Hongdao Meng

mycrofthd@gmail.com | github.com/Mycroft-s | linkedin.com/in/hongdao-meng-70222b306 | (718) 3063737

EDUCATION

New York University Sep. 2024 - May 2026 (Expected)
Master of Science in Computer Science GPA: 3.83 New York, NY

Beijing University of Technology Sep. 2020 - Jul. 2024
Bachelor of Engineering in Information Security GPA: 4.00 Beijing, CN

SKILLS

Languages: Python, C/C++, Java, Go, SQL, JavaScript, TypeScript, HTML/CSS, Shell, PHP, MATLAB, LaTeX
Frameworks: Blade, React, Vue.js, Django, Flask, Node.js, Spring Boot, PyTorch, TensorFlow, Pandas, Scikit-Learn
DataBase: MySQL, Redis, MongoDB, MilvusDB, PostgreSQL, DynamoDB, Oracle, Firebase, RocketMQ, Elasticsearch
Tools: Git, Docker, AWS, Azure, CMake, Postman, CI/CD, Jenkins, Nginx, LangChain, OpenCV, Jira, Figma

WORK EXPERIENCE

***Software Engineer Intern* @ TikTok, Recommendation System Infra, Seattle** May 2025 - Sep. 2025

- Developed a Python-based **DAG-DSL** migration toolkit (leveraging Protobuf and regex) to auto-convert legacy configuration files of the Bytedance Feature Service (BFS), a scalable feature extraction service powering TikTok's recommendation system, into standardized DSL definitions, boosting operator migration throughput by 85%.
- Designed and implemented a **Blade-based** C++ operator migration pipeline for BFS, streamlining the workflow to improve operator migration speed by 83.3%, and personally migrated 20 operators.
- Architected and developed a one-stop workflow for User Data Accessor—a data extraction service for recommendation engine—covering service build, auto generate python operators, DSL compilation, and RPC request, enabling collaboration among dozens of engineers and accelerating development by 73.3%.
- Engineered a C++ debug framework for the DAG-based Service Engine, leveraging gflags to control on-demand logging of each node's execution details within topologically ordered DAG subgraphs.

***Software Engineer Intern* @ C2SMARTER Center, New York** Jan. 2025 - Present

- Developed a full-stack RAG-based chatbot system using React, Redux, and Flask, delivering responsive UI and seamless communication, improving user engagement by 33.7% and reducing data retrieval latency by 23.6%.
- Built a scalable back-end infrastructure with Docker, MongoDB, MilvusDB, and RESTful APIs, enabling efficient data storage and retrieval, which improved query performance by 25.6% and reduced deployment setup time by 15.7%.
- Deployed application on AWS EC2 and implemented CI/CD pipelines with Jenkins, ensuring high availability and automated testing, which decreased deployment cycles by 18.7% and supported concurrent users from 5k to 10k.
- Implemented monitoring and observability with Grafana and Prometheus, providing real-time system insights and reducing mean time to resolution (MTTR) by 25.3%.
- Streamlined development and quality assurance processes with Postman for API testing and GitLab for version control, resolving 82% of integration issues pre-deployment and improving team collaboration efficiency by 22%.

***Software Engineer, Founder* @ DeepFake Detection Startup, New York** Sep. 2024 - Dec. 2024

- Developed core modules of deepfake detection web platform using React and TypeScript for seamless real-time interaction, enabling 1,200+ concurrent users and reducing client-side rendering latency by 21.3%.
- Built real-time communication layer using Django and WebSocket for robust middleware communication, delivering real-time updates with $<$180ms latency, improving user task efficiency by 25.6%.
- Deployed backend services on Kubernetes (AWS EKS) with ELB load balancing and HPA policies.
- Optimized PostgreSQL query execution through composite index tuning, reducing average response time by 18%.

***Software Engineer Intern* @ QingTeng, Product R&D Department** Feb. 2024 - Aug. 2024

- Developed a real-time IoT analytics platform using Django and Kafka, processing 720k+ sensor events/day with 80ms P95 latency, enabling predictive maintenance alerts.
- Configured and deployed Nginx as a reverse proxy for both the analytics platform and the full-stack dashboard, optimizing request routing and enhancing system security.
- Automated AWS resource provisioning using Terraform and managed configuration with Ansible, cutting environment setup time from 4 hrs to 50 min across 3 regions.
- Designed a robust Kafka cluster with cross-AZ replication (factor=3), ensuring 99.95% availability and reliable data ingestion for both the real-time analytics and dashboard functionalities.