# Rajeev Atla

US Citizen | [732-209-3995](tel:732-209-3995) | [rajeevatla101@gmail.com](mailto:rajeevatla101@gmail.com) | [github.com/RajeevAtla](https://github.com/RajeevAtla) | [linkedin.com/in/rajeev-atla](https://linkedin.com/in/rajeev-atla) | [rajeevatla.com](https://rajeevatla.com)

## EDUCATION

**Rutgers University - School of Engineering**  Sep 2025 — Dec 2026
*Master of Science* in Computer Engineering (Specialization in Machine Learning)  *New Brunswick, NJ*

Coursework: Reinforcement Learning, Multimodal AI, High Performance/Distributed Computing

**Rutgers University - School of Engineering**  Sep 2021 — May 2025
*Bachelor of Science* (Triple Major) in Computer Engineering, Computer Science, and Data Science  *New Brunswick, NJ*

Recipient of the Eleanor and Samuel Sneath Endowed Merit Scholarship for Engineering Students

Coursework: AI, Distributed Deep Learning, Data Science, Statistical Learning, Computer Vision

## SKILLS

- **Programming Languages**: Python, R, SQL, Java, C/C++/CUDA, Rust, Bash
- **AI/ML Libraries**: NumPy, PyTorch, JAX, TensorFlow, Keras, Pandas, Scikit-Learn, Transformers, LangChain/LangGraph
- **Data Visualization**: Matplotlib, Seaborn, Plotly, Tableau
- **Cloud & DevOps**: AWS, Microsoft Azure, OCI (Oracle Cloud Infrastructure), GitHub Actions, Docker, Kubernetes
- **Tools & Databases**: Jupyter Notebooks, Apache Kafka, Git, Linux (Ubuntu), PostgreSQL, MongoDB, Jira

## CERTIFICATIONS

- **AWS**: [Certified Cloud Practitioner](), [Certified Machine Learning Specialist](), [Certified AI Practitioner]()
- **Oracle (OCI)**: [AI Foundations Associate](), [Generative AI Professional](), [Data Science Professional](), [Vector AI Search Professional]()

## WORK EXPERIENCE

**AI Engineering Intern**  May 2024 — September 2024
Atlait Inc.  *Remote*
- Developed a Python-SQL compression script for form data, **reducing storage costs by 7%** for enterprise clients
- **Accelerated mean response time by 96 milliseconds** by integrating PyTorch inference models into Kafka microservices
- Created a Retrieval-Augmented Generation (RAG) system for an AI-powered search tool
- Optimized CI/CD pipeline to **speed up build times by 13%** ensuring efficient development cycles

## PROJECTS

**dexMCP**  [https://bit.ly/dexmcp](https://bit.ly/dexmcp)
- Engineered Model Context Protocol (MCP) server exposing **5+ reusable tools** and **5+ Pydantic models**
- Implemented parameter validation across **20+ typed fields** and **100% of tool inputs**
- Built asynchronous clients using DSPy and LangChain to auto-discover tools and execute multi-step requests

**DocuMint**  [https://bit.ly/DocuMint](https://bit.ly/DocuMint)
- Built a 5-agent LangGraph + Gemini doc-modernizer with Gradio, achieved **90%+ modernization coverage** on sample docs, **cut manual edit time 50%** with a **4-tab UX**, hardened with **8 deterministic pytest cases** and network-safe skips
- Authored modular AI agents (fetcher, analyzer, researcher, generator, quality-checker) with structured prompts and severity-prioritized research, **lifting modernization accuracy by 35%** and **trimming LLM API spend by 20%**

**SuperconGAN**  [https://bit.ly/3z7JaqZ](https://bit.ly/3z7JaqZ)
- Built a PyTorch-based GAN to model superconductivity data of various materials, enhancing generative AI applications
- Extracted and processed **80,000+ dataset entries** from the UCI ML Repository using Pandas efficiently
- Released Python package on PyPI, achieving over **80,000+ downloads** and widespread adoption
- Authored a LaTeX paper on findings and future scope, **incorporating 500,000+ data points** effectively

**Cityscape (2nd Overall at HackExeter 2021)**  [https://bit.ly/3OZjJ07](https://bit.ly/3OZjJ07)
- Designed and implemented a city tour mobile app, resulting in **100+ vivid city tours** for users
- Wrote controllers and models for MongoDB using MongooseORM to store **30+ kB of geographic data** in NoSQL schema
- Built mobile user interface allowing users to search, review, rank, and explore **100+ tours** using Flutter/Dart
- Constructed REST API using Express.js and nodemon to **increase development velocity by 20%** with hot-reloading