Zijie Zhou

Phone: +1 703-826-9289 | **Email:** zhou.zijie@northeastern.edu
Personal Website: https://zijiezhou.me | **GitHub:** github.com/Zijie000

Education

Northeastern University — Boston, U.S
Software Engineering System, M.S. — May2025

Arizona State University — Tempe, U.S
Computer Science, B.S. — May 2022

Tech Stack

Programming Language: Python, Java, Go, Scala, JavaScript, SQL, Lisp
Framework: Spring/Spring boot, Gin, Spark
Infrastructure as Code: Terraform, Packer, GitHub Actions, Jenkins
Cloud & Container: Amazon Web Services (AWS), Google Cloud Platform(GCP), Kubernetes, Docker

Academic Project

Real-Time Social Media Keywords Sentiment Analysis System **Location:** Boston **Jan 2025 ～ May2025**

Designed and implemented a real-time, distributed sentiment analysis system for social media platforms (X, Reddit) based on user-defined keywords and time intervals. The architecture emphasizes scalability, fault tolerance, and stream processing.

- Developed ingestion layer with **AKKA Actors** and **Kafka brokers** to parallelly collect and publish social media posts based on keywords and time range.
- Implemented **Spark Streaming** pipeline to consume and normalize Kafka messages in a distributed fashion.
- Integrated **Hugging Face NLP** models for real-time sentiment analysis and used **Spark MLlib (TF-IDF)** to extract attitude-reflecting keywords and perform classification.
- Stored processed data in **PostgreSQL** with a custom schema, supporting efficient querying and scheduled cleanup.
- Built a visualization layer to present trends over time, average sentiment score, representative keywords (word clouds), and category-specific sentiment breakdowns.
- Ensured immutability and parallel safety using functional programming principles in **Scala** and **Spark**; all asynchronous processing managed via **Futures** and actor-based design.

Cloud Computing & Cloud Native **Location:** Boston **Sep 2024 ～ Dec2024**

- Developed and maintained a **RESTful API** user management system using **Golang**, **Gin**, and **GORM** (ORM), delivering efficient and scalable solutions.
- Using **Terraform** defines the **VPC** with multiple private and public subnets. The **RDS** database resides in a private subnet, blocking direct internet access to ensure data security. The core application is deployed in the public subnet.
- Configuring application's **load balancer (ELB)** and **Auto Scaling group** configured in the public subnet, with its domain linked to **Route 53** via an A record, using **TLS/SSL** for secure **HTTPS** encryption.
- Setting up the **S3 bucket** for user image storage, and **AWS Lambda** (serverless) is used to deploy an email verification function, enhancing the user experience and interaction flow.
- Using **Packer**, creating **EC2** images with **HCL** files, embedding a pre-configured Golang Gin RESTful API application to ensure efficient application delivery.
- Hosting the Golang web application source code and **Packer** files in a **GitHub repository**, with a **CI/CD** pipeline implemented via **GitHub Actions**. Each code change undergoes **integration testing**, which must pass before merging. Successful merges trigger Packer to build and upload the EC2 images to **AWS**.

Intern Experience

Automated Data Collection RPA for E-commerce Platforms Mar 2021 – July 2021

Developed a Robotic Process Automation (RPA) solution to automate large-scale data collection from Taobao and Ele.me. The RPA system continuously retrieves real-time product data from these retail platforms, ensuring up-to-date and accurate information for the business analytics department.

- Designed and implemented the **RPA** using Java in combination with **Selenium** and **Appium**, enabling automated data extraction from both web and mobile interfaces.
- Integrated the **RPA** with external **Android** devices through Appium, allowing the system to interact with graphical user interfaces (GUIs) on mobile platforms.
- Optimized the **RPA** workflow to handle large volumes of product data efficiently, minimizing downtime and ensuring continuous operation.