

Aditya Shrestha

Bhaktapur, Nepal | adityashrestha39@gmail.com | +977-9840414854
github.com/AdiShrestha | linkedin.com/in/adityashrestha39

Research Interests

Systems for machine learning (adaptive stream processing, backpressure-driven control).

Education

Kathmandu University

B.E. in Computer Engineering

Dhulikhel, Nepal

Dec. 2022 – Expected Feb. 2027

- Cumulative GPA: 3.91 / 4.00 (through six semesters, Year I–III)
- Semester GPAs: 3.78, 3.81, 3.95, 4.00, 4.00, 3.95 (two semesters at 4.00/4.00)
- Electives: Neural Networks and Deep Learning; Computer Vision
- Relevant coursework: Artificial Intelligence, Digital Signal Processing, Operating Systems, Computer Architecture & Organization, Algorithms & Complexity, Database Management Systems, Communication & Networking, Principles of Programming Languages, Compiler Design, Software Engineering, Numerical Methods, Statistics and Probability, Discrete Structures

Kathmandu Model Higher Secondary School

10+2 (Higher Secondary), Science

Baghbazar, Kathmandu, Nepal

Completed 2022

- CGPA: 3.72 / 4.00

Honors and Awards

- **Winner, Global IME AI/ML Hackathon 2026** — Team Eigen Threat, for *BankSentinel*, a five-agent AI intrusion detection system for Nepali banking networks (June 2026)
- **Academic Excellence Award** — Kathmandu Model Higher Secondary School (2022)

Certifications

- **AWS Certified Solutions Architect — Associate (SAA-C03)** — Amazon Web Services (Sept. 2026 – Sept. 2029) [[Credly Badge](#)]
- **AWS Academy Graduate — Cloud Foundations** — AWS Academy [[Credly Badge](#)]

Research Experience

TopoRing — Higher-Order Topological Learning for

Collusion Detection

Single-author research preprint

Aug. 2026

Python, PyTorch Geometric, TopoNetX, scikit-learn

- Benchmarked 8 model architectures (CCNN/CWN, SCNN/MPSN, GINE, GNN baselines, Tabular) on 5.08M transaction records from the IBM AMLworld benchmark to classify candidate money-laundering cycles
- Implemented algebraic cell and simplicial-complex lifters enforcing boundary-of-boundary invariants ($\mathbf{B}_1\mathbf{B}_2 = \mathbf{0}$) and 2-cell message passing over arbitrary cycle topologies ($k \in [3, 12]$)
- Designed a leak-free 5-fold outer / 3-fold inner cross-validation protocol over 18 protected account-connected groups, evaluating 200 model checkpoints across 5 fixed random seeds
- Executed group-blocked model-swap permutation tests (10,000 permutations) with Holm multiplicity correction and 10,000-sample group-bootstrap 95% CIs to quantify evidence for higher-order topological inductive biases

Backpressure-Adaptive Stream Processing Systems

(KLStream / BPFeat)

Independent systems research, C++

2026 – Present

- Designed and implemented a from-scratch, Kafka-less, single-node stream processing runtime with lock-free SPSC/MPMC queues, cooperative multi-threaded scheduling, bounded-memory execution, and a modular operator library

- Developed an adaptive-windowing control mechanism that dynamically resizes ML-inference batch sizes using an exponential moving average of downstream queue occupancy, directly analogous to TCP additive-increase/multiplicative-decrease congestion control
- Validated a from-scratch C++ Isolation Forest implementation against scikit-learn (Spearman rank correlation $\rho = 0.947$ on held-out synthetic financial tick data)
- Conducted a statistically rigorous evaluation (30 paired runs per configuration, Wilcoxon signed-rank testing, point-adjustment-aware F1 protocol following Kim et al., AAAI 2022) demonstrating a $2.4\times$ reduction in P95 detection latency at a measured accuracy cost under bursty synthetic load
- Extended the mechanism (as “AMSC”) to adaptive Matryoshka-embedding truncation for vector search under backpressure; identified and documented a critical measurement bug involving masked baseline latency through end-to-end validation before drawing conclusions
- Ongoing: core implementation and evaluation checkpoints completed; no manuscript has been submitted or published

Load-Adaptive IIR Filtering for Financial Anomaly Detection

July 2026

Course-affiliated DSP research project (COMP 407, Kathmandu University)

- Designed a first-order IIR filter with pole location driven by an exogenous system-load signal rather than the price signal itself, departing from conventional signal-driven adaptive filters
- Provided a closed-form Z-domain stability proof, DC group-delay formula, and slew-rate bound for the adaptation mechanism
- Ran a 150-seed, three-regime, five-asset statistical evaluation (paired t -test) on real Binance BTCUSDT/ETHUSDT/etc. tick data; reported a statistically null accuracy result ($\Delta\text{AUC} = +0.0003$, $p = 0.67$)
- Quantified a complementary load-shedding mechanism’s 76.1% throughput gain at a $\sim 10\%$ ROC-AUC cost, characterizing the resulting Pareto frontier

Selected Projects

BankSentinel — Five-Agent AI Intrusion Detection System

June 2026

Hackathon project, Global IME AI/ML Hackathon 2026 — 1st

Place

- Designed a five-agent pipeline (Packet, Flow, Behavior, Correlation, Response) combining a Random Forest beacon classifier, six regime-aware Isolation Forest models, and a BiLSTM autoencoder trained exclusively on normal behavior for zero-day detection
- Reduced false-positive rate from 22.7% to 2.1% during high-volume traffic windows and cut raw alert volume by 87% via a four-layer suppression engine, validated on CICIDS-2017 and CTU-13

CoreMesh — Topology-Aware NUMA Scheduler for ARM64

Jan. 2026

Group systems project, C++17/ARM64

- Designed a consistent-hashing task-to-core scheduler with lock-free per-core ring-buffer communication and four-tier NUMA-aware work-stealing for many-core ARM64 systems
- Benchmarked against Linux CFS across 13 configurations (1K–2M tasks): $1.08\times$ average / $1.37\times$ peak speedup, with 100% completion versus CFS’s 92.3% at 2M-task stress scale

Technical Skills

- **Languages:** C++17/20, Python, C, JavaScript/TypeScript, SQL
- **Cloud & Infrastructure:** AWS (VPC, EC2, S3, RDS, Lambda, IAM, CloudWatch), Docker, Linux
- **Systems:** Lock-free concurrent data structures, stream processing runtimes, NUMA-aware scheduling, ARM64
- **ML/Data:** PyTorch, PyTorch Geometric, scikit-learn, NumPy, SciPy, Pandas, Isolation Forest, BiLSTM autoencoders
- **Tools:** Git, CMake, TopoNetX, PostgreSQL, SQLite, FastAPI, React

Languages

Nepali (native), English (fluent), Hindi (fluent), Newari (heritage language — strong comprehension, conversational)