Hui Xu

(516) 457-4066 huixucom@gmail.com linkedin.com/in/huihsuxu github.com/huixu11 huixu11.github.io

EDUCATION

State University of New York at Stony Brook — **Stony Brook, NY**
MS in Engineering Artificial Intelligence — GPA: 3.67/4.0 — *Aug 2024 – Dec 2026*

- **Coursework:** Distributed Systems, Natural Language Processing, Deep Learning Algorithms
- **Publication:** MAQuA: Adaptive Mental Health Screening using IRT (EACL 2026) [arXiv]

Beijing Forestry University — **Beijing, China**
MS in Computer Application Technology (4.0/4.0) — BS in Information Systems (3.9/4.0) — *2009 – 2018*

TECHNICAL SKILLS

Languages: Python (Fluent), TypeScript, JavaScript, C, C++ (Medium), Java (Medium), SQL
Web & AI Frameworks: React, Redux, Django, Flask, FastAPI, PyTorch, TensorFlow, LangChain
Databases & Tools: PostgreSQL, SQLite, Redis, MySQL, AWS, GCP, Docker, Git, Jira, Ray
Distributed Systems: Multi-Paxos, Two-Phase Commit (2PC), DistAlgo, Consensus Algorithms

PROFESSIONAL EXPERIENCE

Mastercard — **Beijing, China**
Software Engineer II — *Nov 2021 – Jul 2024*

- Developed **full-stack business analytics platform** ("Test & Learn") for Chinese banks using **Django (backend), React/Redux (frontend), and PostgreSQL/SQLite**, enabling local deployment and data compliance
- Proposed and prototyped migration from in-house multiprocessing framework to **Ray Core**, enabling **distributed execution across clusters** and containerized environments with minimal code changes
- Engineered high-performance **outlier detection algorithm** using statsmodels leave-one-out statistics; **reduced runtime from 9 hours to 10 minutes (98.1% improvement)** through selective computation and vectorization
- Built **Metric Uploader** feature capable of processing **400MB+ CSV files in under 60 seconds** with row-level validation using fully vectorized algorithms and comprehensive unit testing coverage
- Designed and delivered Driver Summary module featuring driver significance analysis and visual summaries using React, Redux hashmaps, and Recharts; collaborated with Product Managers and Tech Leads to refine requirements

Dazhangfang (Chinese Intuit) — **Beijing, China**
Python Engineer — *Jul 2018 – Oct 2021*

- Managed and deployed **large-scale OCR platform (100,000+ lines of code, 10 servers)** integrating recognition engine, invoice verification service, and web service for automated receipt recognition and accounting
- **Optimized database queries and indexing**, improving Invoice Recognition Web Service performance by **99.99%**, dramatically reducing response latency
- Automated invoice verification process, achieving **90% reduction in human intervention** using edit-distance algorithms for text matching and validation
- Implemented **asynchronous task scheduling and message delivery** using APScheduler and Redis as message queue broker, increasing throughput and system reliability
- Designed caching mechanisms for recognition results and optimized end-to-end OCR pipeline on **Cloud Platform**, reducing compute cost and improving scalability

PROJECTS

AgentBeats – Financial AI Benchmark & Agent Security | *Team Lead, Python, FastAPI, LLMs*

- **Leading 5-member team** building dual-agent adversarial evaluation system for AI financial analysts using Python, FastAPI, LLMs, Multi-Agent Systems, MCP, and LangChain
- Integrating BizFinBench.v2 (29K+ Q&A pairs) and designing **Alpha Score metric** for reasoning robustness evaluation
- Developing **FastAPI services and MCP servers** for real-time SEC and Yahoo Finance data access; analyzing prompt injection and LLM tool-use vulnerabilities (LangChain CVE) for agent security research

Scalable Distributed Transaction System | *DistAlgo, SQLite, Multi-Paxos, 2PC*

- Designed and implemented **9-node, 3-cluster replicated transaction system** with Multi-Paxos for intra-shard consensus and **Two-Phase Commit (2PC)** protocol for cross-shard atomic transactions
- Achieved **sub-1.5s leader failover** through heartbeat-based failure detection and automatic re-election; implemented configurable cluster topology (N clusters × M nodes) with automatic shard map generation

Transformer Language Model from Scratch | *Python, PyTorch, NLP, Transformers*

- Engineered Transformer LM from scratch with **BPE tokenizer, RMSNorm, RoPE, multi-head attention, and SwiGLU** layers; built and tested Linear, Embedding, and Attention modules ensuring stable gradients
- Trained on TinyStories/OpenWebText using multiprocessing pre-tokenization and custom **AdamW optimizer**; achieved sub-2 minute BPE training (10K vocab) with fluent text generation and competitive perplexity

ACHIEVEMENTS

Jane Street GPU Mode Hackathon: 10th Place - Optimized latency and accuracy using dynamic batching strategies