Prudhvi S Chundru

Lubbock, TX | (806)-559-6739 | prudhvichundru01@gmail.com | LinkedIn | GitHub | Porfolio

PROFESSIONAL SUMMARY

AI / Generative AI Engineer with expertise designing and deploying large-scale ML and LLM systems. Led the architecture of an enterprise RAG platform processing ~12 M monthly financial transactions at U.S. Bank and built a PHI-compliant RAG solution for healthcare data at Health Care Service Corporation, delivering measurable improvements in accuracy, latency, and compliance. Proven track record in fraud-detection pipelines, cloud-native MLOps, and cost-efficient fine-tuning (LoRA/PEFT). Seeking to apply this experience to create reliable, scalable AI products that directly drive business outcomes.

TECHNICAL SKILLS

- **Programming Languages**: Python, SQL, Java, Bash
- **Generative AI, LLMs & NLP**: GPT-4, Hugging Face Transformers, LangChain, Retrieval-Augmented Generation, FAISS, Pinecone, Qdrant, LLM Evaluation, LoRA/ PEFT (Pytorch), Cross-Encoder Reranking
- **Machine Learning & Deep Learning**: Scikit-learn, TensorFlow, PyTorch, XGBoost, LightGBM, Model Calibration
- **Data Engineering & ETL**: Apache Spark, PySpark, Airflow, Databricks, AWS Glue, Azure Data Factory
- **Backend & API Development**: FastAPI, REST APIs, WebSockets, gRPC
- **MLOps & Model Deployment**: MLflow, Docker, Kubernetes, AWS SageMaker, Azure ML, Google Vertex AI
- **Cloud Platforms**: Amazon Web Services, S3, RDS, Athena, Redshift, Google Cloud Platform, Vertex AI, BigQuery, Microsoft Azure, Azure OpenAI, Azure DevOps
- **Databases/Data Warehousing**: PostgreSQL, MongoDB, Amazon Redshift, Google BigQuery
- **Version Control & Collaboration**: Git, GitHub, Bitbucket, GitLab, JIRA, Confluence
- **DevOps & CI/CD**: Jenkins, Azure DevOps, GitLab CI, Terraform, Ansible, GitHub Actions

PROFESSIONAL EXPERIENCE

U.S Bank **Mar 2025 - Present**

AI/ML & GenAI Engineer

- Architected an enterprise RAG platform indexing ~3TB of financial documents and supporting ~12M monthly transactions, using LangChain orchestration, FAISS/Pinecone vector indexing, and OpenAI GPT APIs for grounded response generation.
- Performed data exploration and preprocessing with Pandas and NumPy, engineered and selected features using Scikit-learn pipelines, which improved model accuracy by 8% and cut training time by 15%
- Evaluated OpenAI and open-source transformer embedding models using precision@k, latency benchmarks, and cost-per-query analysis, then selected the model that reduced query latency and lowered production cost, enabling faster and cheaper retrieval
- Implemented cross-encoder reranking using transformer-based models to improve retrieval precision by 15% and reduce hallucinated outputs.
- Conducted domain adaptation experiments using LoRA / PEFT fine-tuning in PyTorch, improving classification accuracy by 9% while reducing GPU cost by ~60% compared to full fine-tuning.
- Built a structured LLM evaluation framework in Python using LangChain and Vertex AI that measured hallucination rate, grounding adherence, and latency, and added guardrails via retrieval confidence thresholds and post-generation validation filters, improving answer reliability and reducing hallucination incidents
- Redesigned fraud detection architecture processing ~5M daily transactions using XGBoost/LightGBM on distributed feature pipelines built with PySpark and BigQuery, improving detection accuracy by 22%.
- Introduced retrieval caching and token optimization strategies, reducing LLM API costs by ~28% without degrading answer quality.
- Built CI/CD ML pipelines with Vertex AI Pipelines, MLflow, and GitHub Actions, shortening deployment time from weeks to under 72 hours
- Deployed containerized microservices using Docker and Kubernetes (GKE) with autoscaling and staged rollout (shadow → canary → full production), achieving 99.9% uptime and reducing incidents by 40%.
- Authored architectural design documentation (ADR/RFC) for enterprise LLM platform, defining retrieval boundaries, scaling strategy, and governance model adopted across AI initiatives.
- Mentored 3–5 engineers on RAG design patterns, fine-tuning workflows, and evaluation methodology, establishing reusable internal GenAI standards.

Health Care Service Corporation **May 2024 - Mar 2025**

AI/ML Engineer

- Owned end-to-end architecture of a PHI-compliant RAG platform that indexed ~1.8M healthcare documents, defining embedding strategy, chunking policy, retrieval configuration, and evaluation benchmarks, which enabled faster, secure document retrieval for clinical queries
- Established LLM evaluation standards (precision@k, grounding accuracy, hallucination rate) and formalized acceptance criteria before production rollout, resulting in consistent model quality checks and early detection of hallucinations
- Led model selection analysis between Azure OpenAI and self-hosted transformer deployments, balancing latency, compliance, cost, and operational complexity, and chose the solution that met latency targets while maintaining regulatory compliance

- Designed and enforced data-governance and redaction pipelines using AWS Glue and GitHub Actions to ensure PHI-safe inference across LLM workflows, which prevented PHI exposure in downstream outputs
- Defined model drift monitoring thresholds and retraining triggers using MLflow and Azure ML registry, reducing production degradation incidents by 40%.
- Coordinated integration with downstream care-management systems, ensuring low-latency inference within operational SLAs.
- Guided DevOps team in containerization and AKS deployment design, enabling controlled rollouts and rollback safety.
- Served as the technical point of contact for AI deployment decisions, streamlining architecture reviews and reducing deployment lead time, which allowed the team to launch new GPT-4 and LoRA/PEFT models faster while maintaining compliance
- Led cross-functional design reviews with compliance, infrastructure, and analytics teams to align LLM deployment with PHI governance and audit requirements.
- Established performance SLAs (latency, grounding accuracy, and drift thresholds) and defined rollback criteria for safe production deployment.

TCS Global **Jan 2021 - Dec 2023**
ML Engineer *Bangalore, India*

- Led architectural redesign of fraud detection pipelines processing over 2 TB of transactional data, defining distributed Spark workflows and data partitioning strategies, which accelerated detection speed and lowered false-positive rates
- Defined an ensemble modeling framework using XGBoost and LightGBM and implemented a threshold calibration strategy aligned with financial risk tolerance models, resulting in more accurate risk assessments
- Established a model evaluation and validation protocol that incorporated cost-sensitive metrics and false-positive impact analysis, enabling more reliable model deployment and compliance with regulatory standards
- Designed a scalable ingestion architecture with AWS Glue and Step Functions to orchestrate multi-source data, reducing ingestion failures and improving pipeline uptime
- Spearheaded migration of model training workflows to SageMaker distributed training, reducing training time by 40%.
- Designed and deployed Dockerized inference services with Python and MLflow, integrating them into enterprise decision engines and Redshift pipelines, which improved inference reliability and streamlined the deployment workflow
- Collaborated with risk and engineering stakeholders to convert fraud detection goals into measurable ML KPIs, using JIRA to track progress, which enabled more accurate fraud monitoring across the organization
- Mentored junior engineers on feature engineering and model optimization using Python and MLflow, leading to faster model iteration and higher code quality within the team

IBM **Jan 2019 - Dec 2020**
ML Engineer

- Designed a scalable ML pipeline architecture with PySpark for enterprise anomaly detection, enabling faster identification of anomalies and reducing processing time
- Defined feature engineering and experimentation strategy improving model accuracy by 12–15%.
- Standardized ML experimentation framework across projects, improving reproducibility and collaboration within engineering teams.
- Contributed to cross-team platform discussions, providing guidance on data integration with AWS Glue to align ML pipeline design with enterprise cloud migration, helping ensure a smooth migration process
- Implemented monitoring and alerting standards using AWS CloudWatch to ensure production reliability.
- Automated infrastructure provisioning using Terraform, standardizing reproducible ML environments across projects.
- Collaborated with platform teams to integrate ML services via gRPC and REST APIs into the enterprise system, enabling real-time predictions and reducing model deployment time

PROJECTS

Intelligent Document Retrieval and Q&A System (RAG-based AI Assistant) **Jul 2025 - Oct 2025**
Texas Tech University *Lubbock, TX*

- Designed and implemented a hybrid RAG architecture combining dense embeddings (FAISS) and lexical search to improve multi-domain retrieval robustness.
- Benchmarked embedding models (OpenAI vs BGE-large vs E5) using precision@k and MRR to evaluate retrieval quality across varied query types.
- Implemented cross-encoder reranking to increase precision@5 by 18% without increasing retrieval depth (K).
- Conducted LoRA fine-tuning experiments in PyTorch (HF + PEFT) to adapt model behavior for domain-specific classification and formatting consistency.
- Built hallucination mitigation layer using retrieval confidence scoring and post-generation validation filters.
- Optimized token usage through dynamic chunk sizing and context window control, reducing inference cost by 25%.
- Deployed microservice-based inference pipeline using FastAPI + Docker + Kubernetes with async streaming responses.
- Implemented evaluation dashboard tracking latency, grounding rate, and response consistency.

Hope Speech Detection Project **Mar 2025 - Jun 2025**

- Designed a multi-class NLP classification system on ~50K labeled social media samples to detect hope speech under noisy, real-world text conditions.

- Addressed severe class imbalance (~1:6 ratio) using weighted cross-entropy and focal loss, improving macro-F1 performance on minority class.
- Benchmarked traditional ML baselines (TF-IDF + Logistic Regression) against deep learning models (BiLSTM, BERT, RoBERTa), selecting transformer fine-tuning for optimal precision-recall balance.
- Fine-tuned transformer models in PyTorch using Hugging Face, optimizing learning rate schedules and batch size to improve macro-F1 by 12% over baseline.
- Performed stratified cross-validation and detailed error analysis to identify misclassification drivers such as sarcasm and contextual ambiguity.
- Calibrated classification thresholds to optimize precision-recall trade-offs depending on deployment scenario and exposed inference through a Dockerized FastAPI service.

EDUCATION

Texas Tech University **Jan 2024 - Dec 2025**

Master of Science (MS), Computer Science

- **Coursework:** Software Testing and Quality Assurance, Data Structures and Algorithms, Automated Software Engineering, Machine Learning

Pragati Engineering College **Jun 2016 - Dec 2020**

Bachelors, Electronics and Communications

CERTIFICATIONS

- Solutions Architect Associate: AWS
- Meta Back-End Developer: Coursera
- Linux Fundamentals LFS101: Linux Foundation
- Python Programming Fundamentals: Coursera
- Test-Driven Development with Python: Coursera
- Python Automation with PyTest and Selenium: Udemy
- SQL for Data Science: Udemy