

Rajeev Atla

AI-Native Software Engineer Building Scalable, Secure, Smart Systems

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

EDUCATION

Rutgers University - School of Engineering <i>Master of Science in Computer Engineering (Specialization in Machine Learning)</i>	Sep 2025 — Dec 2026 New Brunswick, NJ
Rutgers University - School of Engineering <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, LLMs, 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, PowerBI
- Cloud & DevOps:** AWS, Microsoft Azure, OCI, GCP, GitHub Actions (CI/CD Pipeline), Docker, Kubernetes, Slurm
- Tools & Databases:** Jupyter, PySpark, Hadoop/Hive, Git, Linux, 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

AI Engineering Intern Atlait Inc.	May 2024 — Sep 2024 Remote
<ul style="list-style-type: none">Developed a Python-SQL compression script for form data, reducing storage costs by 7% for enterprise clientsEngineered PyTorch inference models for real-time predictions, optimizing latency by 96ms and enabling faster decision-makingCreated a > 1TB RAG-PySpark system, utilizing A/B testing to optimize AI-powered search and recommendation accuracyOptimized Airflow-Hadoop data pipeline to speed up analysis by 13% in an Agile environment, speeding up development	

PROJECTS

SuperconGAN	https://bit.ly/3z7JaqZ
<ul style="list-style-type: none">Built a PyTorch-based GAN to create synthetic superconductivity data of various materials, enhancing generative AI applicationsExtracted and processed 80,000+ dataset entries from the UCI ML Repository using Pandas efficientlyReleased Python package on PyPI, achieving over 80,000 downloads and widespread adoption	
DocuMint	https://bit.ly/DocuMint
<ul style="list-style-type: none">Built a 5-agent LangGraph + Gemini API doc-modernizer with Gradio, achieved 90%+ modernization coverage on sample docs, cut manual edit time by 50% with a 4-tab UX, hardened with 8 deterministic pytest cases and network-safe skipsAuthored a modular multi-agent system with structured prompts and severity-prioritized research, lifting modernization accuracy by 35% and trimming LLM API spend by 20%	
raceformer	https://bit.ly/raceformer
<ul style="list-style-type: none">Engineered a high-fidelity “Real-to-Sim” validation pipeline processing 30GB of multimodal sensor data (LiDAR, camera, radar) on 4x A100s, utilizing JAX-based vision-language model to generate ground truth scenarios for critical edge case simulationAchieved a 95% pass rate on safety metrics by leveraging geometric priors to fine-tune RL policies, establishing clear performance baselines and outperforming standard models by 35% in neural path planning and risk avoidance	
dexMCP	https://bit.ly/dexmcp
<ul style="list-style-type: none">Engineered Model Context Protocol (MCP) server exposing 5+ reusable tools and 5+ Pydantic modelsImplemented parameter validation across 20+ typed fields and 100% of tool inputsBuilt asynchronous clients using DSPy and LangChain to auto-discover tools and execute multi-step requests	