

MIR ATHFAN ALI

Chicago, IL 60616 | (312) 721-9154 | mali83@hawk.illinoistech.edu | [linkedin.com/in/mirathfan](https://www.linkedin.com/in/mirathfan) | github.com/mirathfan

SUMMARY

Applied AI engineer who builds and ships production LLM systems end-to-end: agents, retrieval-augmented generation, structured generation, and the evaluation harnesses that keep them reliable. Built and shipped AURA, an AI personal stylist app with a LangGraph styling agent, style-memory RAG, closet-only generation that prevents item hallucination, and a regression-gated offline eval suite. Full-stack range across React Native, Firebase, and TypeScript, on a foundation of deep learning, computer vision, and classical ML.

SKILLS

- **Programming:** Python, TypeScript, JavaScript, SQL, Java, C, C++
- **Frontend/Mobile:** React Native, Expo, HTML, CSS, React
- **Backend & Cloud:** Firebase, REST APIs, Cloud Functions, Firestore, Firebase Auth, Firebase Storage, GCP, AWS, Node.js
- **Data/ML Tools:** Git, GitHub, Docker, Linux, Jupyter, PySpark, SHAP, SMOTE, XGBoost, LightGBM
- **AI/ML:** LLM Agents, RAG, LangGraph, Prompt Engineering, Vector Search, PyTorch, TensorFlow, OpenCV, Vision Transformers, Image Processing, OpenAI API, Embeddings
- **Core CS:** Data Structures, Algorithms, Operating Systems, Computer Networks

EDUCATION

Illinois Institute of Technology | *Master of Applied Science, Computer Science - Specialization in Artificial Intelligence* | Chicago, IL Aug 2024 - May 2026

Dayanand Sagar University | *Bachelor of Technology, Computer Science*

Bengaluru, India Aug 2019 - May 2023

EXPERIENCE

AURA, KASAT Labs | *Full-Stack AI Engineer / Founder*

Oct 2025 - Present

- As sole engineer, built and shipped AURA end-to-end across a React Native/Expo frontend, a Firebase (Auth, Firestore, Cloud Functions, Storage) backend, and an OpenAI + LangGraph service layer, taking the AI personal stylist from concept to TestFlight release
- Delivered a complete styling product: a chat-based assistant with an outfit carousel and "why this works" explanations, a saved-outfits lookbook, calendar planning with wear logging, and like/dislike feedback (including disliked-outfit tracking) that feeds back into future recommendations
- Engineered production reliability and privacy: feature-flagged fallback routing, request timeouts (45s/20s), Firestore per-user scoping, TestFlight-ready QA documentation, and a privacy-safe metrics layer (14 event types) that records only counts and timings, never raw prompts, model responses, or embeddings
- Designed the full product solo: a cohesive dark-mode visual system across a dozen-plus screens (styling chat, closet manager, calendar planner, swipe-to-train preference flow, and an insights dashboard), built from a reusable set of card, pill, and outfit-flatlay components

Tech Mahindra | *Software Engineering Intern - Machine Learning*

Jan 2023 - Mar 2023

- Developed TensorFlow-based computer vision modules in Docker containers, using Jupyter for experimentation, to prototype an object-detection feature that allowed the team to assess detection performance on sample images
- Built image preprocessing and model-experimentation pipelines with TensorFlow, standardizing training and evaluation workflows, which reduced data-preparation effort and ensured consistent model benchmarking

PROJECTS

AURA Styling Intelligence Engine [GitHub](#)

Nov 2025 - Present

LangGraph, OpenAI, Firestore Vector Search, Firebase Functions, TypeScript

- Built a closet-grounded outfit-recommendation engine on a **10-node LangGraph** agent that classifies intent and routes through retrieval, generation, refinement, and explanation with a validate-and-repair loop
- Engineered a closed style-learning loop: **18 kinds of user feedback** (likes, saves, wears, dislikes) become **OpenAI text-embedding-3-large** vectors in Firestore-native vector search, retrieved at inference through metadata-normalized, category-balanced lookup as positive/negative style context that personalizes future outfits
- Made generation reliable with zod-schema structured output, item-ID membership checks, role canonicalization, and automatic LLM repair, enforcing closet-valid outfits with **0 hallucinated items** across the eval suite
- Built a regression-gated offline eval harness: **23 golden cases** (including hard edge cases like empty or invalid closets, with graceful-degradation checks), **Precision@5 of 0.94**, plus hallucination, vector-leakage, and agent-regression gates that block any prompt or retrieval change failing the bar (**100% current pass**)

GPU AI Performance Lab

Feb 2026 - May 2026

PyTorch, ONNX, TensorRT, CUDA, C++

- Built a reproducible GPU benchmarking lab on an RTX 3060 Laptop GPU, comparing PyTorch FP32/FP16, ONNX Runtime CUDA, and TensorRT Execution Provider across ResNet18 classification and YOLOv8n object detection workloads
- Optimized ResNet18 inference with TensorRT EP FP16, achieving **0.860 ms p50 latency** at batch size 1 and **3335 samples/sec** at batch size 16, a **+131.6% throughput improvement** over PyTorch FP32
- Benchmarked YOLOv8n object detection with TensorRT EP FP16, reaching **432 images/sec** at batch size 8 and **+84.8% throughput** over ONNX Runtime CUDA FP32
- Implemented a C++/CUDA Monte Carlo option pricing benchmark for European options, reaching **7.25× GPU speedup** at 10M simulated paths while documenting GPU overhead at smaller workloads

EfficientFormer-L1 Image Classification Benchmark

Feb 2025 - May 2025

PyTorch, Vision Transformers, CNNs, Quantization, CIFAR-10

- Implemented an **EfficientFormer-L1-inspired hybrid CNN/Transformer architecture** in PyTorch with modular training, validation, and evaluation pipelines for CIFAR-10 image classification
- Achieved **87% test accuracy** through data augmentation, hyperparameter tuning, and training optimization while benchmarking performance against a ResNet18 baseline
- Reduced inference time by **~25% compared with ResNet18** using quantization and deployment-focused optimization for efficient edge-device inference

KDD Cup 1999 Network Intrusion Detection Pipeline

Feb 2026 - May 2026

R, Python, Random Forest, XGBoost, LightGBM, SHAP, SMOTE

- Classified malicious network traffic across 5 attack categories from 494K+ real-world records using ensemble ML methods
- Built an end-to-end ML pipeline on **494K+ network traffic records**, mapping **23 attack subtypes into 5 classes** and addressing severe class imbalance with feature selection and SMOTE
- Trained and compared **Decision Tree, Random Forest, XGBoost, and LightGBM** models; achieved **99.94% accuracy / 0.832 macro F1** with Random Forest and used SHAP to interpret high-impact traffic features