LLM-Powered Network Anomaly Detection System

Arnav Kumar
Sujal Patel
Pulakita Maity

HP Internship Project

May 27, 2025

1 Business Drivers

The objective of this project is to enhance network monitoring by developing an open-source solution that automatically detects unusual patterns in network logs, ensuring early identification of issues. This system provides a smarter, adaptive approach to oversight by analyzing log data in real time, using advanced machine learning to spot deviations with high accuracy. Unlike traditional systems that often miss subtle issues due to predefined rules, this offers intelligent detection. It will collect and process large data volumes as they arrive and present clear visualizations of anomalies. The goal is to improve network reliability, enabling IT teams to respond quickly to irregularities for a stable, secure environment with minimal manual effort. The target is to improve IT team productivity by at least 30% through automated anomaly detection, enabling faster issue identification and resolution, and supporting proactive network management.

2 Business Requirements

ID	Description
B01	Centralize log collection from all network devices and servers to ensure consistent and comprehensive capture of system events, errors, and user activities for thorough monitoring.
B02	Enable real-time forwarding and streaming of log data to support timely analysis.
B03	Develop an adaptive anomaly detection system using advanced machine learning to identify deviations from normal network behavior with high accuracy, addressing subtle or any new issues.
B04	Provide security analysts and IT staff with a real-time, interactive dashboard for clear visualization, filtering, and rapid root-cause analysis of detected anomalies to facilitate quick response.
B05	Improve network reliability and performance by enabling early detection and rapid response to unusual network activities, minimizing downtime and operational disruptions.

3 Functional Requirements

Functional Requirement ID	Business Requirement IDs	Requirement Description
F01	B01, B03	Function to parse historical log data from the Network Logs Traffic dataset for input to the LLM, ensuring a robust foundation of past network activity patterns for training and anomaly detection.
F02	B03	Function to fine-tune the T5 LLM using LoRA (Low-Rank Adaptation) technique and train the model on historical log data to recognize normal network behavior and identify deviations effectively.
F03	B03, B05	Function to validate the model's accuracy using F1-score metrics on a test dataset, and save the trained model for deployment in anomaly detection tasks.
F04	B01, B02	Function to collect logs from all network devices and servers, centralizing them using syslog-ng to ensure consistent capture of system events and activities for comprehensive monitoring.
F05	B02, B04	Function to forward collected logs in real time to Apache Kafka, enabling efficient streaming of data for immediate processing and supporting scalability for growing log volumes.
F06	B02, B03	Function to parse forwarded logs from Kafka and prepare them as input to the fine-tuned T5 LLM, ensuring data is formatted correctly for anomaly analysis.
F07	B03, B05	Function to run the fine-tuned T5 LLM on real-time forwarded logs from Kafka, performing anomaly detection by identifying deviations from learned normal patterns.
F08	B04	Function to parse output from the LLM into a structured JSON file, organizing anomaly detection results for easy integration with other system components.
F09	B04, B05	Function to run an explanation mechanism using the LLM, generating human-readable descriptions of detected anomalies to aid in understanding and decision-making.
F10	B04, B05	Function to display the entire output, including detected anomalies and explanations, on an interactive Streamlit UI, providing real-time visualization, filtering, and search capabilities for IT staff and analysts.

4 Use Case Scenarios

Use Case Scenario	U01
Goal in Context	Ensure reliable collection and streaming of network logs from diverse sources to centralized processing
Functional Requirement ID	F04, F05
Scope	Implement syslog-ng configuration and Kafka streaming infrastructure for log aggregation
Level	Infrastructure
Preconditions	Network devices configured to send logs, syslog-ng/Kafka cluster operational
Success End Condition	Logs from all sources appear in Kafka topics within stated latency
Failure End Condition	Log loss exceeding threshold, alerts sent to administrators
Primary Actor	Log collection pipeline (syslog-ng + Kafka)
Trigger	Network device generates log event (e.g., connection attempt, error)

Use Case Scenario	U02
Goal in Context	Identify and explain network anomalies through ML analysis of streaming logs
Functional Requirement ID	F06, F07, F08, F09, F10
Scope	Anomaly detection workflow from Kafka consumption to dashboard visualization
Level	Application
Preconditions	Logs available in Kafka topics, T5 model deployed
Success End Condition	Anomalies displayed in Streamlit UI within given time of log arrival
Failure End Condition	Processing errors logged with root cause analysis
Primary Actor	Anomaly detection system (T5 LLM + Streamlit)
Trigger	New log message arrives in Kafka "netlogs-raw" topic

Use Case Scenario	U03
Goal in Context	Enable historical analysis of network logs to identify past anomalies and improve future detection accuracy through model retraining.
Functional Requirement ID	F01, F02, F03, F06, F08, F09, F10
Scope	Implement functions to parse historical data, retrain the T5 model, validate its performance, and present analysis results for past anomaly investigations.
Level	Not Applicable
Preconditions	Historical log data accessible in a structured format, T5 model previously trained with baseline accuracy, validation metrics defined.
Success End Condition	Historical anomalies are identified, model retraining improves detection accuracy, and results are visualized on the dashboard for review by IT analysts.
Failure End Condition	Errors during data parsing, model training, or result visualization are logged, with alerts sent to data scientists for resolution.
Primary Actor	Data processing and model training pipeline, managed by data scientists or automated scripts.
Trigger	Scheduled batch processing of historical logs or manual initiation of model retraining by IT staff.

5 Technical Components

Component	Implementation
Log Collection	• Syslog-ng for centralized log aggregation • Rest API formatted log ingestion • Scripting for batch upload
Log Streaming (Future Integration)	• Apache Kafka cluster • Kafka-C driver integration • Partitioned topics with 7-day retention
Anomaly Detection	• T5-base model fine-tuned with LoRA • CUDA-accelerated inference • PyOD for threshold tuning
Visualization	• Streamlit user interface • Time series interactive dashboard • Case reports
Model Training	• Historical log dataset • Hugging Face Transformers framework • F1-score validation metrics