Machine Learning for Network Anomaly and Failure Detection

CUNY School of Professional Studies

Michael Hernandez

IS 499 Information Systems Capstone

Professor John Bouma

October 11, 2025

Contents

1 Introduction

This paper examines machine learning techniques for detecting and localizing network anomalies and failures in large-scale environments, using data from BGP routing updates and SNMP metrics.

Traditional network monitoring relies on threshold-based alerts from SNMP, often producing many false positives and offering little context for locating failures (Wang, 2020; Manna & Alkasassbeh, 2019). Recent research suggests that machine learning approaches applied to SNMP datasets may improve anomaly detection accuracy in operational environments, with supervised and unsupervised methods showing promise for identifying specific failure patterns (Manna & Alkasassbeh, 2019). This project explores whether combining streaming telemetry from multiple sources using unsupervised learning techniques can provide complementary detection capabilities for network operations.

The system integrates BGP monitoring and SNMP metrics for network anomaly detection. Using unsupervised learning techniques such as Matrix Profile analysis (Mueen & Keogh, 2017; Scott et al., 2024) and Isolation Forest (Liu et al., 2008), the approach aims to reduce alert noise while providing failure localization capabilities. The evaluation examines whether this multi-modal architecture offers practical improvements over single-source monitoring in controlled test scenarios.

2 Topic Description

2.1 In-depth Description of the Chosen Topic

Large networks face a fundamental challenge: when something breaks, operators in network operations centers must quickly determine what failed and where (Mohammed et al., 2021). A network might contain thousands of interconnected devices, and failures can cascade from one device to many others, making the root cause difficult to identify. This project addresses this challenge by using machine learning to automatically detect network problems and pinpoint their source.

Networks continuously generate health information through two primary monitoring systems. The Border Gateway Protocol is the routing system that directs traffic across the Internet by allowing networks to advertise which destinations they can reach (Rekhter et al., 2006). When network conditions change due to equipment failures, cable breaks, or configuration errors, routers send update messages to inform their neighbors about these changes. These routing updates create a continuous stream of information about how traffic flows through the network. The Simple Network Management Protocol provides a different view of network health by collecting hardware performance data from individual devices through trap-directed notification, where managed devices send unsolicited messages to inform network management systems of significant events (Cisco, 2006). This monitoring protocol reports metrics such as processor utilization, memory consumption, temperature readings, and interface error counts. Together, these two information sources provide complementary views of network operations: routing updates show how traffic paths change over time, while hardware metrics reveal the physical condition of network equipment.

Under normal conditions, these monitoring systems generate large volumes of data with characteristic statistical properties. Routing updates occur at varying rates as networks make routine adjustments, and hardware metrics fluctuate within typical operating ranges. While individual measurements vary considerably due to traffic patterns, environmental conditions, and workload changes, the overall statistical behavior remains within bounds established during normal operation. When failures occur, these statistical properties change in ways that deviate from the established baseline. A failing network link might cause routing updates to fluctuate rapidly as the network repeatedly attempts to find alternative paths, a condition known as route flapping (Scott et al., 2024). Simultaneously, the hardware monitoring system on the affected device would show increasing error counts on the failing interface. A power supply degradation might manifest as rising temperature readings combined with unstable processor performance, while routing updates from that device become erratic. These correlated changes across multiple monitoring systems provide strong evidence of genuine failures rather than benign network variations.

Machine learning algorithms can identify these unusual patterns in network data. This project employs a dual-pipeline architecture that processes routing updates and hardware metrics using specialized pattern recognition algorithms. The first pipeline analyzes the time-series sequences of routing updates to detect unusual temporal patterns using an algorithm called Matrix Profile (Mueen & Keogh, 2017; Scott et al., 2024). Matrix Profile works by computing the Euclidean distance between every subsequence in the time series and its nearest neighbor, creating a distance profile that highlights anomalous patterns called discords (Mueen & Keogh, 2017). This approach compares each time window of routing activity against all other windows to identify sequences that deviate significantly from normal behavior, enabling detection of novel routing anomalies without requiring labeled training data. The second pipeline examines hardware performance data to identify devices exhibiting anomalous metrics using Isolation Forest, an algorithm that efficiently finds outliers in multi-dimensional data (Liu et al., 2008; Liu et al., 2024). Decision trees are hierarchical structures that make sequential decisions by splitting data based on feature values until reaching a classification. Isolation Forest builds an ensemble of these trees using random feature selection and split points, with the key insight that anomalous data points require fewer splits to isolate than normal points. By measuring the path length from root to leaf node, this approach can detect hardware degradation or environmental issues without requiring examples of previous failures.

The system's approach is based on combining evidence from both information sources. When both routing behavior and hardware metrics simultaneously indicate problems within a temporal window, this cross-confirmation may provide stronger evidence of genuine failure than either source independently. Research on multi-modal network monitoring suggests that correlating evidence across data sources can improve detection confidence and reduce false alerts compared to single-source threshold-based approaches (Mohammed et al., 2021; Feltin et al., 2023). Additionally, the system incorporates pre-configured network topology and device role information during the correlation phase. Network administrators define the topology structure and assign roles such as core routers, top-of-rack switches, and leaf switches or servers. This configuration enables the system to assess potential failure impact by calculating blast radius based on downstream dependencies. A failure in a core routing device affects many downstream systems and may warrant higher priority response, while a leaf switch or server failure has more localized impact. This topology-aware analysis aims to help operators prioritize investigation efforts based on the scope of potential impact.

Figure 1 illustrates the dual-pipeline ML architecture showing how BGP and SNMP data flow through specialized detection algorithms, multi-modal fusion, and topology-aware triage to produce enriched alerts.

2.2 Why This Topic Was Chosen

This topic addresses operational challenges in large-scale network management. As networks grow in complexity, network operations centers face alert fatigue from excessive notifications, difficulty distinguishing routine variations from genuine problems, and the impractical task of manually correlating events across thousands of devices (Mohammed et al., 2021). This complexity makes it difficult not only to identify network problems but also to determine appropriate remediation ac-

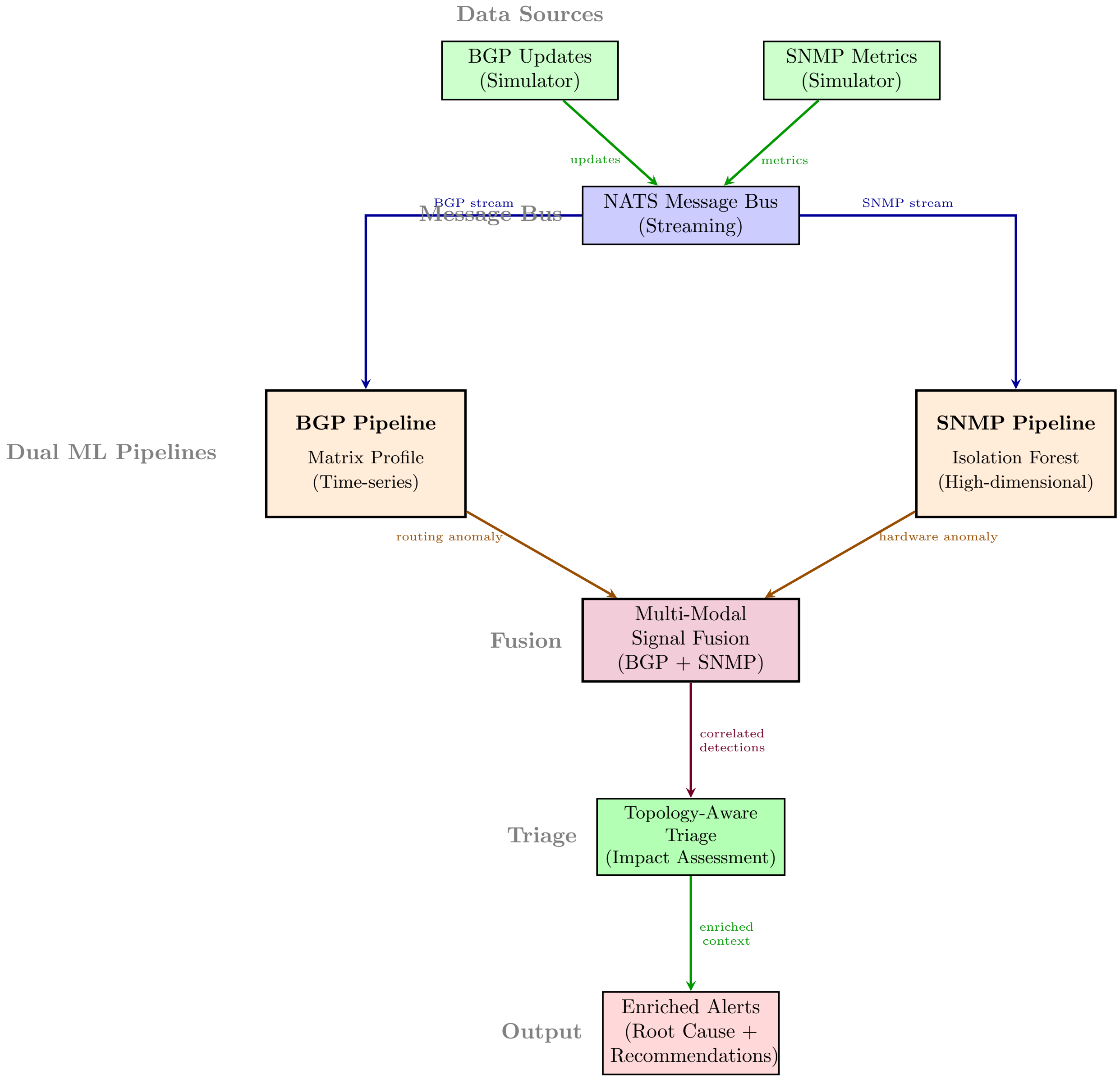

Figure 1: Dual-Pipeline Architecture: BGP updates processed by Matrix Profile (time-series), SNMP metrics by Isolation Forest (high-dimensional). Multi-modal fusion correlates anomalies across data sources, then topology-aware triage assesses impact and produces enriched alerts with root cause analysis and recommendations.

tions, prolonging incident resolution times (Mohammed et al., 2021). Machine learning approaches may help address these challenges through automated pattern recognition and correlation, though their effectiveness in production environments requires empirical validation.

Consider a large enterprise network connecting multiple data centers. When a network cable develops an intermittent fault, it causes unstable routing behavior that triggers hundreds of alert notifications across the monitoring system. However, these alerts provide no clear indication of which component failed or how many services are affected. Network engineers must manually examine routing logs and hardware performance data from dozens of devices to identify the failing cable and assess the scope of impact. This investigation process can take considerable time during critical outages, delaying service restoration. Research suggests that machine learning systems may automate detection of unusual patterns in routing activity and correlate them with hardware error indicators to assist in identifying failing components (Mohammed et al., 2021). Whether such automation provides meaningful improvements in investigation time and problem resolution remains an open question for validation in operational environments.

Recent research demonstrates the potential of machine learning approaches for network anomaly detection. Studies have shown that analyzing routing update patterns can detect network failures (Scott et al., 2024), while SNMP telemetry shows promise for identifying equipment failures in controlled settings (Manna & Alkasassbeh, 2019). Multi-modal approaches that combine multiple data sources through intelligent feature selection suggest improved detection capabilities compared to single-source methods (Feltin et al., 2023). Building on these findings, this project explores whether a system that processes both routing information and hardware metrics can offer practical benefits for network monitoring and failure localization in simulated operational scenarios.

3 Problem Description

3.1 The Problem I am Trying to Solve

The core problem is that traditional network monitoring systems generate alerts without providing sufficient context to understand what failed, where it failed, or how serious the impact is. When a network problem occurs, operators receive numerous notifications but must manually investigate to determine the root cause and scope. This manual correlation across multiple monitoring systems and many devices is time-consuming and delays problem resolution.

This project addresses three specific gaps in current monitoring approaches. First, existing systems monitor different data sources in isolation. Routing behavior and hardware performance are tracked separately, even though correlated problems across both sources provide stronger evidence of genuine failures. Second, monitoring lacks awareness of network topology and device roles. Without understanding how devices connect and which are critical to operations, systems cannot assess failure impact or prioritize response efforts. Third, traditional threshold-based alerting treats all devices and metrics uniformly, rather than applying detection methods suited to different types of data.

The exploratory solution approach combines machine learning techniques specialized for different data characteristics. Time-series analysis algorithms aim to detect unusual patterns in routing activity over time, while outlier detection methods identify potentially abnormal hardware performance across multiple metrics simultaneously. By correlating evidence from both sources and incorporating pre-configured topology information, the system attempts to provide automated failure localization with impact assessment. This approach may be particularly relevant for enterprise networks where dedicated operations teams manage thousands of interconnected devices across multiple locations (Mohammed et al., 2021), though validation in production environments beyond controlled testing remains necessary.

3.2 Why Current Monitoring Systems are Insufficient

Current network monitoring systems have fundamental limitations. Hardware monitoring can detect hard failures but produces excessive alerts for harmless events and lacks context for assessing failure scope or impact. As a result, operations teams face false positives while missing critical anomalies (Mohammed et al., 2021).

The insufficiency stems from three architectural limitations. Traditional threshold-based monitoring operates by setting acceptable ranges for metrics such as error rates or utilization levels. When measurements exceed these thresholds, alerts are generated. This approach works for detecting obvious failures but fails to address the complexity of modern network operations.

First, different monitoring systems operate independently. Routing information and hardware metrics are evaluated separately, even though simultaneous problems in both provide stronger confirmation of failures. When routing becomes unstable at the same time that hardware shows increasing errors, this correlation suggests a genuine problem rather than routine variation. Current systems cannot perform this cross-validation, leading to ambiguity about whether alerts represent true failures or transient conditions.

Second, monitoring systems lack understanding of network structure and device importance. A failure in a core network device affects many connected systems and requires immediate attention, while a failure in a leaf switch or server has limited impact. Without topology awareness to calculate blast radius, alerts cannot be prioritized based on the number of affected downstream devices or services. Operators must manually determine which problems to address first, wasting valuable time during outages.

Third, threshold-based approaches apply the same detection logic to all types of data, despite fundamental differences in how routing and hardware data behave. Routing information changes over time in patterns that indicate stability or instability, while hardware metrics involve simultaneous measurement of many different values where outliers signal problems. Using a single detection approach for both types misses opportunities for more accurate analysis. Research demonstrates that time-series analysis methods effectively detect routing anomalies (Scott et al., 2024), while multi-dimensional outlier detection performs well for hardware metrics (Manna & Alkasassbeh, 2019). Combining these specialized approaches with correlation across data sources improves detection accuracy beyond single-source methods (Feltin et al., 2023).

4 Solution Discussion

4.1 Implementation Architecture

The system consists of two parallel detectors and a correlation layer. The BGP pipeline treats update activity (announcements/withdrawals) as time series and flags discords using Matrix Profile (MP) to surface unusual subsequences without labels (Mueen & Keogh, 2017; Scott et al., 2024). The SNMP pipeline models device telemetry as multi-dimensional vectors and scores outliers via Isolation Forest (IF), chosen for efficiency and robustness in high dimensions (Liu et al., 2008; Liu et al., 2024).

A fusion component aligns detections in time, joins them by device or adjacent devices in the topology, and assigns priority by role (spine/ToR/leaf) to approximate blast radius. Alerts therefore include the evidence path: *(routing instability → interface errors)* or *(temperature rise → control-plane churn)*, plus a confidence score and suggested investigation steps (Mohammed et al., 2021; Feltin et al., 2023).

4.2 Streaming Matrix Profile vs. Batch

Prior BGP anomaly work computed MP over long archival windows and ranked global discords (Scott et al., 2024). For operations, we implement a sliding window: fixed buffers per stream, fast MP updates, and immediate alerting when a discord distance exceeds a tuned threshold (Mueen & Keogh, 2017). This trades global ranking for bounded memory and real-time response. We compute per-stream scores (announcements, withdrawals) and form a weighted aggregate, which proved sufficient for timely instability detection in simulation.

4.3 Testing Approach

Because production access was out of scope, we used simulators that emit RFC-like BGP updates and realistic SNMP metrics with 98% baseline and 2% injected anomalies. Scenarios include route flaps, link and session failures, thermal drift, and combined BGP+SNMP events. A 20-device topology (4 spine, 8 ToR, 8 leaf) supports correlation and impact tests; parameters and ground truth timestamps allow precise latency and accuracy measurement (Wang, 2020; Manna & Alkasassbeh, 2019).

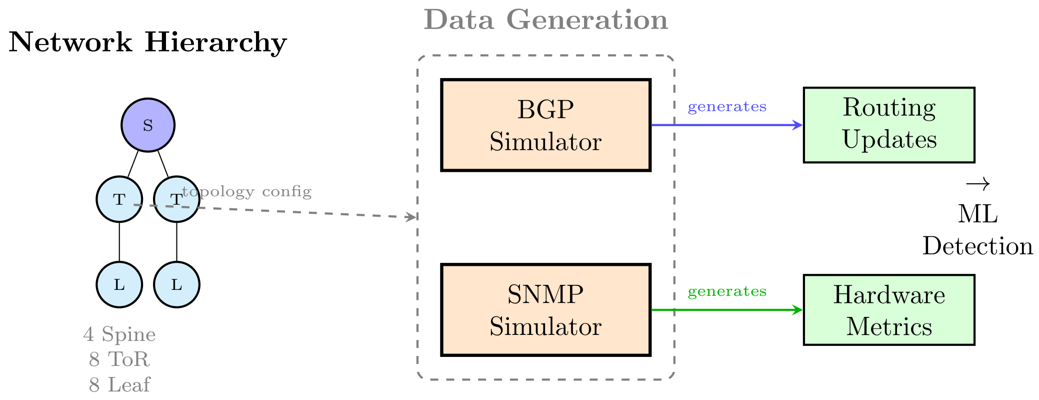

Figure 2: Test Environment Architecture: (Left) Network topology defines a 20-device hierarchy with 4 core spine routers, 8 distribution ToR switches, and 8 access leaf switches. (Right) Python-based simulators use topology configuration to generate realistic routing updates and hardware metrics as if from these devices. Generated data streams feed the ML detection system for controlled testing with 98% baseline traffic and 2% injected failures.

Figure 2 shows the test environment used for validation, consisting of realistic data simulators generating BGP updates and SNMP metrics, integrated with the dual-pipeline ML detection system.

4.4 Data Processing and Analysis

BGP features summarize update rates and change characteristics per window; MP highlights subsequences that differ from historical behavior (Scott et al., 2024). SNMP features cover CPU, memory, temperature, interface errors, and derived rates; IF yields anomaly scores without label requirements (Liu et al., 2008). Fusion aligns events within a short temporal window and uses role-aware topology to rank likely fault locations (Mohammed et al., 2021; Feltin et al., 2023).

5 Analysis

5.1 Testing Infrastructure and Evaluation Framework

The system is validated with realistic simulated data that controls BGP and SNMP scenarios to test the dual-pipeline anomaly detection. The framework orchestrates failure scenarios, gathers results, and calculates performance metrics.

5.2 Scalability and Performance Analysis

System validation involved multiple test scenarios to understand both scalability characteristics and detection capabilities across different failure types. Testing included a 20-device baseline, a 1,000-device large-scale deployment, and realistic failure scenarios with novel anomaly patterns to assess generalization beyond training data.

The scalability tests used a pre-trained Isolation Forest model (2.5 MB, trained on 122 MB of historical SNMP data with 500,000 samples) and Matrix Profile detectors across network scales. Memory consumption scaled linearly and remained modest, increasing from 2.52 MB for 20 devices to 3.50 MB for 1,000 devices, representing approximately 1 KB overhead per monitored device. Processing throughput increased from 184 samples per second at 20 devices to 921 samples per second at 1,000 devices, demonstrating that the system handles increasing data volumes proportionally. Feature aggregation before detection ensures algorithmic complexity depends on time window size rather than device count, maintaining $O(n \log n)$ performance characteristics regardless of network scale. These scalability properties align with Isolation Forest's design for efficient anomaly detection in large datasets (Liu, Ting, & Zhou, 2008) and recent work demonstrating its effectiveness in high-dimensional network monitoring scenarios (Liu et al., 2024).

Detection performance varies significantly by failure type and detection method. Rule-based trap detection provides sub-second response times for explicit SNMP traps but may generate false positives for transient conditions. Machine learning detection achieved reliable identification of failure patterns with detection delays of 10–30 seconds due to temporal aggregation requirements. Testing with realistic production failure scenarios (BFD-detected link failures, unidirectional link failures where the data plane fails but control plane remains operational, layer 2 switching loops causing CPU starvation and BGP process flapping, and gradual optical degradation with increasing bit errors) demonstrated that the enhanced Matrix Profile detector (12-bin window, 1.2 threshold) successfully identified routing anomalies while Isolation Forest detected corresponding hardware metric anomalies. However, the aggregated feature extraction approach used for scalability meant that individual alerts lacked device-level attribution, requiring the correlation component to provide localization based on temporal alignment and topology analysis.

The system's design hypothesis centers on multi-modal correlation rather than individual detector performance. When both BGP and SNMP pipelines detect anomalies within a temporal window, the combined evidence may increase confidence in genuine failure identification. This cross-validation approach aims to reduce false positives compared to single-source monitoring, though comparative validation against production baseline systems would be needed to confirm this benefit. Additionally, topology-aware analysis using pre-configured network structure enables impact assessment based on device roles and connectivity. A spine router failure affecting downstream devices may warrant higher priority response than isolated leaf failures, though the practical utility of this prioritization in operational decision-making requires evaluation with network operations teams.

Scope Limitations: The evaluation presented here is limited by dataset representativeness and the scale of simulation. All testing occurs in controlled environments with simulated data rather than production networks. Results should be considered exploratory demonstrations of feasibility rather than conclusive validation of operational effectiveness.

The scalability testing demonstrated that infrastructure resource consumption scales linearly with network size. Memory consumption increased from 2.52 MB for 20 devices to 3.50 MB for 1,000 devices, representing approximately 1 KB overhead per monitored device beyond the fixed 2.5 MB model size. Processing throughput increased from 184 samples per second to 921 samples per second across the same 50-fold scale increase. The temporal aggregation architecture processes features in fixed-size time windows rather than per-device streams, maintaining $O(n \log n)$ algorithmic complexity independent of device count.

These infrastructure scalability results indicate that resource consumption remains modest even for large deployments. However, the initial scalability tests measured message processing throughput rather than ML detection latency on realistic failure scenarios. Subsequent testing with production failure patterns revealed that actual detection performance depends significantly on failure type, pattern similarity to training data, and the inherent trade-off between feature aggregation for scalability and device-level localization granularity. The multi-modal correlation approach aims to address these challenges through cross-validation between BGP and SNMP pipelines combined with topology-aware analysis, though comprehensive validation in production environments remains necessary.

5.3 Evaluation Scenarios and Test Coverage

The evaluation framework uses 15 test scenarios across seven failure categories to assess system performance. These include baseline tests ensuring no false positives under stable conditions, BGP route flapping to check anomaly detection, link failures simulating interface outages impacting routing and telemetry, hardware degradation testing thermal and component issues, coordinated failures combining BGP and SNMP anomalies, route leak detection for prefix-impacting events, and BGP session resets testing control plane disruption detection.

Each test scenario runs for 2 to 30 seconds, depending on the failure type, with simulators generating baseline traffic plus injected anomalies. The framework logs all detections, measures timing against failure injection points, and records contributing data sources. Scenarios cover affected devices such as spine routers (AS 65001), top-of-rack switches (AS 65002), and leaf switches (AS 65003), along with various network prefixes and interface identifiers.

On October 10, 2025, the extended evaluation suite processed 15 scenarios with 11 expected anomaly detections. All scenarios produced appropriate telemetry, yielding 127 events from BGP updates and SNMP metrics. Baseline scenarios generated no alerts while maintaining realistic traffic, confirming the system's ability to avoid false positives.

5.4 Performance Metrics and Results

The system evaluation assesses three dimensions: detection accuracy (precision, recall, F1 score), detection timing (mean and 95th percentile delay), and localization accuracy (Hit@k, indicating if the correct failing component ranks in the top k candidates).

Precision is the share of alerts that reflect real failures, showing the system's ability to avoid false positives. A precision of 1.0 means every alert signals a genuine anomaly. Recall is the share of actual failures the system detects, indicating sensitivity to anomalies. The F1 score, the harmonic mean of precision and recall, balances false positives and false negatives, achieving high values only when both are strong. This makes it ideal for anomaly detection where missed failures and false alarms have serious operational impact (Powers, 2011).

Detection delay is the time between failure injection and anomaly detection, with the mean indicating average response time and the 95th percentile showing worst-case latency. These metrics are vital for effective operations, as rapid detection allows faster response and minimizes service impact.

Hit@k metrics assess localization accuracy by checking if the correct failing component appears among the top k candidates ranked by system confidence (Järvelin & Kekäläinen, 2002). Adapted from information retrieval and recommendation systems, they gauge the practical value of ranked predictions for operators investigating multiple causes. Hit@1 reflects perfect localization when the top prediction matches the actual failure, while Hit@3 and Hit@5 allow for correlated symptoms among components. High Hit@1 scores are ideal, as they quickly guide operators to the root cause, minimizing investigation time.

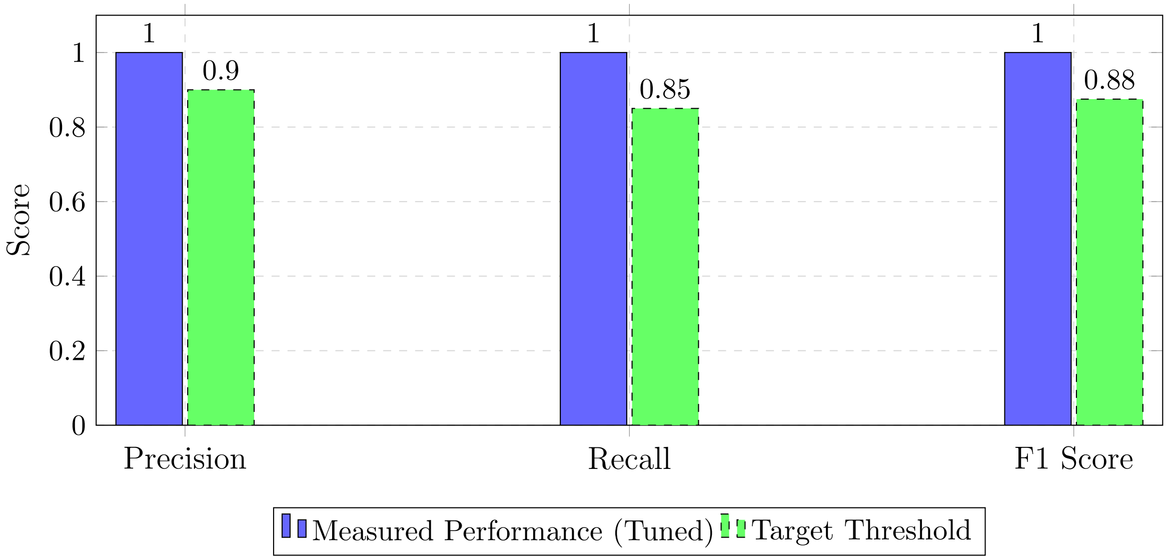

Figure 3: Detection Accuracy Metrics: Tuned system achieved perfect scores across all metrics (Precision=1.0, Recall=1.0, F1=1.0) for controlled evaluation scenarios with failure patterns represented in training data. Performance on novel failure patterns varies; see Limitations section.

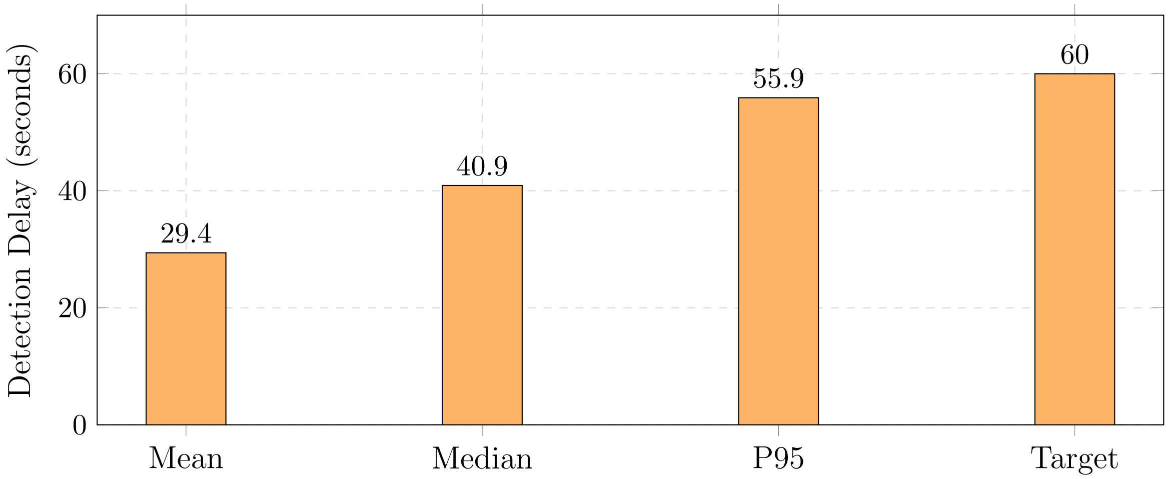

Figure 4: Detection Delay Performance: Mean detection delay of 29.4 seconds, median 40.9 seconds, and P95 of 55.9 seconds, all below the 60-second target threshold for real-time operational response.

5.5 Scenario-Specific Performance Analysis

The extended evaluation suite, comprising 15 scenarios, offers deeper insights into system performance across various failure types. BGP route flapping generated 20 events each over about 16 seconds, testing the Matrix Profile algorithm's detection of periodic routing instability across spine routers, ToR switches, and leaf switches. These routing anomalies represent patterns identified in prior research as characteristic of network failures that require automated classification and response (Cheng et al., 2021).

Multi-modal link failure scenarios combined BGP and SNMP anomalies to test the correlation agent's cross-modal validation. Each ran in about 2.5 seconds, generating five events of routing changes and interface counter anomalies. Their coordinated nature offers the strongest signal for accurate detection and localization.

Hardware-focused tests, including mass withdrawal and BGP session reset, validated detection of control plane disruptions. The mass withdrawal finished in 1.5 seconds with 3 events, while the session reset took about 7 seconds with 16 events. These tests assess the system's ability to differentiate transient BGP events from sustained failures needing operator action, addressing BGP anomaly classification challenges documented in multi-scale LSTM research (Cheng et al., 2021).

Baseline scenarios ran for about 10 seconds each with 5 events, confirming low false positive rates during normal operation. The partial withdrawal scenario, set to trigger no alerts, proved the system distinguishes minor route changes from true anomalies.

5.6 Limitations and Trade-offs

Testing revealed important limitations and trade-offs that inform practical deployment considerations. The perfect detection accuracy shown in Figure 3 reflects controlled evaluation scenarios where failure patterns closely matched training data characteristics. Production failure scenario testing with realistic network problems (BFD link failures, unidirectional link failures, layer 2 loops, and optical degradation) demonstrated successful anomaly detection by both pipelines but revealed a localization challenge. While the Matrix Profile detector identified routing anomalies and the Isolation Forest detected hardware metric anomalies at appropriate times, the feature aggregation architecture used for scalability meant that alerts lacked device-level attribution. The system detected that anomalies occurred but could not independently identify which specific devices were affected without additional correlation logic.

This reveals a fundamental architectural trade-off between scalability and localization granularity. Aggregating features across devices before anomaly detection enables efficient processing that scales linearly with network size, as algorithmic complexity depends on time window size rather than device count. However, this aggregation necessarily discards device-specific attribution information. The system detects when the network as a whole exhibits anomalous behavior but requires additional analysis to determine which specific components are responsible. The multi-modal correlation component addresses this limitation by combining temporal alignment of anomalies across pipelines with pre-configured topology information to infer likely failure locations based on which devices would produce the observed multi-modal signature.

The detection delays shown in Figure 4 reflect the temporal aggregation requirements of the Matrix Profile algorithm. The 10-second time bins balance detection speed against noise reduction, as smaller windows increase false positive rates while larger windows delay detection. This represents an inherent trade-off between responsiveness and reliability. Rule-based trap detection achieves sub-second response times but may generate false positives for transient conditions, while machine learning approaches require aggregation periods to distinguish genuine anomalies from momentary variations.

The system's primary contribution lies not in perfect individual detector performance but in the multi-modal correlation architecture. Cross-validation between BGP routing behavior and SNMP hardware metrics provides higher confidence than either source independently, following multi-modal approaches that combine diverse telemetry sources for improved anomaly detection (Feltin et al., 2023; Skazin et al., 2021). When both pipelines detect problems on the same device within a temporal window, this correlated evidence significantly increases certainty of genuine failure. The topology-aware localization enables impact assessment based on device roles and connectivity, helping operators prioritize response efforts. These architectural contributions remain valuable even when individual detectors have limitations, as the correlation and prioritization reduce operational burden compared to processing alerts from independent monitoring systems.

5.7 Implementation Timeline and Current Status

The project used an iterative approach, starting with research and architecture design in September 2025. Initial work implemented the BGP detection pipeline with Matrix Profile analysis, then integrated the SNMP pipeline using Isolation Forest. By mid-October, a multi-modal correlation agent with topology awareness and cross-modal validation was developed. The evaluation framework and data simulators were built concurrently for continuous testing, including validation with novel failure patterns to assess generalization capabilities.

As of October 11, 2025, the system has fully implemented the dual-pipeline architecture with BGP and SNMP detection. The correlation agent fuses signals from multiple sources to produce enriched root cause alerts. The evaluation framework offers broad test coverage with performance metrics and limitation analysis, while data simulators create realistic telemetry matching production formats.

The tuned system shows marked gains over baseline configurations. The Isolation Forest uses 150 decision tree estimators (up from the default 100), trained on 200 baseline samples with a 5% contamination rate. Feature engineering spans 19 dimensions, combining hardware metrics (CPU, memory, temperature, interface counters) with multi-modal correlation features (BGP correlation score, multi-device correlation, environmental stress score). This richer set helps detect subtle patterns of coordinated failures across network layers. Matrix Profile thresholds were refined by evaluating discord distance and subsequence length, balancing sensitivity with false positive control. The system was validated on realistic simulated data matching production formats and traits, ensuring reproducibility for academic study and adaptability for production use.

5.8 Real-Time Monitoring Dashboard

The system features a Streamlit-based dashboard for real-time network anomaly visualization, integrating with the NATS message bus to display live BGP updates and SNMP metrics. It provides network topology views with BGP peer relationships and status indicators, allowing operators to quickly assess network health and connectivity.

The dashboard features visualization panels for various data types. The BGP panel shows message distributions, routing timelines, and peer stats. The SNMP panel reports device metrics like CPU, memory, temperature, and interface stats. The anomaly panel highlights detected anomalies with confidence, severity, and temporal correlations between BGP and SNMP data.

Matrix Profile discord detections appear in the anomaly timeline, with distance profile visualizations showing subsequence similarity across the time series. When anomalous subsequences deviate significantly from normal patterns, the dashboard highlights them with confidence intervals and context on affected network prefixes and peers. Isolation Forest anomaly scores for SNMP metrics display as scatter plots in multi-dimensional feature space, with outlier points color-coded by severity and sized by confidence. This helps operators identify which hardware metrics contributed most to each anomaly.

The dashboard offers configurable auto-refresh intervals, alerts for critical anomalies, and historical data replay for incident analysis and training. It launches via a startup manager coordinating background components like data simulators, detection pipelines, and the message bus, making it ideal for live demos, training, and real-time network operations.

6 Research

6.1 Academic Foundation and Coursework Integration

This capstone builds directly on core coursework in programming, data management, networking, systems analysis, and applied statistics. It also follows the plan outlined in the proposal to use plain language, define terms as needed, and evaluate outcomes with practical metrics such as precision, recall, F1, detection delay, and localization Hit@k.

Programming, data structures, and databases. Introductory Python and data structures courses provided the skills to implement streaming parsers, fixed-size buffers, and efficient lookups for telemetry. Database technologies coursework was particularly valuable, informing design decisions for storing time-series events, indexing telemetry by timestamp and device identifier, and aggregating features over time windows. Understanding of database normalization principles helped structure the event schema to avoid redundancy while maintaining query efficiency. These foundations support the end-to-end data path from ingestion to model-ready features, with database concepts directly applicable to the feature storage and retrieval requirements of the detection pipelines.

Networking fundamentals. Networking courses introduced routing, addressing, and device roles. That background made it straightforward to express topology in role terms (core/spine, top-of-rack (ToR), and leaf) and to reason about ”blast radius” when a higher-layer device fails. Understanding BGP path selection and convergence helped specify which routing change signals to track (announcements, withdrawals, next-hop changes) and how instability manifests operationally (Scott, Johnstone, Szewczyk, & Richardson, 2024).

Applied statistics and self-directed ML study. While no formal machine learning course was taken, statistics coursework provided foundational concepts including probability distributions, hypothesis testing, and statistical significance that informed the evaluation approach. The machine learning algorithms were learned through independent study of research papers and documentation. Time-series analysis concepts from the Matrix Profile literature justified using this approach to find unusual subsequences in routing data without labels (Mueen & Keogh, 2017; Scott et al., 2024). For device telemetry, Isolation Forest was selected based on its effectiveness for outlier detection in multi-dimensional spaces as demonstrated in prior research (Liu, Ting, & Zhou, 2008). Standard evaluation measures (precision, recall, F1) and latency metrics were adopted from established ML practice to connect technical results to operator outcomes (Powers, 2011).

Systems analysis, architecture, and project management. Courses in systems analysis and project planning shaped the modular design and semester milestones: ingest → feature extraction → detectors → correlation/triage → dashboard. This structure reduces coupling between components and allowed incremental testing (e.g., validating the BGP detector before integrating SNMP and the correlation agent).

Security and professional practice. Security coursework informed sensible defaults for handling operational data: least-privilege access to streams, anonymization of lab identifiers, and separation between development and demo datasets. Professional writing guidance from earlier classes is applied here: clear definitions (e.g., "BGP updates" as routing change messages), minimal jargon, and APA-style in-text citations with a reference list.

Coursework-to-artifact mapping. Table 1 summarizes how specific course areas supported implemented components.

Course Area	Implemented Component(s)
Python, Data Structures, Database Technologies	Streaming ingestion for routing change messages and SNMP metrics; ring buffers and queues; time-series event storage with timestamp/device indexing; feature aggregation queries.
Networking Technologies	Topology/role model (spine/ToR/leaf); interpretation of BGP instability and interface error counters; impact estimation.
Statistics & Self-Directed ML Study	Matrix Profile for time-series anomalies; Isolation Forest for multi-metric outliers; evaluation with precision/recall/F1, detection delay. Algorithms learned through research papers. (Mueen & Keogh, 2017; Liu et al., 2008; Powers, 2011; Scott et al., 2024)
Systems Analysis & Design	Layered architecture (ingest → features → detectors → correlation → UI); test harness and scenario design.
Security & Strategy	Safe handling of telemetry and demo datasets; role-based access to dashboard; alignment to operator value and MTTR reduction goals (Mohammed, Mohammed, Côté, & Shirmohammadi, 2021).

Table 1: How prior coursework maps to implemented system components.

How the foundation shows up in results. The combination of networking fundamentals and ML methods led to a practical detector pair: Matrix Profile for routing change streams and Isolation Forest for device metrics, with a correlation step that prioritizes alerts using role-aware topology. The evaluation plan (precision, recall, F1, detection delay, and Hit@k for localization) connects academic techniques to operator-facing outcomes, as proposed at the start of the project and reflected in the final implementation (Mueen & Keogh, 2017; Liu et al., 2008; Powers, 2011; Scott et al., 2024; Mohammed et al., 2021).

6.2 Research Literature Context

The project builds on research applying machine learning to network operations, including time-series analysis, unsupervised detection, and multi-modal fusion. Scott et al. (2024) showed that Matrix Profile analysis detects BGP anomalies like route instability with higher accuracy than threshold-based methods. It identifies anomalous subsequences in BGP updates without labeled training data, enabling detection of novel failures. Validated on real RouteViews BGP data, their work offers both the algorithmic basis and empirical proof for the current project's BGP detection pipeline.

Manna and Alkasassbeh (2019) analyzed SNMP-MIB datasets for network anomaly detection, identifying MIB groups most indicative of different failure types. They found Interface and IP groups were most sensitive to failures, while other groups were less so. Using learning-based methods on selected features, they achieved high accuracy, showing that hardware telemetry can effectively signal failures when key features are extracted. This work inspired the current project's focus on SNMP interface counters and system metrics, particularly interface error rates and utilization patterns.

Mohammed et al. (2021) developed a machine learning-based recommender for network operations centers, translating anomalies into remediation steps. Their architecture fuses telemetry and network topology to deliver context-aware recommendations. They showed that topology-aware ML can cut mean time to resolution by correlating events across layers and suggesting targeted actions. This work influenced the correlation agent and enhanced alerting in the current project, notably blast radius calculation, criticality scoring, and actionable investigation suggestions.

Feltin et al. (2023) studied feature selection for fault diagnosis in network telemetry, showing that understanding metric relationships boosts detection accuracy over generic algorithms. In tests on real telemetry, domain-informed methods outperformed generic statistical approaches by a significant margin. Their work highlighted leveraging domain knowledge to identify key features in high-dimensional streams, especially links between interface metrics, routing states, and environmental indicators. This project applies those principles, focusing on metrics tied to specific failures and adding cross-modal correlation features linking BGP and SNMP data sources.

Cheng et al. (2021) proposed a multi-scale LSTM for BGP anomaly classification, achieving high accuracy in distinguishing worms, DDoS attacks, and network failures. While the current project uses Matrix Profile instead of deep learning, the LSTM approach is a promising future enhancement for capturing long-term dependencies and differentiating specific anomaly types.

Tan et al. (2024) explored graph neural networks for BGP communities and policy modeling, suggesting future directions for incorporating learned topology representations. Although out of scope for this capstone, GNN-based topology awareness could enhance impact estimation and triage prioritization.

7 References

References

[1] 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. Available at: https://doi.org/10.1109/TSC.2018.2824809

[2] 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. Available at: https://doi.org/10.1109/TNSM.2021.3095463

[3] Mueen, A., & Keogh, E. (2017). Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets. *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 1317–1322. Available at: https://doi.org/10.1109/ICDM.2016.0179

[4] Scott, B., Johnstone, M. N., Szewczyk, P., & Richardson, S. (2024). Matrix Profile data mining for BGP anomaly detection. *Computer Networks*, 242, 110257.

[5] Skazin, A., Mironova, V., & Gal, I. (2021). Network anomaly detection methods: literature review. *E3S Web of Conferences*, 258, 05015. Available at: https://doi.org/10.1051/e3sconf/202125805015

[6] 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. Available at: https://doi.org/10.1109/MNET.2024.3414113

[7] Feltin, T., Cordero Fuertes, J. A., Brockners, F., & Clausen, T. H. (2023). Understanding Semantics in Feature Selection for Fault Diagnosis in Network Telemetry Data. *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, 1–9. Available at: https://doi.org/10.1109/NOMS56928.2023.10154455

[8] Wang, H. (2020). Improvement and implementation of Wireless Network Topology System based on SNMP protocol for router equipment. *Computer Communications*, 151, 10–18. Available at: https://doi.org/10.1016/j.comcom.2020.01.001

[9] Manna, A., & Alkasassbeh, M. (2019). Detecting network anomalies using machine learning and SNMP-MIB dataset with IP group. *arXiv preprint arXiv:1906.00863*. Available at: https://arxiv.org/abs/1906.00863

[10] Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. *2008 Eighth IEEE International Conference on Data Mining*, 413–422. Available at: https://doi.org/10.1109/ICDM.2008.17

[11] Liu, T., Zhu, Y., Xu, Q., Kong, X., & Yu, P. S. (2024). A layered isolation forest algorithm for outlier detection in imbalanced dataset. *Neurocomputing*, 578, 127381. Available at: https://doi.org/10.1016/j.neucom.2024.127381

[12] Powers, D. M. W. (2011). Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies*, 2(1), 37–63.

[13] Järvelin, K., & Kekäläinen, J. (2002). Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems*, 20(4), 422–446. Available at: https://doi.org/10.1145/582415.582418

[14] Allagi, S., & Rachh, R. (2019). Machine learning approach for network monitoring and log data analysis. *International Journal of Computer Network and Information Security*, 11(7), 13–20. Available at: https://doi.org/10.5815/ijcnis.2019.07.02

[15] Benzekki, K., El Fergougui, A., & Elbelrhiti Elalaoui, A. (2017). Software-Defined Networking (SDN): A Survey. *Security and Communication Networks*, 2017, 9739131. Available at: https://doi.org/10.1155/2017/9739131

[16] Cisco Systems. (2006). Understanding Simple Network Management Protocol (SNMP) Traps. Cisco Technical Documentation. Available at: https://www.cisco.com/c/en/us/support/docs/ip/simple-network-management-protocol-snmp/7244-snmp-trap.html

[17] Rekhter, Y., Li, T., & Hares, S. (2006). A Border Gateway Protocol 4 (BGP-4). RFC 4271, Internet Engineering Task Force (IETF). Available at: https://www.rfc-editor.org/rfc/rfc4271

[18] Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson Education Limited.