (Fusionopolis, Singapore).
- Designed and implemented an OFDM radar algorithm with **Static Clutter Cancellation (SCC)** at **59 GHz / 100 MHz** bandwidth, suppressing static reflections to isolate moving vehicle targets with distinct Doppler signatures.
- Developed a **Doppler-based vehicle trajectory prediction** algorithm forecasting future positions from past sensing estimates, achieving close trajectory tracking at update intervals as short as **0.333s** for proactive beam alignment.
- Implemented sensing-aided beamforming using predicted vehicle positions to update beamforming weights in real time, achieving BER comparable to perfect-DOA benchmarks at lower computational cost.
- First-authored **IEEE ISCAS 2026**; co-authored **npj Wireless Technology** (Nature Portfolio).

Research Assistant *Sept 2025 – Dec 2025*
NTU College of Computing & Data Science (CCDS), Singapore *Python, Peer Evaluation Systems, Literature Review*

- Developed specifications for a bias-resilient peer evaluation system for NTU.
- Conducted literature review on rubric-based peer evaluation methodologies.

Postgraduate Teaching Assistant *Aug 2024 – Jan 2025*
NTU, School of EEE, Singapore *Python, TensorFlow, Mask R-CNN, OpenCV, NumPy*

- Developed lab manual and reference implementation for object detection and instance segmentation using **Mask R-CNN**. Graded assignments for IE2108: Data Structures & Algorithms in Python.

Android Project Trainee *Feb 2024 – Mar 2024*
Zoho Corporation, Chennai, India *Kotlin, Android, Coroutines, REST APIs*

- Built a Kotlin-based banking application with account management and transaction features using Coroutines and asynchronous workflows.
- Won **1st place** at Cliq-Trix 2024 out of 70,000 participants (bot-building competition) and awarded **Rs. 1,00,000** cash prize and offered a full-time position at Zoho.

Projects

Fintech Payment Risk Dashboard | *Java Spring Boot · React · MongoDB · Docker · Groq LLM · Python FastAPI · GitHub Actions* *2026* `github.com/madhumitha-murthy/payment-risk-dashboard`

- Designed and built a full-stack fintech application with **4 Spring Boot microservices** (User, Transaction, Risk, Intelligence) and a **React** dashboard; services communicate via REST, persisted to **MongoDB**, fully containerised with **Docker Compose**.
- Engineered a hybrid ML Risk Service bridging Java to a **Python LSTM Autoencoder** anomaly detection API: weighted scoring (rule-based 40% + ML 60%), auto-flags HIGH-risk transactions, and degrades gracefully if ML service is unavailable.
- Built a GenAI Intelligence Service using **Groq API (Llama 3.1)** for plain-English transaction explainability and natural language → structured filter query parsing over live transaction data.
- CI/CD via **GitHub Actions** builds and tests all 4 services on every push; multi-stage Docker builds for optimised production images.

Agentic AI Tech News Pipeline / *Python,Gemini 2.0 Flash,ChromaDB,Sentence Transformers,RAG,Docker,Kubernetes,GitHub Actions* *2026* GitHub

- Built an end-to-end agentic AI pipeline that autonomously fetches articles from ArXiv, HackerNews, TechCrunch, VentureBeat, and AI research blogs; filters by relevance using keyword scoring; and delivers a styled HTML digest to Gmail at 11:55 PM SGT daily via APScheduler.
- Implemented a RAG query interface using **ChromaDB** vector store and **Sentence Transformers** (all-MiniLM-L6-v2) embeddings; users can query stored articles in natural language with Gemini 2.0 Flash generating grounded answers with source citations.
- Containerised with **Docker** (multi-stage build, non-root user) and deployed as a **Kubernetes CronJob**; CI/CD via **GitHub Actions** (lint, test, Docker build & push to GHCR); pipeline metrics logged per run to JSON with historical evaluation reporting.
- Tracks retrieval relevance score distributions across RAG query runs; flags semantic drift when rolling mean cosine similarity drops $> 2\sigma$ below the 5-run baseline, recommending ChromaDB re-indexing when indexed articles no longer match query topics.

RAG-Powered Document Q&A API / *LangChain,FAISS,HuggingFace,FastAPI,Docker,MLflow,AWS* *2025* GitHub

- Built end-to-end Retrieval-Augmented Generation (RAG) pipeline: PDF ingestion, HuggingFace sentence-transformer embeddings, FAISS vector store, Groq Llama 3.3-70B (LLM) for grounded answers via a FastAPI REST API, containerised with Docker.
- Engineered prompts for context-grounded answering; deployed on **AWS EC2** with PDFs and FAISS index persisted to **AWS S3** for durable cross-session storage.
- Implemented section-aware chunking with natural language context prefixes, improving keyword recall from **28% to 64%** over flat chunking baseline.
- Tracked 3 retrieval configurations (chunk size × top-k) in MLflow; best config achieved **64% keyword recall** at **0.63s** average latency across 5 test questions.
- Evaluated RAG output quality using RAGAS (Faithfulness: **0.783**, Answer Relevancy: **0.932**); applied output guardrails to suppress hallucinated or out-of-context responses.
- Implemented answer grounding check: computes cosine similarity between LLM answer embedding and retrieved context chunks using `all-MiniLM-L6-v2`; API returns `grounding_score` and `hallucination_risk` (low/medium/high/refused) on every query to detect ungrounded responses.

Domain-Specific LLM Fine-Tuning with LoRA / *HuggingFace PEFT,LoRA,DistilBERT,FastAPI,Docker,MLflow,AWS* *2025* GitHub

- Fine-tuned DistilBERT transformer on Financial PhraseBank for 3-class sentiment using LoRA (parameter-efficient fine-tuning), reducing trainable parameters by **98.7%** (887K vs 67M full fine-tune). Best run: **F1 = 0.846, Accuracy = 89.2%**.
- Tracked 3 MLflow experiments (A/B testing across learning rate and LoRA rank). Deployed large language model (LLM) inference via FastAPI REST API on **AWS EC2**; adapter checkpoints stored in **AWS S3**, containerised with Docker and managed with Git CI/CD.
- Registered best adapter in **MLflow model registry**; `evaluate.py` loads the Production model weekly and exits with code 1 if val F1 drops below 0.82; `promote.py` compares Staging vs Production F1 and transitions registry stages; automated via **GitHub Actions** `retrain.yml` (weekly schedule + manual trigger).

Time-Series Anomaly Detection Pipeline / *PyTorch, LSTM Autoencoder, SciPy linprog, SimPy, scikit-learn, FastAPI, Docker, MLflow* *2025* GitHub

- Built anomaly detection pipeline on NASA SMAP satellite sensor dataset (54 channels, 562 anomaly sequences); LSTM Autoencoder achieved **F1 = 0.737, AUC-ROC = 0.857, False Alarm Rate = 0.7%** at deployment threshold derived from training errors — no test labels used. All experiments tracked in MLflow.

- Formulated anomaly segment inspection as a **fractional knapsack linear programme**: decision variables $x_s \in [0, 1]$ represent inspection fraction per segment; objective maximises total anomaly signal captured subject to a **10% inspection budget constraint**. Solved via `scipy.optimize.linprog` (HiGHS back-end); LP captured **+17.6% more anomaly signal** than naive greedy on the same budget.
- Built a **SimPy discrete event simulation** of the downstream inspection workflow: N parallel stations with exponential MTTF/MTTR machine breakdowns; LP-prioritised vs greedy-prioritised job schedules run head-to-head. On a constrained-budget scenario with 2 inspection stations, LP schedule reduced makespan by **24%** (7.6 vs 10.0 time units) and improved machine utilisation from **50% to 76%** vs naive greedy.
- Added **KS-test drift monitor**: rolling window of 200 inference scores tested against training baseline using `scipy.stats.ks_2samp` (flags drift at $p < 0.05$ or mean shift $> 3\sigma$); `/drift/status` endpoint for operational dashboards and retraining triggers.
- Deployed as **FastAPI REST API** (~12ms/window on CPU), containerised with Docker; **150-test suite** (pytest) with GitHub Actions CI/CD.

Industrial Defect Detection Pipeline / *scikit-image,OpenCV,PyTorch,ResNet18,MLflow,FastAPI* *2025* GitHub

- Built an end-to-end defect detection and characterisation pipeline on the **MVTec Anomaly Detection** dataset (industrial surface images: crack, cut, hole, print defect types).
- Applied **scikit-image** preprocessing pipeline: Gaussian denoising, **Otsu thresholding**, morphological cleaning, and **regionprops** measurement extracting area, perimeter, eccentricity, and solidity per detected defect region.
- Applied **OpenCV** for contour detection, image filtering, and mask post-processing; fine-tuned **PyTorch ResNet18** for binary defect classification (normal vs defective), achieving **>90% classification accuracy** and tracked all experiments in **MLflow**.
- Deployed end-to-end inference pipeline as a **FastAPI** REST API returning defect classification, confidence score, and per-region measurements.

Instance Segmentation Mask R-CNN / *TensorFlow,Mask R-CNN,OpenCV,scikit-image,Python,NumPy* *2024* GitHub

- Implemented Mask R-CNN from pretrained COCO weights for simultaneous bounding-box detection and pixel-level instance segmentation; applied **OpenCV** for image preprocessing: resizing, normalisation, morphological filtering, and mask post-processing.
- Extended pipeline with **scikit-image** post-processing: Otsu thresholding, morphological cleaning, and **regionprops**-based region measurement (area, perimeter, eccentricity, solidity) on each detected mask analogous to defect characterisation in engineering inspection pipelines.

Fake News Detection in Dravidian Languages / *XLM-RoBERTa,mBERT,ALBERT,HuggingFace* *Mar 2024* GitHub

- Fine-tuned XLM-RoBERTa, mBERT, and ALBERT for multilingual misinformation detection; achieved Macro-F1 of **0.84** and **3rd place** on the DravidianLangTech@EACL 2024 shared task leaderboard.
- Published at DravidianLangTech Workshop, EACL 2024.

Publications

- **IEEE ISCAS 2026** (Accepted) - ISAC for Intelligent Transportation: Ray-Tracing, Clutter Cancellation, and Sensing-Aided Beamforming.
- **npj Wireless Technology** (Accepted) - Integrated Communication and Sensing: Algorithms & 3D Simulation Insights.
- **EACL 2024**, DravidianLangTech Workshop -Fake News Detection Using Deep Learning Models.
- **CEUR, FIRE 2023** - Sarcasm Detection in Dravidian Languages using Transformer Models.
- **ACL 2023**, LT-EDI Workshop - Transformer Models to Detect Levels of Depression from Social Media Text.

Skills

ML / Deep Learning: PyTorch, TensorFlow, HuggingFace Transformers, scikit-learn, large language models (LLMs), LSTM, transformer

NLP: Transformer fine-tuning, LoRA/PEFT, multilingual NLP, LLM guardrails, XLM-RoBERTa, mBERT

Agentic AI: Agentic pipelines, AI agents, Generative AI, RAG, ChromaDB, FAISS, vector database, Sentence Transformers, vector embeddings, retrieval evaluation, RAGAS, prompt engineering

MLOps / Deploy: FastAPI, REST API, Docker, Kubernetes, CI/CD (GitHub Actions), MLflow, model versioning, drift detection, model monitoring, retraining automation, LangChain, APScheduler, A/B testing, experiment tracking, Git, HuggingFace Spaces, AWS (EC2, S3)

Optimisation & Simulation: Linear programming (LP), fractional knapsack, constrained optimisation, SciPy linprog (HiGHS), genetic algorithms, discrete event simulation (SimPy), MTTF/MTTR modelling

Scientific Computing: NumPy, Pandas, SciPy, Matplotlib

Programming: Python, MATLAB, SQL, Kotlin
Signal Processing: OFDM Radar, 5G/6G NR, ISAC, Beamforming, Delay-Doppler Processing, Channel Modelling
Simulation: NVIDIA Sionna ray tracing, Blender, OpenStreetMap, digital twin

Achievements

- **1st Place**, CliqTrix 2024 (Zoho Corporation) - Bot-Building Competition; awarded Rs. 1,00,000 prize and full-time offer.
- **3rd Place**, DravidianLangTech@EACL 2024 - Fake News Detection Shared Task (Macro-F1: 0.84).
- **Top-10**, FIRE 2023 - Sarcasm Detection in Dravidian Languages (8th and 5th across two tasks).