

# SUJAY ANANTHA

Virginia | (571) 525 6692 | ananthasujay@gmail.com

## EDUCATION

**New York University, Tandon School of Engineering** (September 2022 - May 2024, New York, NY)

M.S. in Computer Engineering, GPA: 3.6, Academic Excellence Scholarship Recipient

**Ramaiah University of Applied Sciences** (August 2017 - May 2021, Bangalore, India)

B.Tech in Electronics and Communication Engineering, GPA: 3.6

## TECHNICAL SKILLS

**Programming & Data Science:** Python, Java, SQL, JavaScript, C/C++, MATLAB, R, Shell, Pandas, NumPy, SciPy, Excel, Alteryx

**Machine Learning & AI:** TensorFlow, PyTorch, Keras, Scikit-learn, Hugging Face, XGBoost, LightGBM, ONNX, MLflow, BERT, CNNs, RNNs, LSTMs, GNNs, NLTK, SpaCy, Gensim

**Big Data & Cloud Platforms:** Spark, Hadoop, Kafka, Databricks, AWS (SageMaker, Glue, Redshift), Azure, GCP (DataProc, BigQuery)

**Databases & Storage:** MySQL, PostgreSQL, MongoDB, DynamoDB, Snowflake, Oracle, Neo4j

**Deployment & DevOps:** Docker, Kubernetes, Jenkins, GitHub Actions, Flask, FastAPI, REST APIs, CI/CD, Git, GitHub, GitLab, Bitbucket

**Visualization & BI:** Tableau, Power BI, Looker, Google Data Studio, Superset, Matplotlib, Seaborn, Plotly

**Core Competencies:** ETL, Data Modeling, MLOps, Statistical Modeling, Forecasting, API Integration, A/B Testing, Agile

## WORK EXPERIENCE

**New York University – VIP: Merger and Acquisition** (June 2023 – May 2025, Brooklyn, NY)

### Machine Learning Engineer

- Designed a GNN benchmarking suite in Python/Scikit-learn to evaluate architecture trade-offs on accuracy and latency, improving model selection efficiency by 20%.
- Accelerated PyTorch node classification training through GPU resource optimization and batch tuning, reducing model training runtime by 35%.
- Constructed scalable ETL workflows with Spark, Pandas, and NumPy to preprocess financial datasets, cutting data pipeline runtime by 50%.
- Developed Tableau and Power BI dashboards to visualize model KPIs and financial time-series patterns, boosting stakeholder visibility into performance metrics by 40%.
- Implemented containerized APIs with Flask, FastAPI, and Docker to serve predictions in real time, enabling rapid iteration and scalable production deployment.

**J. R. Engineering** (January 2021 – July 2022, Bangalore, KA)

### Data Analyst

- Forecasted material demand using ARIMA and LSTM models in Python, reducing procurement stock-outs by 20% and improving planning reliability.
- Automated daily inventory tracking through Python scripts and Excel macros, cutting manual reconciliation efforts by 40% across operations teams.
- Architected a Snowflake-based warehouse schema for procurement and usage history, improving monthly reporting accuracy and efficiency by 25%.
- Illustrated operational KPIs including stock turnover and scrap ratios in Tableau and Power BI, increasing transparency and decision-making for team leads by 40%.
- Engineered anomaly detection pipelines in Scikit-learn to flag irregular SKU movements, reducing audit delays and financial discrepancies by 35%.

## **Raman Research Institute** (June 2019 – December 2019, Bangalore, KA)

### Research Intern

- Programmed RF signal acquisition pipelines using Raspberry Pi and RTL-SDR, enhancing data reliability and stream stability by 15% for experiments.
- Enhanced Doppler detection accuracy through FFT and denoising algorithms in MATLAB, increasing signal-to-noise ratio by 30% for weak signal environments.
- Optimized RF dataset processing with Hadoop and Hive, reducing query latency by 40% and improving downstream workflow efficiency.
- Applied CNN classifiers in PyTorch to identify RF spectrum anomalies, improving anomaly classification accuracy by 25% in experimental datasets.
- Automated SDR validation tests with Bash and GitHub Actions, cutting manual QA time by 50% and ensuring consistent input integrity.

## **Spatez Technology LLP** (May 2018 – December 2018, Thrissur, Kerala)

### Embedded Systems Intern

- Prototyped IoT monitoring systems with Arduino, NodeMCU, and MQTT, shortening prototype turnaround cycles by 15% and enabling real-time telemetry.
- Coded modular C++ sensor loops and anomaly-detection scripts, improving fault detection accuracy by 20% during hardware stress testing.
- Integrated FastAPI dashboards with Azure IoT Hub, enabling real-time telemetry visualization and proactive anomaly detection.
- Containerized firmware simulations using Docker, reducing deployment risk and ensuring isolated edge device testing environments.

## **PROJECTS**

---

### **Financial Security Fraud Detection and AML Model (Scikit-learn, Spark, Hadoop)**

- Engineered fraud detection models using Scikit-learn and Spark to classify 180M+ financial transactions, achieving 95% model accuracy.
- Built real-time Tableau dashboards to surface fraud insights and anomaly patterns, reducing operational triage time by 30%.
- Deployed a full ML pipeline on a simulated Hadoop cluster to demonstrate scalable fraud detection in production-like conditions.

### **Emotional Valence Predictor (Python, TensorFlow, Keras, Pandas, Matplotlib)**

- Designed a deep learning model using TensorFlow with pre-trained word embeddings and dropout regularization to classify text sentiment, boosting accuracy by 15%.
- Optimized model training through batched data loading and memory-efficient NumPy preprocessing to reduce training time by 20%.
- Modularized the ML pipeline into scripts for preprocessing, training, and evaluation to support reproducibility and scaling.

### **Project-SAFE-Bank (Java, Spring Boot, MySQL, Java Script, REST API)**

- Built secure Spring Boot APIs to support ML-driven risk assessment and authentication logic, improving response latency by 40%.
- Normalized transactional schema designs in MySQL to ensure data reliability and optimized querying for ML feature extraction.
- Conducted edge-case testing for fraud scenarios to reduce misclassification rates and improve model-informed system behavior.

### **Fine-tuning LLM's on limited hardware (Hugging Face, PyNVML, WandB)**

- Fine-tuned a 1.5B GPT-2 model using Hugging Face Transformers on constrained GPUs, achieving a 15% improvement in validation perplexity.
- Monitored and optimized GPU memory consumption using PyNVML and logged key training metrics with Weights & Biases for debugging.
- Customized tokenization and dynamic padding strategies to reduce input variability and stabilize learning curves.