

Adriza Mishra

azirdamishra@gmail.com | +91-9791116882 | linkedin: adriza-mishra | github: azirdamishra

Technical Skills

Programming Languages: Python, Java, TypeScript

Tools: Git, Databricks, PostgreSQL, Kubernetes, Docker, Azure Cloud, Helm, GitHub Actions, Redis

Libraries & Frameworks: FastAPI, MLFlow, TensorFlow, Keras, PySpark, Behave/BDD, Microservices, CI/CD

Education

Indian Institute of Technology Madras

July 2018 – June 2023

Dual Degree in Biotechnology

CGPA: 8.02

CBSE Std. XII

May 2016 – May 2018 92.1%

Professional Experience

Software Engineer II

July 2023 – Present

Honeywell — AI/ML Platform Engineer

- Engineered an enterprise ML training orchestration platform on Databricks using Python, MLFlow, and PySpark to automate model lifecycle management for 100+ models across 20+ multi-tenant clients. Reduced manual setup by 80% while maintaining 99.5% job success rate in production.
- Developed FastAPI-based microservices, implementing async endpoints for model training orchestration, security scanning, and inference serving. Leveraged Pydantic validation, dependency injection, and background tasks to handle concurrent requests.
- Implemented autoencoder-based anomaly detection model for industrial IoT predictive maintenance with automated drift monitoring across 50+ variables, significantly reducing manual maintenance overhead. Extended support to 10+ customer model types across TensorFlow, PyTorch, and scikit-learn frameworks.
- Designed and filed a patent for comprehensive security validation pipeline integrating 5 scanning tools (Coverity, ModelScan, Twistlock, Malware and File Extension Scan, Black Duck Hub) to perform 8+ automated checks. Cut validation time by 75% to under 10 minutes, identified 40+ critical vulnerabilities, and maintained 100% SOC 2 compliance for model onboarding.
- Developed an automated deployment system that packages models into containers and deploys to Kubernetes or Databricks. Reduced deployment time from 3 days to 30 minutes and cut infrastructure costs by 35% through optimized container sharing.
- Built a high-availability ML inference service processing 10K+ daily predictions across 50+ production models; sub-100ms API latency. Implemented async Delta Lake writes & drift detection, reducing model debugging cycles by 60%.
- Led quality validation for Honeywell's flagship LLM maintenance assistant, winning the company-wide bug bash competition. Authored a comprehensive whitepaper on LLM testing strategies and developed proprietary testing frameworks that resulted in multiple internal patent submissions.
- Developed a decoupled test automation suite using TypeScript WebDriverIO and Python Behave, automating 150+ scenarios with 90% coverage across UI and API layers. Compressed regression testing from 40 hours to 5 hours per sprint, enabling daily deployments and improving defect detection speed by 60%.

Tools: MLFlow, FastAPI, Databricks, Unity Catalog, TensorFlow, Coverity, ModelScan, Twistlock, Black Duck Hub, Azure Cloud, Postgres, Azure Delta Lake, Kubernetes, Python Behave, Typescript, WebdriverIO

Portfolio

Eatery Recommender — Smart Group Dining

November 2024 – Present

- Built a context-aware restaurant recommender that computes centroids from multiple locations, queries Google Maps for candidates, and ranks results using engineered features including distance, ratings, price alignment, and dietary compatibility.
- Integrated an LLM layer to augment recommendations with contextual information — identifying signature dishes and tagging venues with semantic attributes (group-friendly, budget-conscious, etc.), generating concise, explanation-driven recommendations on top of the numeric scoring engine.
- Prototyped a React + Tailwind interface with map-based visualization, interactive session creation (cuisine chips, budget sliders, dietary filters), and a conversational assistant interface that structures ambiguous group preferences into inputs.

Tools: FastAPI, Git, Google Maps API, OpenAI, Pandas/NumPy, Google Maps JavaScript API, Axios, React + Tailwind CSS, Postgres, Redis

Lox Programming Language Interpreter

February 2024 – August 2024

- Implemented a tree-walk interpreter for the Lox programming language in Java, building a complete pipeline from lexical analysis through recursive descent parsing to runtime evaluation. Added support for closures, lexical scoping, classes, and inheritance, with optimized scope resolution through cached variable lookups in nested environments.
- Extended the base interpreter with custom data types (lists and dictionaries) and refactored the error reporting system with color-coded diagnostics. Implemented environment-chain variable resolution and call stack management for nested closures with dynamic dispatch for method calls, improving both performance and developer experience.

Tools: *Java, Maven, Git, Java Collections Framework, JUnit, Recursive Descent Parsing, Visitor Pattern, Environment Chaining*

HylyTool Chrome Extension — Study Tool

May 2024 – July 2024

- Built a study tool browser extension using TypeScript and Chrome's Manifest V3 APIs that lets users highlight and filter text with persistence across browser sessions and device syncs. Used Chrome Storage API with URL-based keying to maintain user data reliably.
- Designed a three-tier text-matching algorithm that compares nodes and traverses the DOM hierarchy to identify and restore specific text occurrences from duplicate content, achieving 100% highlight accuracy on page reloads even with dynamic content.
- Implemented real-time content manipulation using DOM APIs (TreeWalker, MutationObserver, Range/Selection) and message passing between extension components, enabling instant highlight application and blur effects across multiple tabs simultaneously.

Tools: *TypeScript, Webpack, Chrome APIs: Manifest, Storage, Runtime, Scripting, Tabs. DOM APIs: TreeWalker, MutationObserver, Selection, Range*