Sai Teja Srivillibhutturu

saiteja.srivillibhutturu@gmail.com | linkedin.com/in/saitejasrivillibhutturu | saitejasrivilli.github.io

Education

University of Texas at Arlington Arlington, TX
Master of Science in Computer Science,GPA: 4.0/4.0 *Aug 2023 – May 2025*

- Relevant Coursework: Neural Networks, Machine Learning, Artificial Intelligence, Computer Vision, Data Mining, Software Engineering, Scalable Search and Optimization

Andhra University Visakhapatnam, India
Bachelor of Technology in Computer Science *Jun 2015 – Apr 2019*

SKILLS

Languages: Python, Java, JavaScript, SQL, HTML/CSS
ML Frameworks: PyTorch, TensorFlow, Hugging Face Transformers, OpenAI, vLLM, ONNX Runtime, CUDA
ML Models & Methods: BERT, CLIP, YOLOv8, Deep SORT, OpenCV, RAG, LLM Fine-Tuning, Quantization
LLM & Deployment: Pinecone, Speculative Decoding, Gradio, Flask, React.js, REST/SOAP APIs
Cloud & DevOps: AWS (EC2, S3, SageMaker), Docker, GitHub Actions, Git, JUnit, Mockito

EXPERIENCE

DentalScan (ReplyQuickAI LLC) Remote
Machine Learning Engineer Intern *Dec 2025 – Present*

- Developing supervised CNN models for intra-oral image classification across 6 clinical categories (gingivitis staging, plaque detection, recession classification) trained on a 50,000+ labeled dental image dataset
- Built an automated ML retraining pipeline on AWS (S3, EC2, SageMaker, Lambda) with dentist-corrected label feedback loops and CI/CD integration via GitHub Actions, reducing model update cycles from weeks to hours
- Containerized ML inference endpoints using Docker for reproducible deployments; implemented data augmentation and class-balancing workflows, reducing class imbalance by 40% and improving minority-class recall by 22%
- Collaborated with backend engineers to deploy real-time ML inference APIs for dental analysis from smartphone-captured images, achieving sub-500ms inference latency
- Implementing experiment tracking and dataset versioning pipelines, maintaining reproducibility across 15+ model iterations with systematic evaluation of precision, recall, and F1 across all clinical categories

University of Texas at Arlington Arlington, TX
Graduate Research Assistant – TopGPT *Jun 2025 – Present*

- Fine-tuned GPT models on 3+ domain textbooks and built a RAG pipeline over 1,000+ research paper chunks stored in Pinecone on AWS, backed by PostgreSQL and Kafka for retrieval metadata and streaming ingestion, enabling context-aware academic Q&A with 85%+ retrieval relevance
- Built a full-stack React interface with Flask backend for researchers to interactively query academic papers, visualize retrieval results, and compare model responses, reducing research lookup time by 50%
- Designed chunking, embedding, and retrieval strategies for large-scale academic corpora, optimizing query latency by 30% through hybrid search and metadata filtering in Pinecone
- Deployed the end-to-end RAG application on AWS EC2 with Docker containers, implementing automated health checks and logging for production-grade reliability

University of Texas at Arlington Arlington, TX
Graduate Teaching Assistant – CTMap Research *Aug 2024 – May 2025*

- Built an LLM-powered 6G path optimization system CTMap integrating OpenStreetMap real-time location data with Sionna 6G channel simulation for next-generation network routing
- Fine-tuned LLMs on 10K+ Dijkstra-generated optimal paths with validated geospatial coordinates; applied model to Sionna 6G output, improving path accuracy by 25% over baseline heuristics
- Processed and validated coordinate datasets from OpenStreetMap, building an automated data pipeline that generated 10K+ labeled start-to-end paths for supervised LLM fine-tuning
- Benchmarked 4+ LLM architectures on path prediction tasks, evaluating latency, accuracy, and token efficiency to select the optimal model for real-time 6G routing inference

Tata Consultancy Services Chennai, India
System Engineer *May 2021 – May 2023*

- Architected middleware integrations connecting 5+ financial systems via REST/SOAP APIs on Linux-based infrastructure, automating invoice-to-payment reconciliation workflows reducing manual processing time by 40%
- Led a system migration initiative, building automated ETL pipelines that transferred 500K+ financial transaction records from on-premise databases to modernized services with 99.8% data integrity validation and zero production downtime
- Owned production support and on-call rotation for mission-critical financial services in a 10-engineer team, resolving P1/P2 incidents within SLA and sustaining 99%+ system availability across quarterly close periods
- Built automated alerting pipelines for transaction failures, reconciliation mismatches, and SLA breaches during month-end close, increasing release velocity by 30% while reducing regression defects by 28%
- Drove delivery planning and sprint coordination across a 10-engineer team with product managers, QA, and architects, achieving 98% on-time delivery across 12+ quarterly releases

Tata Consultancy Services Chennai, India

Assistant System Engineer *Jun 2019 – Apr 2021*

- Built Java Spring Boot REST/SOAP APIs for automated invoice validation and three-way matching (purchase order, receipt, invoice) across the accounts payable workflow, reducing manual approval cycles for enterprise clients
- Optimized 50+ SQL queries and stored procedures across financial reporting pipelines using index tuning, query restructuring, and execution plan analysis, cutting average execution time by 35%
- Implemented automated test suites using JUnit and Mockito achieving 85%+ code coverage across core financial modules, reducing post-release defects by 25% and establishing testing standards adopted by the wider team
- Triaged and resolved 50+ production incidents through log analysis and database debugging, building the operational expertise that led to full on-call ownership in the senior role

Projects

vLLM Throughput Benchmark – Python, vLLM, Hugging Face, CUDA, NVIDIA T4/L4

github.com/saitejasrivilli/vllm-throughput-benchmark

- Benchmarked vLLM vs Hugging Face under concurrent request load, achieving 18.6× throughput gains through dynamic token batching, KV-cache optimization, and improved GPU utilization
- Evaluated end-to-end latency, queueing behavior, and cost efficiency under bursty traffic, demonstrating 3.6× lower cost per million tokens with sub-100 ms P95 latency for scalable LLM serving

Quantization & Speculative Decoding Benchmark – PyTorch, CUDA, ONNX, bitsandbytes

github.com/saitejasrivilli/quantization-speculative-decoding-benchmark

- Evaluated 5 quantization methods (FP16, INT8, INT4-NF4, GPTQ, AWQ) and speculative decoding, achieving 75% memory reduction and 3.3× inference speedup while preserving 99%+ model accuracy
- Optimized latency, throughput, and cost across GPU and cross-platform deployments via ONNX Runtime and speculative decoding, demonstrating 2–3× decoding acceleration and $3.6M projected annual cost savings at scale

Multi-Object Tracking for Autonomous Systems – YOLOv8, Deep SORT, Kalman Filter, OpenCV

github.com/saitejasrivilli/Multi-Object-Tracking-for-Autonomous-Systems

- Developed a multi-object tracking system using YOLOv8 + Deep SORT + Kalman filtering, achieving 30–60 FPS on GPU with 90%+ mAP and 85%+ MOTA on MOT17/KITTI datasets across 80+ object classes
- Built real-time visualization pipeline with trajectory forecasting for autonomous navigation and collision avoidance, deployable via batch processing and Gradio web interface

LLM Brand Safety System – PyTorch, BERT, CLIP, Gradio, Multi-modal Inference

github.com/saitejasrivilli/llm-brand-safety-system

- Built a multi-modal brand safety system combining fine-tuned BERT for text and CLIP for images, delivering real-time content moderation with category-level risk scoring across 6 text and multiple visual safety categories
- Designed explainable decision pipelines with policy-based escalation and Gradio deployment, enabling ad-safe inference and human-understandable audit trails reflecting Trust & Safety ML practices

DistributedKVStore – Python, Distributed Systems, REST APIs

github.com/saitejasrivilli/DistributedKVStore

- Designed a distributed key-value store using consistent hashing, HashMap-based storage, and replication, achieving 99% read availability under simulated node failures
- Implemented leader-based coordination and fault-tolerant request routing, reducing write latency by 25% while supporting horizontal scaling across multiple nodes
- Built REST APIs for read/write operations demonstrating core distributed system design patterns including partitioning, replication, and consistency trade-offs