

# Rajeev Atla

Machine Learning Engineer | GenAI, RAG & Search Systems Specialist

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

## EDUCATION

<b>Rutgers University - School of Engineering</b> <i>Master of Science in Computer Engineering (Specialization in Machine Learning)</i>	Sep 2025 — Present New Brunswick, NJ
<b>Rutgers University - School of Engineering</b> <i>Bachelor of Science (Triple Major) in Computer Engineering, Computer Science, and Data Science</i>	Sep 2021 — May 2025 New Brunswick, NJ

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

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

Coursework: AI, ML, Distributed Deep Learning, Data Science, Robotics and Computer Vision, Info and Network Security

## SKILLS

- Programming Languages:** Python, R, SQL, Java, C/C++/CUDA, JavaScript/TypeScript, Rust, Bash
- AI/ML:** NumPy, PyTorch, JAX, TensorFlow, Keras, Pandas, Scikit-Learn, OpenAI API, LangChain/LangGraph, OpenCV, DSPy, RAG, HuggingFace (Transformers, Tokenizers, Datasets, Diffusers), vLLM, pgvector, Pydantic, FastAPI, NLTK, spaCy
- Data Visualization:** Matplotlib, Seaborn, Plotly, Tableau
- Cloud & DevOps:** AWS, Microsoft Azure, OCI, GCP, GitHub Actions (CI/CD Pipeline), Docker, Kubernetes, Slurm
- Tools & Databases:** Jupyter, PySpark, Kafka, Git, Linux (Ubuntu), PostgreSQL, MongoDB, Jira, ROS2, Codex, Claude Code

## 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

<b>AI Engineering Intern</b> Atlait Inc.	May 2024 — Sep 2024 Remote
<ul style="list-style-type: none"><li>Developed a Python-SQL compression script for form data, <b>reducing storage costs by 7%</b> for enterprise clients</li><li><b>Accelerated mean response time by 96 milliseconds</b> by integrating PyTorch inference models into Kafka microservices</li><li>Created a <b>&gt; 1TB</b> RAG-PySpark system, utilizing A/B testing to evaluate and optimize AI-powered search accuracy</li><li>Optimized CI/CD pipeline to <b>speed up build times by 13%</b> in an Agile environment, ensuring efficient development cycles</li></ul>	

## PROJECTS

<b>raceformer</b>	<a href="https://bit.ly/raceformer">https://bit.ly/raceformer</a>
<ul style="list-style-type: none"><li>Engineered a high-fidelity “Real-to-Sim” validation pipeline processing <b>30GB of multimodal sensor data</b> (LiDAR, camera, radar) on 4x A100s, utilizing JAX-based vision-language model to generate ground truth scenarios for critical edge case simulation</li><li>Achieved a <b>95% pass rate on safety metrics</b> by leveraging geometric priors to fine-tune RL policies, establishing clear performance baselines and <b>outperforming standard models by 35%</b> in collision avoidance testing</li></ul>	

<b>dexMCP</b>	<a href="https://bit.ly/dexmcp">https://bit.ly/dexmcp</a>
<ul style="list-style-type: none"><li>Engineered Model Context Protocol (MCP) server exposing <b>5+ reusable tools</b> and <b>5+ Pydantic models</b></li><li>Implemented parameter validation across <b>20+ typed fields</b> and <b>100% of tool inputs</b></li><li>Built asynchronous clients using DSPy and LangChain to auto-discover tools and execute multi-step requests</li></ul>	

<b>DocuMint</b>	<a href="https://bit.ly/DocuMint">https://bit.ly/DocuMint</a>
<ul style="list-style-type: none"><li>Built a 5-agent LangGraph + Gemini API doc-modernizer with Gradio, achieved <b>90%+ modernization coverage</b> on sample docs, <b>cut manual edit time by 50%</b> with a <b>4-tab UX</b>, hardened with <b>8 deterministic pytest cases</b> and network-safe skips</li><li>Authored a modular multi-agent system with structured prompts and severity-prioritized research, <b>lifting modernization accuracy by 35%</b> and <b>trimming LLM API spend by 20%</b></li></ul>	

<b>SuperconGAN</b>	<a href="https://bit.ly/3z7JaqZ">https://bit.ly/3z7JaqZ</a>
<ul style="list-style-type: none"><li>Built a PyTorch-based GAN to create synthetic superconductivity data of various materials, enhancing generative AI applications</li><li>Extracted and processed <b>80,000+ dataset entries</b> from the UCI ML Repository using Pandas efficiently</li><li>Released Python package on PyPI, achieving over <b>80,000 downloads</b> and widespread adoption</li></ul>	