Mahyar “Mike” Vahabi

Software Engineer Morgan Hill, CA • mahyarvahabi@gmail.com • 831-332-7980

Technical Skills

Programming Languages: C, C++, Python, JavaScript, SQL, HTML, CSS

Databases, Cloud Services & Dev. Tools: PostgreSQL, MySQL, Git, AWS (EC2, S3, Lambda, RDS, CloudFormation, CloudWatch, Elastic Beanstalk)

Domain Expertise: OOP, RestAPI, Full-stack, Data Structures, Network programming, Machine learning, Databases

Certifications: AWS Certified Developer Associate (Sep. 2024)

Languages: Farsi, English, Spanish

Education

University of California Santa Cruz **M.S. in Computer Science** Santa Cruz, CA Sep. 2024 - Jun. 2025

University of California Santa Cruz **B.S. in Computer Science** Santa Cruz, CA Sep. 2020 - Jun. 2024

- **GPA:** 3.72

Work Experience

AIEA Lab Research Software Engineer Santa Cruz, CA Sep. 2024 - Present

- Aiming to collaborate on innovative projects, such as creating an advanced chatbot or developing a real-time translation service.
- Engineering a system for translating natural language into formal logic, applying advanced **NLP** techniques, and designing an **automated reasoning** framework to mitigate hallucinations in LLM outputs, ensuring higher accuracy and consistency.
- Architecting **semantic consistency** systems, leveraging inference engines and ML algorithms to enhance logical coherence in LLM-generated content, contributing to the robustness of AI models.
- Integrating **external knowledge sources** into LLM reasoning processes, utilizing sophisticated **data integration** techniques to improve factual accuracy and contextual relevance, advancing the system's adaptability.
- Developing visualization tools for LLM-generated arguments, creating human-readable explanations and interactive displays to improve interpretability and user engagement with AI-driven insights.

Baskin School of Engineering Computer Science Tutor Santa Cruz, CA Jan. 2023 - Jun. 2024

- Educated over **1,000** students in Data Structures & Algorithms using **C/C++**, adapted teaching methods to diverse learning styles, and mentored students, which helped many secure internships and research opportunities.
- Evaluated software solutions, deploying testing scripts using **Bash** to assess students' code for **unit tests, functional tests, integration tests**, and **memory leaks** employing tools such as **Valgrind** and custom scripts to automate the grading process, achieving comprehensive testing coverage of **95%**.

Scale AI Software Engineer San Francisco, CA Jun. 2023 - Sep. 2023

- Contributed to the **Software Development Life Cycle** by enhancing generative AI models, leading projects that significantly improved model accuracy.
- Optimized the code generation process for **Google's Gemini**, enhancing efficiency by **80%** by deploying extensive code samples, leveraging **Python**, **C++**, **C**, **JavaScript**, and **MySQL**, achieving a peak performance metric of under **50**ms.
- Revolutionized an intelligent chatbot response system utilizing **TensorFlow** and **PyTorch**, incorporating tokenization, sequence modeling, and named entity recognition to accurately handle coding-related queries, enhancing user satisfaction by achieving a response accuracy of **95%**.
- Developed robust solutions for intricate AI challenges using full-stack expertise, enhancing model performance by integrating **Angular** and **React** with **Node.js**, **Python**, and **Django**, and applying machine learning and deep learning techniques, resulting in a **20%** performance boost.
- Implemented diverse coding solutions by providing code samples in multiple languages, such as **Python** and **C/C++**, leveraging manual memory management, pointer arithmetic, low-level system programming, and **pthreads** concurrency, addressing user prompts with a response accuracy of **85%**.
- Collaborated within a multidisciplinary team including data scientists, software engineers, and product managers, maintaining close communication and asking calibrated questions to remove bottlenecks. Delivered **technical presentations** to the team on new designs, features, and development plans.

Software Projects

Multi-Threaded HTTP Server Jun. 2023 - Dec. 2023

- Developed a robust server with socket programming in **C** and **Python**, to manage network connections and HTTP requests, integrating **semaphores** for thread synchronization, leveraging techniques to mitigate race conditions and ensuring error handling plus failover mechanisms, ultimately achieving system stability with a peak response time of under **20ms**.
- Monitored server performance metrics, identified and resolved bottlenecks to enhance throughput, and implemented rigorous testing and optimization, achieving a server performance increase of **90%** through **load balancing** and **caching**.

Bitcoin Crypto Price Prediction

Oct. 2022 - Nov. 2022

• Developed an **RNN model** featuring **LSTM** and **GRU layers** for predicting Bitcoin price trends and enhanced its efficacy using **Python** and machine-learning techniques, integrating autoregressive recurrent neural network, refining normalization, feature scaling, and employing hyperparameter tuning and regularization, with a prediction accuracy of **75%.**

• Explored and adjusted multiple hyperparameters within **TensorFlow**, **SKlearn**, and **Keras**, enhancing model accuracy, and concurrently developed detailed visualizations of model outcomes using **Matplotlib** and **Seaborn**, which facilitated a deeper analysis and refinement of prediction.

Word Filtering Program inspired by “1984” by George Orwell

Sep. 2021 - Dec. 2021

• Implemented **Bloom Filters** for efficient word creation and membership testing, designed **Hash Tables** to store 'oldspeak' to 'newspeak' translations, and integrated **Binary Search Trees** for effective word location and replacement, to build a Text Censorship program in **C/C++**.

• Optimized search and retrieval operations by implementing **Trie structures** for fast prefix-based searching and auto-completion of censored words, enhancing the overall efficiency of the censorship process.

• Utilized **Linked Lists** to manage dynamic lists of replacement rules, ensuring efficient memory usage and easy updates to the censorship criteria.