# Machine Learning for Real-Time Network Failure Detection and Localization

CUNY School of Professional Studies

Michael Hernandez

IS 499 Information Systems Capstone

Professor John Bouma

September 12, 2025

# 1 Topic Description

This project builds a practical system that processes live BGP updates and device logs, using accessible machine learning to quickly detect failure-induced incidents and pinpoint likely origins (e.g., top-of-rack switch, spine/route reflector, or edge/provider device). The focus is operator value: fewer false alerts, faster action, and clear explanations of what broke and where.

# 2 Problem Description

Large BGP-routed environments generate many alarms but little guidance on what matters or where to start. SNMP and syslog thresholds flag hard failures but often over-page on benign local events and under-explain control-plane or egress faults (Mohammed, Mohammed, Côté, & Shirmohammadi, 2021). Engineers then manually sift through logs and correlate devices, increasing detection and resolution times. In large-scale networks, this delay impacts availability and on-call workload. A system that links control-plane churn with structured log patterns and understands network role topology can cut detection time, suggest likely fault locations, and suppress noise isolated to a rack or host.

# 3 Solution Approach

The system extracts features from BGP updates (withdrawals, AS-path churn, next-hop shifts) and device logs (template counts, severity, burstiness). It applies a time-series anomaly detector to BGP data (Scott, Johnstone, Szewczyk, & Richardson, 2024) and an unsupervised model to per-device log vectors (Cheng, Li, Lv, Liu, & Wang, 2021). Scores are normalized, fused, and analyzed by a topology-aware localizer using a role map (server, ToR, spine/RR, edge) to identify origins and down-rank edge-local flaps (Tan, Huang, You, Su, & Lu, 2024). A dashboard displays each alert with the suspected location and top signals, aiming for accuracy and clarity without heavy labeling or complex models.

# 4 Coursework Foundations

My coursework provided the foundation, while industry experience added domain depth. In Python, data structures, and databases (IS 210/211, IS 361, IS 362), I learned to build parsers, design queryable schemas, and process streaming records. Networks and infrastructure (IS 205, IS 260) gave me context to define BGP failure modes and set an SNMP/syslog baseline. Systems analysis, enterprise architectures, and project management (IS 320, IS 300, PROM 210) shaped my layered design (ingest $\rightarrow$ features $\rightarrow$ models $\rightarrow$ localization $\rightarrow$ UI), requirements, UMLs, and semester plan. Security and strategy (IS 250, IS 350) guided my use of lab-based data, secure telemetry handling, and clear communication of operator value. Where the degree built general skills, my work experience provided operational specifics like BGP-routed fabrics, anycast, VXLAN, and incident response which makes the project feasible and relevant.

# 5 Development & Evaluation Plan

Implementation will be in Python, using libraries for streaming feature extraction and unsupervised detection, with a containerized virtual lab generating realistic BGP updates and logs. The

lab includes two spines, two top-of-rack switches, two edges, and multiple "server" peers. I will inject failures—one-way signal loss, a route-reflector crash, an edge/provider outage, and a server crash—and compare alerts against a baseline of manual SNMP/syslog triage. The repository contains the dashboard, ingestion scripts, experiment notebooks, diagrams, and a LaTeX/BibTeX setup for citation and PDF export.

### 5.1 Evaluation Metrics

To show practical value, I will report three measures against the baseline. F1 reflects alert quality by balancing real failures caught (recall) with false alarms avoided (precision). Detection delay is the time from failure onset to the first alert, given as a median with interquartile range, and compared with the baseline's delay for the same runs. Hit@k measures localization accuracy by checking if the true failure origin is among the top-k suspects (e.g., Hit@1, Hit@3). These metrics capture fewer false alerts, quicker detection, and clearer first actions.

## 6 Deliverables & Timeline

The final submission will include a paper, a presentation, a live demo against the virtual lab, and a public repository with code, documentation, and weekly updates. Early work centers on stabilizing the lab and data plumbing, then moves to feature extraction, unsupervised scoring, topology-aware localization, and the dashboard. The final phase focuses on experiments, ablations, results packaging, and presentation polish. The scope is sized for a semester, aiming for measurable outcomes and a working proof of concept.

## 7 Writing & Formatting

The paper will use plain, professional language and avoid jargon where possible, defining terms when needed (e.g., "BGP updates" as "routing change messages"). In-text APA citations will be included with a reference list.

## References


Cheng, M., Li, Q., Lv, J., Liu, W., & Wang, J. (2021). Multi-scale lstm model for bgp anomaly classification. *IEEE Transactions on Services Computing*, *14*(3), 765–778.

Mohammed, S. A., Mohammed, A. R., Côté, D., & Shirmohammadi, S. (2021). A machine-learning-based action recommender for network operation centers. *IEEE Transactions on Network and Service Management*, *18*(3), 2702–2713.

Scott, B., Johnstone, M. N., Szewczyk, P., & Richardson, S. (2024). Matrix profile data mining for bgp anomaly detection. *Computer Networks*, *242*, 110257.

Tan, Y., Huang, W., You, Y., Su, S., & Lu, H. (2024). Recognizing bgp communities based on graph neural network. *IEEE Network*, *38*(6), 232–238.