Master Work-History / Resume / CV Log for Keerthana Purushotham — Expanded Technical Deep-Dive with Skill Mapping

---

📂 Phase 1: Personal / Professional Profile

- **Name:** Keerthana Purushotham (She/Her)
- **Emails:** keerthanap0808@gmail.com | keep.consult@proton.me | kpurusho@ucsd.edu | kpurusho@ieee.org
- **Phones:** +1 (858) 203-8957 (PST) | +1 (360) 328-1182 (Google Voice)
- **Visa:** H1B FY2026
- **Profiles:** LinkedIn, GitHub, Google Scholar, ResearchGate, Scopus, ORCID, IEEE Author, ACL Anthology, Medium, Substack, Art Portfolio, YouTube.

Summary: Full-stack Software Engineer at AWS with expertise in **Linux security, low-latency distributed systems, and applied ML/NLP**. Blends **deep systems programming, AI-driven automation, and security research** to deliver scalable, correctness-critical software.

📂 Phase 2: Work Experience

Amazon Web Services (AWS) — Software Developer | EC2, Amazon Linux Threat Mgmt. | Seattle, WA

Aug 2022 – Present (3+ yrs)

- **Security & Vulnerability Engineering:**
  - Triaged/remediated **1,300+ Linux & kernel CVEs**, mapping to **Threat & Vulnerability Management, Risk Engineering, Vulnerability Research, CVE Assessment**.
  - Developed exploit reproducibility testing → **Reliability, System Performance, Security Hardening**.
- **Systems & Infrastructure:**

    - Migrated Rust services → Python Lambdas, integrating **Serverless Framework, AWS CloudFormation, CDK**.
    - Engineered nested alarms with CloudWatch → **System Reliability, Load Balancing, Distributed Algorithms**.
    - Built telemetry pipelines → **Analytics, Dashboards, Cloud Development**.
- **Low-Latency & Distributed Workflows:**
    - Designed CVE similarity classifier, SLA breach predictor → **Distributed Systems, Ultra Low Latency, Anomaly Detection, Statistics**.
    - Automated advisories ingestion → **Knowledge Acquisition, Risk Engineering, DevOps Pipelines**.
- **Value:** Demonstrated **multi-domain expertise** in kernel-level security, ML-driven risk modeling, and distributed cloud engineering.

UC San Diego — Graduate Roles | San Diego, CA

2021 – 2022 (1.5 yrs)

- **Research Apprentice (NLP/Finance):**
    - Implemented Transformer sentiment classifiers → **Machine Learning, Statistics, Research Computing**.
    - Combined ML signals with stock-market hedging → **Analytics, Algo Trading, Data Analysis**.
- **Teaching Assistant (Algorithms, Data Structures, Systems Programming):**
    - Created/debugged OS assignments → **Systems Programming, Operating Systems, Scheduling Algorithms, ARM Assembly**.
    - Led review lectures → **Communication, Public Speaking, Theoretical Computer Science**.

BP Logix — Software Engineer Intern | San Diego, CA

Jun 2021 – Sep 2021 (4 mos)

- Migrated **TFVC → Git**, mapping to **Version Control, DevOps, SDLC**.
- Debugged **C# workflow bugs**, validated with stress/load testing → **System Performance, Reliability**.
- Value: **Enterprise migration and resilience analysis**.

Cleo Communications — Software Engineer | Bengaluru, IN

Jul 2018 – Dec 2020 (Internships + Full-time)

- Built RNN-based EDI anomaly detection → **Anomaly Detection, Data Analysis, Python, NLP**.
- Designed pipelines with Logistic Regression, Fuzzy Logic → **Machine Learning, Statistics**.
- Debugged SaaS workflows with Postman → **Cloud Computing, SDLC**.
- Value: **Structured-data ML expertise**.

Samsung R&D India — PRISM Research Intern | Bengaluru, IN

Mar 2019 – Nov 2019 (9 mos)

- Crawlers for conversational data → **Optimized Web Crawling, Low Latency, Distributed Systems**.
- Filtering pipeline (TF-IDF, cosine similarity) → **NLP, Statistics, Scheduling Algorithms**.
- Recognition: **Top-5 project finalist**.

Research Assistant — Dr. Annapurna P. Patil’s Lab | Bengaluru, IN

Sep 2018 – Dec 2020 (2+ yrs)

- Contributions:
  - Autoencoder denoising → **Image Processing, Computer Vision, PyTorch, Keras**.
  - Cloud threat survey → **Computer Security, Risk Engineering**.
  - Conversational dataset filtering → **NLP, Optimized Crawling**.
  - Compiler heuristics → **Theoretical CS, Analytics**.

📂 Phase 3: Education

- **UC San Diego — M.S. Computer Science (GPA 3.83/4.0)** (Dec 2020 – Jun 2022)
  - Courses mapped to: Algorithms → **Algo, Data Structures**; OS → **Systems Programming**; ML/NLP → **AI, Statistics, Deep Learning**.
- **Ramaiah Institute of Technology — B.E. CS (GPA 9/10)** (Aug 2016 – Aug 2020)
  - IoT project with Raspberry Pi → **IoT, Computer Hardware, Automation of Irrigation Systems**.
- **Chethana P.U. College** (2014 – 2016) — IMO Silver Medal.
- **Sophia High School** (2001 – 2014) — Olympiads.

📂 Phase 4: Skills

- **Languages:** Rust, Python, JavaScript, TypeScript, C, C++, C#, Java, Bash, SQL, Ruby, ARM Assembly, YAML.
- **Cloud & Infra:** AWS stack (EC2, S3, API GW, DynamoDB, CloudWatch, IAM, CDK, CloudFormation, Lambda), Kubernetes (ingress, load balancing), Docker, Serverless Framework.
- **Distributed Systems & Performance:** Low Latency, Ultra Low Latency, Distributed Systems, Distributed Algorithms, Network Load Balancing, Load Balancing, Reliability, System Performance, Stress Testing, Scheduling Algorithms.
- **AI/ML & Data:** PyTorch, TensorFlow, Keras, scikit-learn; NLP (Transformers, RNNs, CRFs, BERT, Attention), CV (CNNs, Image Processing, Autoencoders), Statistics, Data Science, Anomaly Detection.
- **Security:** Linux kernel debugging, CVE triage, Threat & Vulnerability Management, Vulnerability Assessment, Vulnerability Research, Risk Engineering, Computer Security.
- **Tools & Frameworks:** Redis, Celery, DevOps pipelines, Git/Version Control, SDLC workflows, Dashboards & Analytics.
- **Systems:** Operating Systems (Linux, Unix, Ubuntu System Measurement), Systems Programming, Computer Architecture, Theoretical CS.
- **IoT & Hardware:** IoT (Arduino, Raspberry Pi), Computer Hardware, Automation of Irrigation Systems.
- **Professional & Research Skills:** Technical Writing, Research Computing, Knowledge Acquisition, Communication, Public Speaking, Leadership, Project Management, Coding Standards, Attention to Detail.

📂 Phase 5: Projects

- **Text-to-Video API Orchestrator (Aug 2025 – Present) — GitHub:**
  - Skills: **Low Latency, Ultra Low Latency, Distributed Systems, Redis, Distributed Algorithms, Load Balancing, Kubernetes, Ingress, Celery, Stress Testing, Computer Vision, Image Processing, Computer Hardware**.

    - Architecture: Rust async workers, Python FastAPI orchestration, Redis messaging, K8s cluster with Prometheus telemetry.
- **Accuracy Is Not Enough (2025) — GitHub:**
    - Skills: **Statistics, Analytics, Machine Learning, Anomaly Detection, Technical Writing**.
- **NER & Parsing (UCSD 2021):**
    - Skills: **PyTorch, CRF, NLP, Statistics, Research Computing**.
- **Ubuntu System Profiling:**
    - Skills: **System Measurement, Stress Testing, Reliability**.
- **IoT Irrigation (2017):**
    - Skills: **IoT, Automation of Irrigation Systems, Raspberry Pi/Arduino, Organic Chemistry integration (soil moisture)**.

📂 Phase 6: Publications

- IEEE/ACL/IJRESM papers mapped to: **Cloud Security, NLP, Image Processing, Compiler Optimization, Anomaly Detection**.
- **Citations (Sept 2025):** 49 | **h-index:** 2 | **i10-index:** 1

📂 Phase 7: Honors, Awards & Tests

- Best Final Year Project (2020), Samsung PRISM Finalist (2019), Olympiad Medals (Math/Science/English).
- GRE 323 | TOEFL 114 | CodeSignal GCA 534/600.

📂 Phase 8: Organizations & Volunteering

- Member: CWE.org, IEEE Reviewer, GHC/AnitaB.org.
- Core Member: Debate Society, Quiz Club, College Magazine.
- Athletics: Football (state/university), track/field.

📂 Phase 9: Project Deep Dives — Niche, Novelty, Challenges, and Value

9.1 Text-to-Video API Orchestrator — Niche & Novelty

[GitHub]

Recap: MVP asynchronous **Text→Video** API around **Genmo Mochi-1**, designed for **Kubernetes** GPU clusters. Supports REST prompt submission, job tracking, and artifact retrieval. **Hybrid Rust + Python**: Rust workers for GPU inference; Python (FastAPI) for orchestration, queues, retries. Success metrics pre-defined (P95 latency, throughput, job success, cluster utilization, API availability). Clear **non-goals** (e.g., RBAC, advanced schedulers) for scoped MVP.

Why it's novel:

- Wraps video generation into a **production-style, horizontally scalable, asynchronous multi-GPU API**, not a single-node demo.
- **Rust+Python split** leverages memory safety + deterministic workers with agile control plane.
- Formal **complexity & correctness thinking** (NP-hard scheduling, idempotency, determinism) uncommon in side projects.
- **Metrics-first design** and explicit non-goals show product and reliability maturity.

Key challenges addressed / flagged:

1. **Scheduling & load balancing (NP-hard):** heuristics, queue policies, preemption, retries, batching.
2. **Concurrency & idempotency:** async workers across nodes without races; deterministic Rust modules.
3. **Prompt variability & VRAM estimation:** workload contours, load prediction.
4. **Reliability:** DLQ, fallback nodes, alarms on long-running jobs.
5. **Latency guarantees:** target **P95 ≤ 10 min** (MVP) across queueing→inference→transfer.
6. **Cost/utilization:** aim **70–90% GPU** without over-provisioning.
7. **Security (future):** tokens, presigned URLs, rate limits, sandboxing.
8. **Cross-language contracts:** schema versioning, backward compatibility.
9. **Model non-determinism & noisy neighbors:** outlier detection, reproducibility checks.
10. **Rollouts & cluster mgmt:** canaries, autoscaling, regionalization, RBAC (future).

How to frame (value statements):

- Built a **production-grade async text-to-video API** on K8s with **multi-GPU orchestration**.
- **Hybrid Rust+Python** architecture for safety, performance, agility.
- Defined & enforced **P95 latency/throughput** targets; designed for failure under **NP-hard constraints**.
- Modular design balancing **latency, cost, correctness**, with hooks for advanced scheduling & auth.

9.2 "Accuracy Is Not Enough — Confusion-Matrix Metrics for CVE Impact Prediction" — Niche & Novelty

[GitHub]

Recap: Technical write-up + visuals arguing **accuracy is insufficient** for cyber-risk modeling. Proposes **Leveled Metrics Framework (L0–L6)**: from counts → Precision/Recall → **FOR/NLR** → **MCC/Youden's J**, with guidance for asymmetric error costs (FN vs FP) in security.

Why it's novel:

- **Domain-specific metric guidance** for **CVE/vulnerability prediction** where FN and FP have very different costs.
- **Hierarchical (L0–L6)** scaffolding improves interpretability and adoption.
- **Actionable recommendations** (lead with Recall/FNR; monitor FOR/NLR; use MCC/J; treat accuracy as sanity check).
- Cheat-sheets & trade-off diagrams translate stats → operational decision-making.

Challenges handled: class imbalance; thresholding; interpretability of advanced metrics; mapping metrics → domain costs; communicating to non-statisticians.

How to frame (value statements):

- Authored a **metric framework** for cyber risk under **class imbalance & asymmetric error**.
- Connected statistical metrics to **real-world security costs** (missed exploits vs alert fatigue).

- Produced **diagrams & cheat sheets** enabling security/ML teams to select appropriate metrics.

9.3 Context-Based Comment Filtering — Niche & Novelty

[GitHub]

Recap: Context-conditioned semantic filter that, given a **seed context** (query/article/prompt), filters **relevant** comments from web-scraped social data using **semantic similarity** (beyond keyword matching). Includes notebook + analysis PDF.

Why it's novel:

- Moves beyond per-comment classification (sentiment/toxicity) to **context-aware relevance**.
- Bridges **IR + NLP embeddings** to reduce noise in social datasets.
- Backed by experiments & error analysis (precision/recall trade-offs).

Key challenges: semantic matching under paraphrase & topic drift; thresholding (FN/FP balance); noisy/imbalanced data; scalability (ANN/ batching); domain robustness; labeled evaluation & error analysis.

How to frame (value statements):

- Built **context-conditioned semantic filters** to surface signal from noisy social data.
- Tuned similarity thresholds; evaluated **precision/recall/ROC**; documented failures & mitigations.
- Reusable for **data curation, conversational systems, IR pipelines**.

9.4 Detecting Pneumonia from Chest X-rays — Niche & Novelty

[GitHub]

Recap: Comparative study of **CNN depth** for medical imaging—**3/4/5-layer CNNs vs VGG16**—to analyze performance, overfitting, and generalization on chest X-rays (pneumonia detection). Multiple Jupyter notebooks document experiments.

Why it's novel:

- Focus on **architecture depth trade-offs** in **low-data medical regimes** (where deeper ≠ always better).
- Analytical lens on variation factors: augmentation, regularization, LR schedules, capacity control.
- Educational clarity via parallel notebooks for reproducibility.

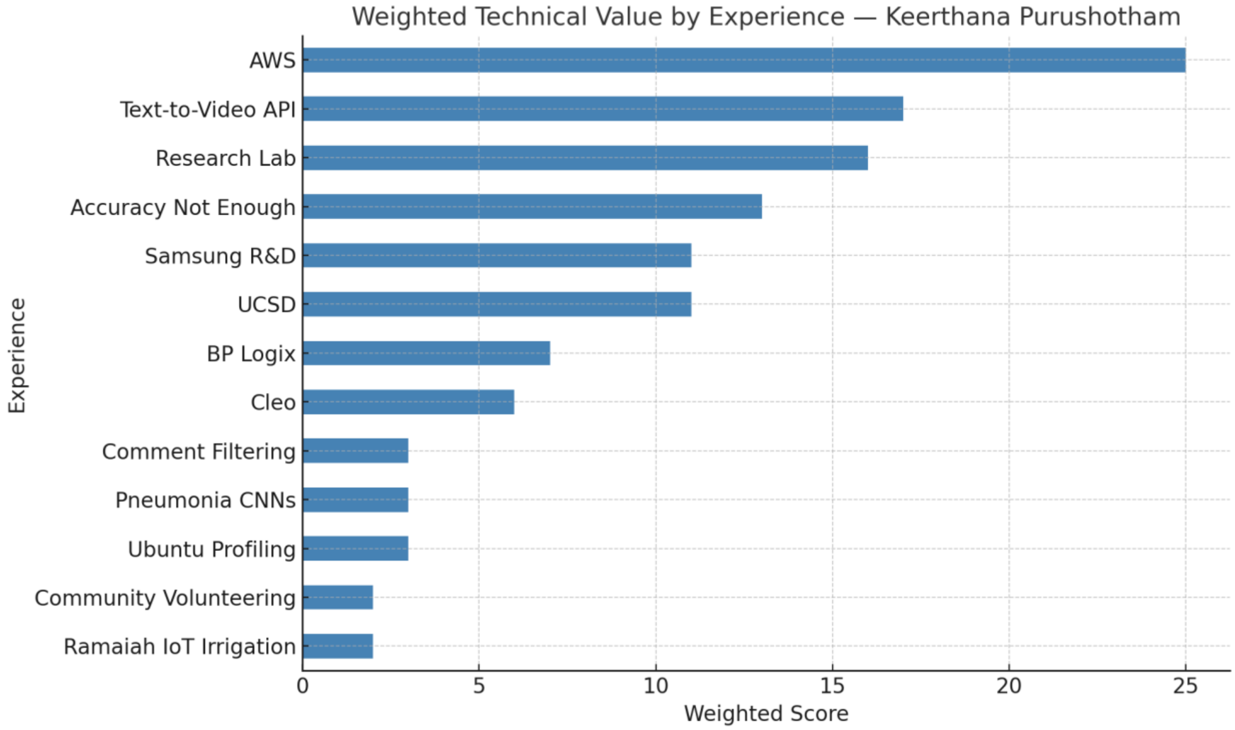
Key challenges: class imbalance & scarcity; fair comparisons (seeds/splits); medical-grade metrics (**sensitivity/specificity/ROC AUC**); preprocessing variance; compute limits.

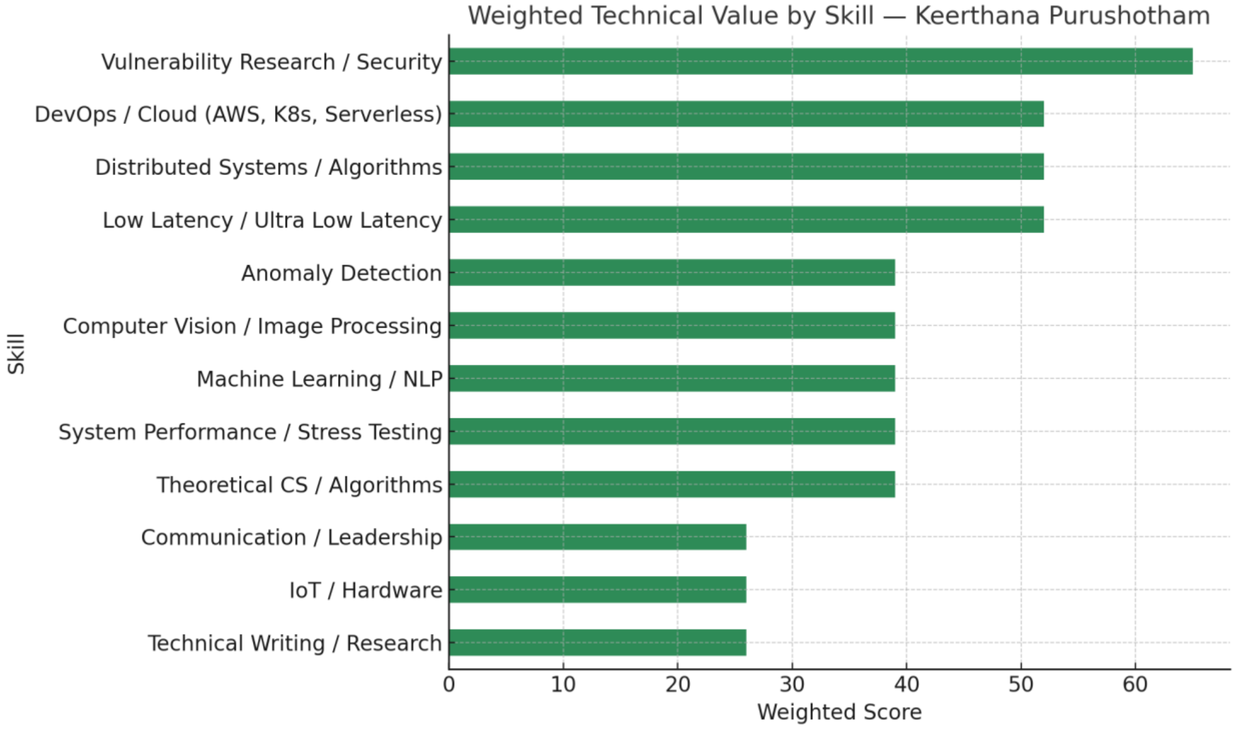
How to frame (value statements):

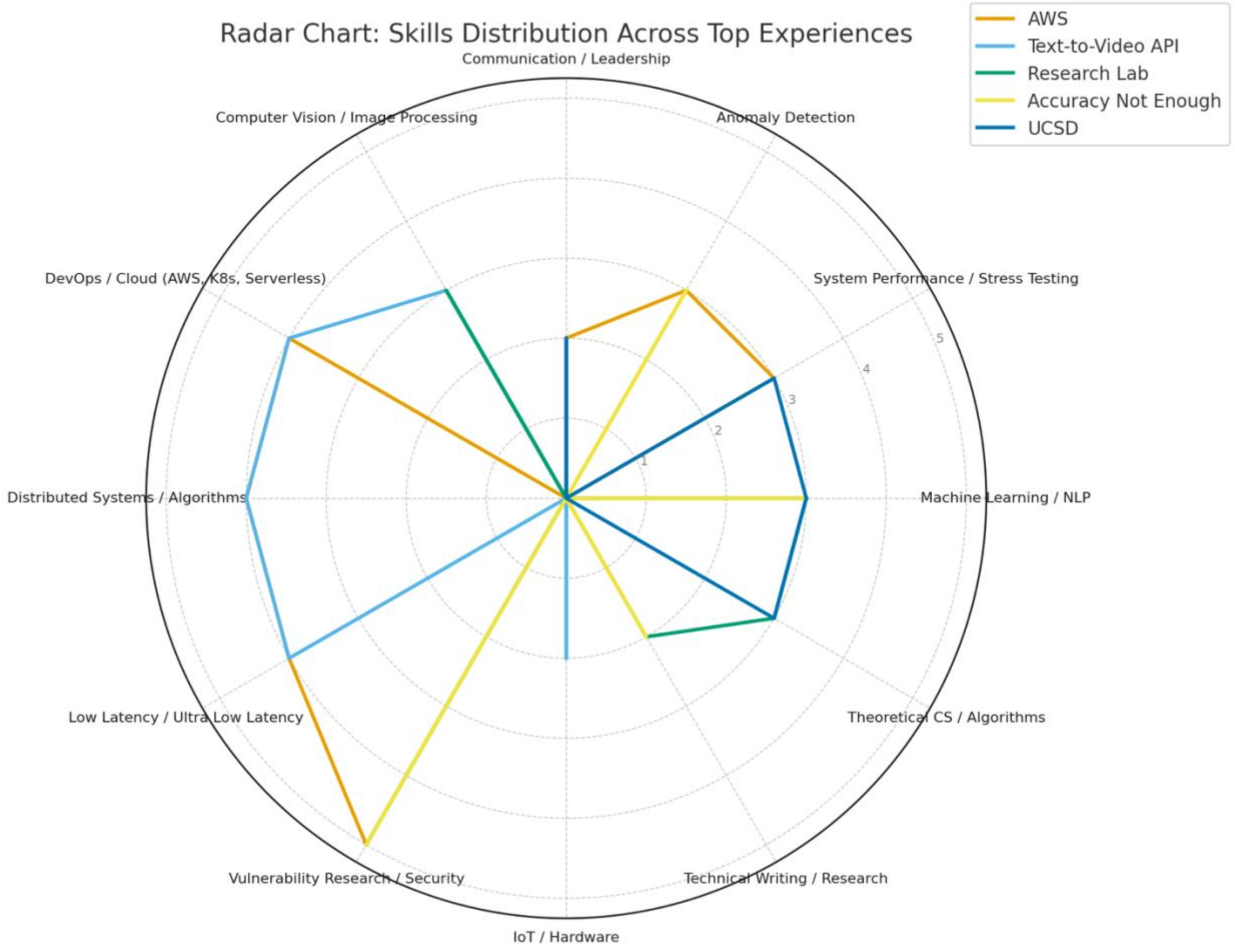
- Designed reproducible **comparative CNN experiments** (shallow vs VGG16) for medical imaging.
- Evaluated **generalization vs overfitting** with domain-appropriate metrics; documented trade-offs.

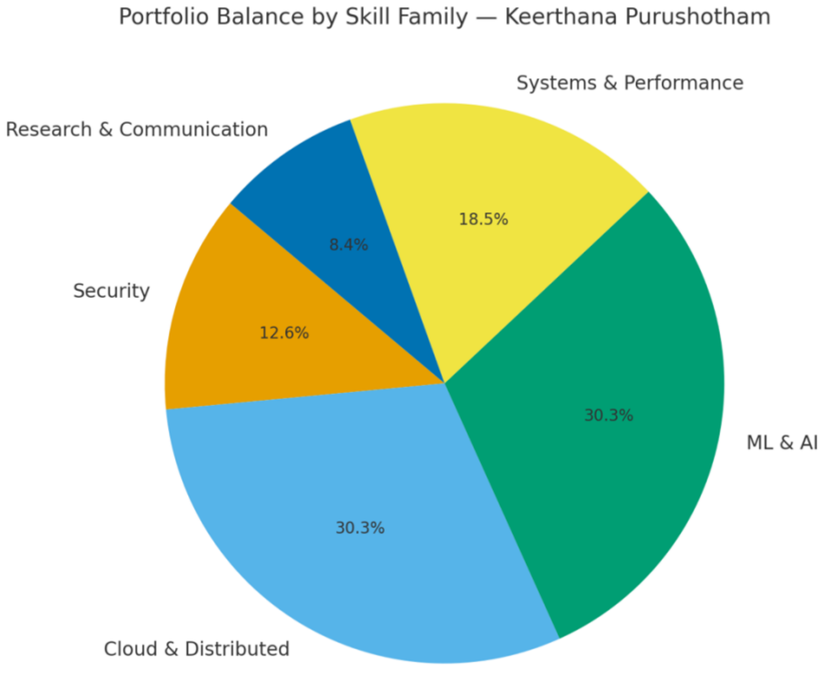
Cross-Project Themes (What this portfolio proves)

- **Systems rigor:** metrics-first design, explicit failure modes, idempotency, and scheduling trade-offs.
- **Security + ML fusion:** applying **statistical rigor** to operational security decisions (CVE, risk).
- **IR/NLP practicality:** context-aware filters for real-world noisy data.
- **Scientific mindset:** controlled comparisons, clear non-goals, and transparent experimentation.


Weighted Technical Value by Experience — Keerthana Purushotham
Experience
AWS
Text-to-Video API
Research Lab
Accuracy Not Enough
Samsung R&D
UCSD
BP Logix
Cleo
Comment Filtering
Pneumonia CNNs
Ubuntu Profiling
Community Volunteering
Ramaiah IoT Irrigation
0
5
10
15
20
25
Weighted Score


Weighted Technical Value by Skill — Keerthana Purushotham
Skill
Vulnerability Research / Security
DevOps / Cloud (AWS, K8s, Serverless)
Distributed Systems / Algorithms
Low Latency / Ultra Low Latency
Anomaly Detection
Computer Vision / Image Processing
Machine Learning / NLP
System Performance / Stress Testing
Theoretical CS / Algorithms
Communication / Leadership
IoT / Hardware
Technical Writing / Research
0
10
20
30
40
50
60
Weighted Score


Radar Chart: Skills Distribution Across Top Experiences
AWS
Text-to-Video API
Research Lab
Accuracy Not Enough
UCSD
Communication / Leadership
Anomaly Detection
System Performance / Stress Testing
Machine Learning / NLP
Theoretical CS / Algorithms
Technical Writing / Research
IoT / Hardware
Vulnerability Research / Security
Low Latency / Ultra Low Latency
Distributed Systems / Algorithms
DevOps / Cloud (AWS, K8s, Serverless)
Computer Vision / Image Processing
1
2
3
4
5


Portfolio Balance by Skill Family — Keerthana Purushotham
Systems & Performance
18.5%
Research & Communication
8.4%
Security
12.6%
ML & AI
30.3%
Cloud & Distributed
30.3%