Yuhan Tan

(607)–233–3661 - yt628@cornell.edu - linkedin.com/in/yuhan-aaron-tan

EDUCATION

Cornell University   **Ithaca, USA**
Candidate for Master of Information Science; GPA: 3.8/4.0   *08/2023 - Present*

University of Liverpool   **Liverpool, UK**
(First Class Hons)BEng in Computer Science and Electronic Engineering; GPA: 3.8/4.0   *09/2021 - 07/2023*

Xi'an Jiaotong-Liverpool University   **Suzhou, China**
Major in Computer Science and Technology   *09/2019 - 07/2021*

SKILLS

Languages: Java, Python, C++, C#, C, SQL, Shell, HTML, R, JavaScript

Framework&Tools: PyTorch, TensorFlow, Numpy, NLP, Computer Vision, Spring Boot, MySQL, Redis, MyBatis Plus, Kafka, Docker, Linux, CentOS, Knife4j, Hadoop, Hive, HDFS, Qt, GIT, AWS, Jira, Agile, LLM, Vue

EXPERIENCE

Graduate Teaching Assistant | Machine Learning, R   **05/2024 - 08/2024**
Cornell University   *Ithaca, NY*

- Assisted **20+** graduate and undergraduate students in mastering applied machine learning techniques, including random forests, neural networks, and LLM prompt engineering, leading to a **20%** improvement in assignment performance scores.
- Conducted weekly office hours to provide personalized guidance on machine learning assignments, resolving over **90%** of student queries related to R, and Shiny.
- Used tools like R and Shiny to create educational visualizations and provide students with clearer insights into model performance and interpretability.

Software Development Intern | Spring Boot, MyBatis Plus, MySQL, Redis   **05/2024 - 08/2024**
NextTier   *Sacramento, CA*

- Implemented **distributed sessions** to synchronize login states across distributed servers by using **Redis**. Enabled single field modification by using **Hash** instead of **JSON** to store user information, reducing memory **by 20%**.
- Worked closely with frontend teams to reduce API response time **by 95%** by **caching** user information in **Redis** and ensuring data integrity with **custom Redis serializers**.
- Improved database write operations by using **custom thread pools** with **CompletableFuture** concurrency, reducing import time for **1 million** rows **from 300 seconds to 54 seconds**.
- Enhanced concurrency control by **Redisson distributed locks** to prevent duplicate team joining and exceeding team capacity, ensuring **mutual exclusion** and **API idempotency**.

Machine Learning Researcher | NLP, BERT, PyTorch   **02/2024 - 05/2024**
American Express   *Ithaca, NY*

- Collaborated with cross-functional teams to develop a method to detect **BERT**'s prediction errors over financial dataset by applying **Louvain** and **HDBSCAN clustering**.
- Developed a method to identify BERT's misclassification patterns by categorizing mislabels into topics, and analyzed BERT's **internal activations** using **Captum** to visualize attention score and uncover error-prone areas.
- Led a team to adjust the model's architecture, introducing a **Specialized Attention Layer** and applying **Weighted Loss** for error-prone classes, resulting in a **13% improvement** in model accuracy.

Software Development Intern | Kafka, Spring Boot   **08/2023 - 12/2023**
Eth Technology   *Newark, CA*

- Coordinated with senior engineers to develop a **streaming microservice** capable of processing over **1000 events** concurrently, using **Kafka** and **Spring Boot** to ensure efficient and **scalable** event handling.
- Consulted with the product team to design and build **REST APIs** for event consumption and publication to Kafka topics, implementing Kafka Consumer and Producer patterns.
- Implemented **unit tests** and **integration tests** using **JUnit** and **Embedded Kafka**, achieving **90% code coverage**; Conducted **end-to-end testing** for different scenarios of data-streaming APIs using **Postman**; Implemented **concurrency testing** & **automatic load testing** process using **Jmeter**.
- Integrated **Spring JPA** and utilized **H2 database** to store events metadata.

PROJECTS

Online Coding Judgement System | Spring Cloud, Vue3, Docker   09/2024 - Present

- **Developed an online judging system** where administrators can create and manage coding problems, and users can search, solve, and submit solutions. **Built a reusable code sandbox** for secure code execution.
- **Designed modular backend architecture**, separating user, problem, and evaluation services. Used **Spring Cloud Gateway** for service aggregation and API routing, simplifying client requests and ensuring security.
- Improved code sandbox stability using **Docker isolation** and enabled secure code execution by passing parameters through **tty** to Docker for enhanced safety. Implemented **strategy pattern** for dynamic problem evaluation, supporting language-specific constraints (e.g., Java memory limits).
- Optimized asynchronous workflows using **RabbitMQ** to queue problem evaluation tasks, reducing response time by **40%** and improving system QPS by **32%**.
- Integrated **Vue3 + Arco Design** for a responsive frontend, enabling features like problem search, editing, and solution submission. Initialized the project with **Vue CLI** and developed a global page layout along with a frontend template for future reuse.

Scalable Microservices with Spring Cloud and Google Cloud Platform   08/2024 - 09/2024

- Migrated the application's database from an **embedded HSQL** to **Cloud SQL** (MySQL), and then to **Spanner**, ensuring higher scalability and reliability through managed cloud infrastructure.
- Implemented asynchronous message processing using **Google Pub/Sub** and **Spring Integration**, utilizing a **message gateway** interface to decouple the messaging system.
- Containerized and deployed microservices to **Google Kubernetes Engine (GKE)** using **Docker** and **Kubernetes** for automated scaling, load balancing, and high availability.

Asynchronous Processing Framework: AaronFlow | Spring Boot, MySQL, Redis   03/2024 - 09/2024

- Implemented a two-layer architecture with Flowsvr (Server) providing **HTTP services via APIs** for **task querying, scheduling, and management**, and Aaron (Worker) pulling and consuming tasks, achieving a separation between **distributed task scheduling** logic and **business** logic.
- Designed database tables by separating task information, configuration, and scheduling, achieving a **loosely coupled** structure that reduces dependencies between tables. Used **indexing** for quick **task retrieval**, allowing flexible **task registration** and **management**.
- Implemented **timeout task monitoring** and **recovery**, using a **polling** mechanism to regularly check task status. Monitored table size to trigger **table partitioning** when thresholds are reached.
- Optimized multi-worker coordination by initially using **MySQL row-level locking** to prevent multiple Workers from pulling the same batch of tasks, and later improved performance by introducing **Redis distributed locks** on the Aaron (Worker) side.
- Implemented performance optimization by conducting **stress testing** with **wrk and Lua scripts** to analyze bottlenecks. Utilized a **MySQL connection pool** and increased the maximum number of connections, improving throughput **from 100 to 500 QPS**.

Machine Learning to Analyze 3D Microscopy Images | Deep Learning, UNet   09/2022 - 05/2023

- Collaborated with cross-functional team to develop and fine-tune an **ensemble of five UNet-based deep learning models** to segment IMPDH2 structures in 3D microscopy images, achieving an average **Dice score of 0.81**.
- Designed and implemented robust pre-processing pipelines to handle domain shifts using **automatic scaling** and **normalization**, improving average Dice scores **by 258%, from 0.279 to 0.72**, across various conditions.
- Developed a **Qt-based** desktop application with multi-threaded processing and replaceable weights files, resulting in a **6000% increase in efficiency** compared to manual segmentation methods.
- This project was awarded the **Edgar Walford Marchant Prize**, and the research paper based on this project will be published in the **BMC Bioinformatics** journal.