

Chinmay Arvind

Minneapolis, MN

chinmayarvind23@gmail.com [LinkedIn](#) [GitHub](#)

Education

University of Minnesota - Twin Cities

Expected May 2027

Master of Science in Computer Science

Minneapolis, MN

The University of British Columbia

May 2025

Bachelor of Science in Computer Science (Honors)

BC, Canada

Relevant coursework

The University of British Columbia

Algorithms · Data Structures · Software Engineering · Databases · Parallel Computing · Machine Architecture · Machine Learning · Deep Learning · Artificial Intelligence · Linear Algebra · Probability · Regression Analysis

Experience

Vidasmi

Jun 2026 – Sep 2026

AI Engineer Intern

Remote

- Built an autonomous drone mission platform that converts inspection requests into multi-step missions, combining natural-language planning, real terrain/weather data, PX4 flight execution, and GPT-5 Mini advisory, achieving 100% safe recovery across 200+ fault-injected flights.
- Designed a governed multi-agent LLM and deterministic planning architecture integrating intent extraction, allowlisted strategy generation, route construction, terrain-aware altitude planning, geofencing, real-weather ingestion, aircraft feasibility, energy-reserve analysis, route scoring, bounded refinement, human approval, and preflight authorization, achieving 98% end-to-end workflow success and 96% validation coverage across registered mission constraints while automatically rejecting unsafe missions, reducing drone mission-planning time by 80% (10 → 2 minutes).
- Engineered a production-style mission-execution backend with durable workflows and single-use, revision-bound authorization to block duplicate or unapproved launches, preserve mission state across workflow retries, and retain request-to-flight audit trails for failure investigation and recovery.
- Cut repetitive flights by 60% by implementing LiDAR-based obstacle detection, wind-aware route adaptation, and operator-approved replanning that enables interrupted missions to continue.
- Scaled the control plane for 1,000+ mission requests per day and 50+ concurrent planning workflows, sustaining sub-5-second median mission-generation latency and sub-15-second p95 end-to-end planning latency while executing terrain, geofence, weather, aircraft, and energy validation.

University of Minnesota
Graduate Research Assistant
Minneapolis, MN

Sep 2025 – May 2026

- Built a multimodal financial-fraud research system processing 10+ GB production datasets and evaluating models over 3,200 invoice and payment cases; combined document vision, financial text, transaction features and heterogeneous graphs; achieved 0.871 AUROC, 0.81 AUPRC, and 84% Recall@Top-5%, improving AUPRC by 19% over the strongest unimodal baseline with engineered logistic scoring, supported by three-seed temporal-holdout evaluation, modality ablations and paired bootstrap testing.
- Developed a bounded investigation and robust case-prioritization workflow, comparing parallel document, graph and transaction evidence gathering against a single-worker baseline, with schema-validated read-only tools, checkpointed trajectories, action budgets and deterministic guardrails; achieved 100% evidence-attribution accuracy across 300 checks, 99.6% valid tool calls, rejected 100% of unauthorized tool requests, and captured 13.16% more expected fraud loss versus probability ranking under a fixed analyst-review budget while reducing median investigation time from 14.2 to 6.0 minutes.
- Automated point-in-time feature generation, graph construction, training and evaluation through a reproducible MLOps pipeline scheduling batch workloads on Slurm-managed HPC infrastructure with versioned artifacts, checkpointed retries and model lineage, reducing retraining time from 64 to 21 minutes; implemented drift detection and statistical canary promotion gates, detecting 93% of seeded data/concept-drift events with a 4.2% false-alert rate, configuring daily scheduled runs with checkpointed retries, sustaining 99.4% successful scheduled runs, and rolling back statistically regressing candidates in approximately 0.12 seconds.

University of British Columbia
ML & Backend Engineer
BC, Canada

May 2024 – Apr 2025

- Built an AI-powered educational content platform that converts PDFs and DOCX files into context-aware questions and document-grounded answers, processing 15+ GB of educational content and generating 1,000+ questions/hour; maintained 98% extraction accuracy while reducing manual question creation by 80%.
- Improved the platform's retrieval and generation pipeline through model fine-tuning, document grounding, and asynchronous processing, increasing response-generation speed by 25%, model accuracy by 15%, and question relevance by 30%; productionized the system for 100+ users at 99.1% uptime, reducing API response time by 30% and improving file-retrieval efficiency by 50%.
- Built the end-to-end ML production lifecycle spanning data ingestion and cleaning, embedding generation, model tuning and evaluation, retraining, deployment, fallback, and monitoring; added secure access controls and persistent document/question storage while reducing serving and infrastructure cost by 5% and supporting reliable production inference.

University of British Columbia
Teaching Assistant
BC, Canada

Sep 2024 – Apr 2025

- Capstone Course: Mentored 120+ students across 20+ teams, advising on GitHub workflows, coding standards, and leading team consultations on architecture and feature planning—strengthening collaboration and project quality.
- Introductory Programming Course: Led labs for 30+ students; delivered personalized feedback improving assignment completion rates and student fluency in Java.

Publications & research

DynamicMem: A Long-Horizon Memory Benchmark with Evolving User Context

Jan – Sep 2026

NeurIPS · under review

- Co-authored an LLM evaluation benchmark for long-horizon memory, designing evaluation methodology, analyzing model failure modes, and synthesizing findings to improve reliability in conversational foundation-model systems.

Undergraduate Honors Thesis

Sep 2024 – Apr 2025

PNRL: Reinforcement Learning Development for an Undergraduate AI Course

- Engineered a modular client-server Java-based RL system capable of processing 10,000+ training episodes per session with efficient Q-table updates, enabling students to conduct real-time experimentation of RL strategies.
- Implemented an epsilon-greedy Q-learning policy, increasing final cumulative rewards of the RL agent by 15% within the first 100 training episodes.
- Built a Python-Java bridge via Py4J, enabling 15+ teams to use Python ML libraries in a Java game—cutting development time by 50%.

Achievements

Carlson School of Management: Analytics for Good Hackathon

Jan 2026

2nd Place

- Placed 2nd by delivering an end-to-end analytics prototype and communicating nuanced ideas clearly to non-technical stakeholders, translating technical findings into concise, actionable recommendations.

Origin House Hackathon

Apr 2026

Finalist

Careflow.ai · home-health referral automation

Built a multi-agent referral workflow from discharge-document intake through eligibility, placement, and scheduling.

- **24 hr** hackathon build
- **<1 min** intake → routed · hackathon demo

The challenge

Home-health referrals arrive as unstructured discharge documents and require several manual checks before care can begin.

The approach

Careflow.ai combines extraction and validation with browser agents for eligibility, placement, and scheduling, keeping the workflow visible and traceable.

The result

Reached the final at the Origin House Hackathon 2026 with an end-to-end referral demo. The under-one-minute result describes the hackathon demo, not a production clinical deployment.

Projects

Permission-Aware Commercial Lending RAG & Underwriting Evaluation Platform

Sep 2026

Python · FastAPI · LlamaIndex · Weaviate · Snowflake Cortex · DeepEval

- Built a permission-aware lending RAG system across 3,800+ pages of synthetic policy and borrower documents using hybrid retrieval, semantic chunking, selective query rewriting, reranking, and page-level citation enforcement, achieving 86.54% Recall@10 and 0.788 nDCG@10, a 12.49 percentage-point improvement in retrieval recall.
- Created a 240-question adversarial evaluation set spanning policy interpretation, borrower evidence, missing documents, contradictions, restricted data, and underwriting exceptions, achieving a 92.5% grounded-answer pass rate while reducing unsupported claims to 1.7%.
- Implemented response caching, policy-aware access controls, end-to-end retrieval tracing, and CI quality, latency, and cost gates, improving answer quality by 13.4 percentage points, measuring 300 ms p95 latency at 8 concurrent clients across two local API processes with warm caching, and reducing average cost per request by 31% to \$0.043.

Distributed GPU Inference Serving & Automated Model Release Platform

Sep 2026

PyTorch · Ray Serve · vLLM · Redis · MLflow · Airflow

- Built a distributed text and multimodal inference platform with GPU-aware routing, continuous batching, prefix caching, and speculative-decoding evaluation, increasing sustained throughput from 13.0 to 34.6 requests/s, token throughput by 2.64×, and average GPU utilization from 29% to 58%.
- Benchmarked 6,000+ inference requests across baseline and optimized serving configurations; in a separate quantized 3B release benchmark, reduced median client TTFT from 247 ms to 116 ms and p95 latency from 2.24 s to 0.86 s, achieving 100% correctness on a 32-case release regression suite.
- Automated model evaluation, deployment, and rollback orchestration with CI performance gates, an 18-task Airflow pipeline, and MLflow evidence tracking, completing 512/512 release benchmark requests and validating 60 post-deployment probes and 1→2→1 GPU engine scaling.

Agentic Kubernetes Payment-Incident Investigation & Remediation System

Sep 2026

Python · LangGraph · Kubernetes · Prometheus · Elasticsearch · OpenTelemetry

- Built an agentic incident-response system correlating Kubernetes health, logs, traces, deployment changes, and payment telemetry across 24 reproducible failure scenarios, achieving 91.7% root-cause Recall@1 and Recall@3 in predicate-assisted development replay.
- Developed a failure-oriented evaluation harness covering OOMKills, bad rollouts, scheduler saturation, dependency outages, misleading telemetry, and payment-slice degradations, rejecting 100% of 120 unauthorized remediation attempts in component evaluations.
- Added structured investigation traces, incident memory, deterministic approval gates, GraphQL, Slack notifications, and durable Pub/Sub processing, achieving 8.41-second median diagnostic replay latency, reducing median investigation time from 11.8 min to 2.9 min, and maintaining \$0.07 average LLM cost per incident.

High-Throughput Payment Webhook Reliability & Risk-Scoring Gateway

Go · Rust · Redis Streams · Kubernetes · Prometheus · scikit-learn

- Sustained 12,249 payment events/sec at 46.46 ms p99 acknowledgement latency in a 60-second local benchmark using atomic deduplication, enqueueing, and asynchronous processing; all 735,006 requests were accepted.
- Produced zero duplicate risk-decision records across 1M+ replayed webhook deliveries using shared event IDs and atomic decision publication, with concurrent-retry and worker-restart verification.
- Achieved 0.95 F1 versus a 0.87 rules baseline on 2,000 held-out transactions, with 196 ns p99 in-process scoring, using a calibrated lightweight classifier and reproducible evaluation pipeline.

Risk-Adjusted Hallucination Detection

Feb – Apr 2026

Python · Qwen · Calibration · Self-consistency · Abstention

- Built a calibrated hallucination-risk detector using 4 uncertainty/evidence signals: token uncertainty, self-consistency disagreement, semantic entropy, and groundedness consistency.
- Engineered an end-to-end ML pipeline for answer generation, feature extraction, feature-table construction, detector training, calibration, abstention thresholding, frozen model bundles, and transfer evaluation.
- Evaluated on 5.3K grounded QA examples across 4K PHANTOM and 1.3K WikiQA samples using Qwen3-8B generation, Qwen3-14B judging, and k=5 self-consistency sampling.
- Achieved 77.3% AUROC, 64.5% AUPRC, 75.9% accuracy, and 63.7% F1 on PHANTOM hallucination-risk detection.
- Improved PHANTOM selective accuracy from 68.3% to 78.6% by abstaining on the riskiest 18.2% of answers while preserving 81.9% coverage.
- Compared Platt scaling, isotonic regression, temperature scaling, and raw probabilities, reaching calibrated test ECE of 0.0734 on PHANTOM and 0.0725 on WikiQA.
- Validated robustness limits through bidirectional transfer experiments, PHANTOM→WikiQA and WikiQA→PHANTOM, diagnosing feature-distribution shift, calibration instability, and threshold non-portability.

Agentic Quantitative Research Orchestration & Backtesting Platform

Sep 2026

Python · LangGraph · Polars · MLflow · Neo4j · PyTorch

- Built an autonomous quantitative-research system that turned natural-language investment ideas into relevant literature, structured hypotheses, hybrid factor strategies, sandboxed backtests, and iterative research decisions, backed by a deterministic execution engine exercised across 45 historical signal-and-cost cases covering 30,000+ historical data records.
- Developed a validation pipeline with multi-paper synthesis, walk-forward and purged validation splits, transaction-cost modeling, and statistical robustness checks to evaluate published-factor reproduction and factor-spec extraction accuracy across 15 price-based signals across 3 cost settings achieving 100% agreement across 40+ portfolio valuations and 15+ trade fills—including fees against independent LEAN execution of a reference backtest.
- Recorded successful and failed experiments as reusable research memory and training trajectories, preserving 100% of prompts, datasets, code, backtests, and experiment lineage, verifying linked research artifacts, tracing 1,200+ agent/tool spans, and reducing average research runtime from 46 min to 18 min at \$0.84 average LLM cost per research run.

Jersey Number Recognition Boosted

Mar – May 2025

PyTorch · OpenCV · FlashAttention · ViTPose · Temporal smoothing

- Built an end-to-end computer vision pipeline to extract jersey numbers from sports video, combining person tracking, pose-guided ROI cropping, and OCR/scene-text recognition to produce per-player number predictions.
- Implemented pose-detection pipeline improvements by adding temporal smoothing for keypoints and enabled GPU reduced-precision inference to improve stability and performance.
- Hardened pipeline execution by improving device selection (CPU/GPU), fixing keypoint serialization for JSON outputs, tuning pose-based cropping thresholds, for reduced latency and reproducible runs.

EEG Sleep Deprivation Classification

PyTorch · EEG · CNNs · ViTs · GRU/LSTM

- Built an end-to-end EEG deep learning framework on 8,300 resting-state EEG epochs to classify normal sleep vs. sleep deprivation and predict session-level SSS/KSS sleepiness scores.
- Designed leakage-safe 70/15/15 subject-wise splits with 5,738 train, 1,246 validation, and 1,316 test samples, ensuring no subject overlap across train, validation, and test sets.
- Demonstrated transfer of learned EEG representations to downstream sleepiness prediction, achieving 0.692 Pearson and 0.738 Spearman correlation on held-out subjects using feature-based KSS modeling with leakage-safe subject-wise splits.
- Benchmarked 4 EEG representation learners—Residual EEG CNN, 2-Branch EEG CNN, DeiT-Tiny, and ViT-Small—and reused the best encoder for downstream sleepiness modeling.
- Implemented 5-fold subject-wise cross-validation to evaluate generalization across held-out subjects and prevent identity leakage in EEG experiments.
- Built downstream prediction pipelines using pretrained EEG embeddings with Ridge/MLP/GRU/LSTM models, direct ordinal prediction, and multi-task learning for joint classification and sleepiness estimation.
- Improved training robustness with AdamW, ReduceLROnPlateau, early stopping, dropout, weight decay, label smoothing, class-weighted loss, weighted sampling, time jitter, and random time-crop augmentation.

Drift.ai

Apr 2026

Next.js · FastAPI · Supabase · Ollama · Qwen

- Built a full-stack AI personal finance platform that replaced a fragmented 7-step manual analysis workflow with a unified product flow across 2 transaction ingestion paths (CSV + Plaid), automated drift detection, what-if simulation, and PDF/email report delivery.
- Engineered a hybrid deterministic + LLM pipeline that combined BLS CPI-U inflation-adjusted spending analysis with local Qwen via Ollama to run multi-turn behavioral interviews, classify spending into 6 behavior tags, and generate personalized 30-day recovery plans backed by 8 cited financial evidence fields.
- Implemented a production-style, privacy-first architecture with Auth0, Stripe Checkout, Resend, Vercel, and a 4-service Dockerized stack; kept raw transaction rows local while persisting only 6 summary data entities in Supabase, enabling monetized report delivery with minimal sensitive-data retention.

CleanBreak

TypeScript · Next.js · Playwright · SQLite · Solari Desktop

- Built an autonomous subscription-cancellation platform that reduced average cancellation time from 8 minutes to 45 seconds, a 91% reduction, automating provider navigation, retention-offer handling, guarded execution, and independent billing verification.
- Helped 10 pilot users avoid \$2,400 in annual subscription renewals by converting unwanted recurring expenses into independently verified cancellation outcomes with durable receipts.
- Reduced cancellation-related manual follow-up by 80% through automated billing-state verification and auditable receipts confirming that future renewal charges had stopped.

Disagreement-Aware RAG

Aug 2025

FastAPI · LlamaIndex · Next.js · Hugging Face Transformers

- Built a disagreement-aware RAG system that abstains when predicted disagreement risk is high.
- Trained a lightweight logistic regression model over self-consistency variance (k-resampled generations), evidence overlap (ROUGE-L-style), and an token-level entropy
- Engineered a full-stack app that allows querying of questions on a test dataset and plots coverage vs. hallucination; achieved ~97% coverage @ 3–4% hallucination ($\tau \approx 0.55$ – 0.60).
- Implemented a zero-shot entailment auditor with BART-MNLI to flag likely hallucinations in evaluation.

Alvaluate

Jun – Aug 2024

JavaScript · Node.js · Express.js · React.js · Docker

- Led a team in developing a course platform integrating OpenAI's Assistants API to automate grading and feedback for student submissions—reducing grading time by 70% and increasing grading consistency.
- Implemented a microservices architecture with robust user authentication and role-based access control to ensure modularity, fault isolation, and streamlined continuous integration and deployment.
- Leveraged test-driven development (TDD) across services, increasing test coverage to 90% and reducing post-deployment bugs by 40%.

Algorithmic Trading System

Jan – Feb 2025

FastAPI · PyTorch · scikit-learn · Pandas · NumPy

- Developed a stock trading simulation platform combining market trends, news sentiment, and ML models to mimic real-world trading—enabling informed, data-driven strategies.
- Constructed a hybrid model using K-Means, Agglomerative Clustering, and logistic regression to classify market regimes—improving detection accuracy by 34%, enabling faster strategy shifts and boosting simulated returns.
- Designed a CNN-LSTM model with reinforcement-based position sizing, increasing prediction accuracy by 12% and boosting simulated returns by 30% during backtesting.
- Built a sentiment-analysis pipeline from RSS feeds to generate real-time trading signals and enhanced predictive insight.

Bank Account Fraud Detection

Network science · Fraud detection · Graph analysis

- Builds similarity graphs from bank-account application records to investigate structural patterns associated with fraud.
- Uses centrality, clustering, connected components, and graph cores to compare network structure, with reproducible analysis and visualizations in R and Quarto.

Technical skills

Languages: Python, TypeScript, Java, C, R, JavaScript, SQL, HTML, CSS

LLMs, agents & retrieval: LangGraph, LangChain, LlamaIndex, Hugging Face Transformers, Sentence Transformers, Ollama, OpenAI API, Snowflake Cortex Search, NLTK, OpenSearch, Elasticsearch, FAISS, Weaviate, BM25, Hybrid Retrieval, Reciprocal Rank Fusion, Semantic Chunking, Query Grounding, Cross-Encoder Reranking

Machine learning: PyTorch, scikit-learn, XGBoost, NumPy, Pandas, SciPy, Polars, OpenCV, CUDA, Matplotlib, Seaborn, PyTorch Geometric, BERT, Vision Transformers (ViT), GraphSAGE, Multimodal Learning, Late Fusion, Probability Calibration

Backend & APIs: FastAPI, Flask, Node.js, Express.js, Uvicorn, Pydantic, SQLAlchemy, HTTPX, REST APIs, GraphQL, gRPC, Protocol Buffers

MLOps & observability: Apache Airflow, MLflow, Artifact Versioning, Checkpointed Pipelines, Statistical Canary Gates, Drift Detection, DeepEval, RAGAS, OpenTelemetry, Prometheus, Grafana, LangSmith, Temporal Validation, Modality Ablation, Paired Bootstrap Testing

Databases & infrastructure: PostgreSQL, MySQL, SQLite, Redis, Supabase, Docker, Docker Compose, Kubernetes, Terraform, AWS ECS/Fargate, ECR, ALB, CloudFront, CloudWatch, IAM, SQS-Compatible Messaging, Hugging Face Spaces, Hugging Face Hub, Render, AWS CloudFormation, Slurm, High-Performance Computing (HPC)

Data, quantitative systems & tooling: LEAN, PySpark, Apache Spark, PyArrow, Parquet, ReportLab, Pillow, Bun, D3.js, pytest, pytest-cov, Ruff, mypy, Git, GitHub Actions, DroneCI, uv, JUnit