EDUCATION

Master of Science, Computer Engineering Aug 2023 - May 2025

- New York University, NYC, NY CGPA: 3.9 / 4

Bachelor of Technology, Computer Science and Engineering July 2017 - June 2021

- Jawaharlal Nehru Technological University, India CGPA: 3.92 / 4

SKILLS

Languages: Python, Go, Java, C++, C#, TypeScript, JavaScript, SQL

Full Stack: React, Next.js, Angular, FastAPI, Flask, Spring Boot, Node.js, Express.js, Microservices

Databases: PostgreSQL, MySQL, MongoDB, DynamoDB, Redis, Milvus, pgvector

Data / Infra: Kafka, Amazon MSK, AWS (S3, DynamoDB), GCP (Cloud SQL, Pub/Sub), Docker, Kubernetes, Spark

Testing: Unit Tests, Integration Tests, Test Coverage, pytest, JUnit, Selenium

Tools: GraphQL, Jenkins, Git, SVN, Figma, CloudWatch, Prometheus, Grafana

Libraries: Pandas, NumPy, OpenCV, Matplotlib, NLTK, PyTorch, TensorFlow, Transformers, TRL, HuggingFace, WandB

WORK EXPERIENCE

AI Engineer | Hybridge Implants, New York Sept 2025 – Present

- Architected an event-driven, GCP-hosted AI workflow using Python/FastAPI to ingest Zoom call webhooks, run transcript ETL + rule-based scoring, and deliver post-consultation doctor coaching reports in <5 minutes.
- Built a production scoring RAG with Cloud SQL (PostgreSQL + pgvector), GraphQL, and rubric backed SQL analytics to persist longitudinal doctor performance and generate feedback,

contributing to +130% treatment acceptance and +43% revenue growth.

- Operationalized ML infra with containerized API/worker services, queue-based orchestration, and tool-agnostic CI/CD promotion gates to scale across multi-doctor concurrent workflows while reducing hallucinations by 35% under HIPAA-compliant controls

Software Development Engineer Intern | Optimal Living Systems, New York May 2025 – Sept 2025

- Built an enterprise version of the Optimal Living Systems AI platform by extending and integrating core product frontend, backend, and chatbot etc. repositories (React, Next.js, Golang), enabling modular and client-specific SaaS deployments
- Created a production-ready AMI by provisioning frontend, backend, bots and databases (MySQL, Redis, Milvus) within a lightweight K3s Kubernetes cluster on AWS EC2, enabling seamless deployment to enterprise private clouds
- Implemented tag-scoped RAG retrieval with Milvus filtering at index/query time; added CloudWatch for on-call troubleshooting and supported cross-region migration (‘ap-southeast1‘→‘us-east-1‘) across S3, ECR, RDS, and ElastiCache, cutting latency by 50ms.

Software Engineer | Shure Incorporated, India Aug 2021 – Aug 2023

- Engineered and deployed RESTful APIs with Flask (Python) to streamline the audio analytics data pipeline for Shure Cloud, leveraging AWS services (DynamoDB, S3, Amazon MSK); optimized data flow and reduced processing time by 20%
- Designed scalable software frameworks in Python with Selenium, applying Object-Oriented Design with Factory and Strategy patterns; automated 150+ test cases to improve test coverage and reduce manual efforts by 70%
- Operationalized backend services and CI/CD pipelines using Python and Jenkins, streamlining build, testing, and release workflows, reducing deployment errors by 40%, and improving delivery timelines by 35%

RESEARCH EXPERIENCE

Machine Learning Researcher | New York University, New York May 2024 – Sep 2025

- Engineered a data pipeline using Python, SQL, and Airflow-style orchestration to ingest and clean 118 monthly CMV shards, normalize debate threads, and materialize data into preference pairs, providing ready-to-use data for downstream model training
- Trained LLM-based reward models with Transformers, TRL, and QLoRA on GPU clusters, adding checkpointed recovery and logging loss, precision, recall, and F1 in WandB, which enabled performance monitoring and faster iteration of model improvements
- Implemented an RLHF policy-optimization lifecycle from SFT to GRPO/ORPO, and producing comparative inference artifacts to improve persuasion quality.

PROJECTS

Loan Radar (GitHub)

- Architected end-to-end ML CI/CD for loan-default scoring with containerized training & serving (Docker, FastAPI), model artifact lineage in MLflow (MinIO + PostgreSQL), and automated quality gates (unit/integration tests) before promotion.
- Developed low-latency backend microservices (FastAPI, Flask, Uvicorn) with production observability (Prometheus, Grafana); delivered 0.79ms median and 0.87ms p95 inference latency at 33k+ samples/sec throughput.
- Operationalized retraining reliability via Airflow pipelines and Terraform-provisioned multi-node infra; packaged Ray head/worker, API, MLflow as pod-oriented services for Kubernetes scaling, rolling updates, and rollback-safe deployments.