Proactive IT network monitoring through log analysis using ML and Open AI

Asha Munemo
Department of Computer Science National University of Science and Technology
Bulawayo, Zimbabwe
ashamunemo@gmail.com

Samkeliso Suku Dube
Department of Computer Science National University of Science and Technology
Bulawayo, Zimbabwe
samkeliso.dube@nust.ac.zw

Tinahe Peswa Dube
Department of Agricultural Information Technology
National University of Science and Technology
Bulawayo, Zimbabwe
tinahe260@gmail.com

***Abstract*— This research focused on a machine learning technique (XGBoost – Extreme Gradient boosting), Transformer models (all-MiniLM-L6-v2 a sentence embedding model developed by Microsoft) based system for proactive network monitoring, performing log analysis for real-time anomaly detection and pattern analysis for root cause evaluation. This was done in order to address the challenge of reacting to problems only after they occur which leads to business revenue loss and increased idle time for workers when business operations are disrupted. The system makes use of the online NLP (natural language processing) model specifically (OPENAI or Cohere), which are inferred for intelligent problem explanation and solution recommendation. The methodology used was CRISP-DM for Data Science and incremental software methodology. The system enables network administrators to identify emerging problems within the network and address them pro-actively through system provided recommendations and anomaly evaluation insights before full negative impact on business operations.**

***Keywords*—Log analysis, Machine learning, Explainable AI, Pattern Analysis, Artificial Intelligence, Natural language processing)**

I.Introduction

In this modern world networks serve as the critical component in communication data exchange and service delivery throughout various industries. Effective and efficient network management has become most vital as networks grow in complexity and scale. Proactive network monitoring is the continuous monitoring and analysis of network performance as well as device behaviour to identify potential issues before they turn into a major failure [1] and reactive monitoring in contrast is triggered in response to an incident after it has occurred [1]. Reaction after a problem has occurred creates significant operational problems and financial loses, but predicting failure can ensure the continuous smooth-running of networks. Machine learning (ML) is an artificial intelligence subset that allows systems to discover patterns from data without being explicitly programmed [2]. .In network monitoring, ML models, including supervised and unsupervised algorithms, are used for the analysis of log data, identifying outliers, and finding patterns that commonly indicate faults or inefficiencies [3]. These models improve as more training data becomes available, thereby presenting a good solution for analysing huge clusters of unlabelled or semi-structured logs generated by network devices [3].

Anomaly detection refers to finding rare items, events or observations in data that do not conform to expected behaviour or differ significantly from the majority of data [3]. In many ways, an unusual pattern is a deviation from normal behaviour. Therefore, in network log analysis, an event may involve unusual log entries, and anomaly detection looks for those outliers in the data that can signal the occurrence of system failure, security breaches, or performance issues. Pattern detection involves identifying recurring sequences or structures within data. In network operations, this can include recognising common network traffic patterns, log entry sequences, or other repetitive behaviours that can indicate normal or abnormal activity [4].

Pattern analysis is the process of examining data to identify meaningful patterns or trends. It involves using statistical and machine learning techniques to uncover hidden structures and relationships within the data, which can be useful for making predictions, understanding behaviour, and making informed decisions [4]. Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. It involves the development of algorithms and models that can perform tasks that typically require human intelligence, such as visual perception, speech recognition, decision-making, and language translation [3]. OpenAI is known for developing advanced AI models such as GPT (Generative Pre-trained Transformer) series [1].

II.Literature review

A log file is data that is produced automatically when certain events occur in systems, networks, and applications [12]. Security and information technology operations teams use these to investigate and respond to abnormal system activities [12]. Some examples of log types include audit logs, transaction logs, event logs, error logs, message logs, and security logs.

Current networks and IT systems produce a vast amount of these logs and the continuous operation and uptime of these systems are critical to the success of the current businesses. IT (information technology) downtime creates a chain reaction, halting productivity and efficiency. Employees struggle to complete tasks, facing obstacles even with simple activities [13]. However, with the lack of an effective system that will parse, analyse, and learn from logs, the business risks failing to predict future issues and experiencing downtime, and, lack of an intelligent solution recommender. Administrators face extended estimated time of resolution [ETR].

Rresearch has been conducted in anomaly and pattern detection, and advanced techniques in anomaly detection with special emphasis on time series data because of its suitability

in log analysis for IT network monitoring. Time series anomaly detection is important in data analysis, focusing on identifying unusual patterns in time series data that significantly differ from expected behaviour [5]. It also incorporates feature extraction (FE) and dimensionality reduction (DR), such as auto encoders (AEs), to pre-process time series data that is complex and has a good amount of noise. This step is needed to improve the detection of collective-anomalies, which are deviations over time relating to a number of observations [6].

Some researchers have pointed out the exploitation of process mining (PM) techniques that can be drawn from time series data and comparison against real-time data for anomaly detection. [7] proposed a model which is integrated into supervised and unsupervised machine learning techniques with the use of domain knowledge in a very apt manner as far as identifying the anomalous log events is concerned. It can overcome the problems posed by having huge unlabelled data as well as associated log complexity that is important towards enabling proactive monitoring [7].

Research has been made in applying Open AI on managing tasks in a line server, managing security for the system by the AI agent, including going through logs for failed attempts of login and also pointing out the vulnerabilities [8].

Fault management has also been addressed in mobile networks through the use of ML techniques. Supervised learning methods are employed, achieving high accuracy rates of 95% for stationary and 90% for mobile signal level prediction, alongside multi-stage autonomous fault management [3].

Challenges in the field of log analysis based on [9] involve data volume and heterogeneity, privacy and retention policies for log data, cost and timeliness due to large data processing tooling support for developers

Opportunities include machine learning for log recommendation and extracting actionable insights [10]. [11] mentions integration of advanced machine learning techniques for log analysis in the context of anomaly detection and security auditing

III.Methodology

Design science and CRISP-DM methodologies were used in this research. These are specific in that the research embraces the designing and development of new or improving existing constructs-artifact-systems or processes and even models, toward solving a real world problem.

Business Data understanding

Organisations experience longer periods in resolving network problems when network systems go down leading to high losses in revenue and sales and increasing idle time among workers in different departments thereby halting productivity.

Developing a proactive network monitoring system assists administrators to proactively resolve issues before they negatively impact business operations. This system achieves the following: receives logs from networked devices, to predict failures of these devices through log analysis, thereby identifying possible causes using OpenAI API/Cohere, to generate actionable recommendations using the OpenAI API.

Data understanding

At this stage, the principal aim was to understand the log data that is used for training and evaluating the machine learning model. The first step was to collect log data from network devices in syslog format and then store it into batches for processing. Since comprehensive and up-to-date public datasets were not readily available or suitable for the target environment, logs were collected from a live network and further used as templates to simulate log messages issued by different network devices. These logs were classified and analysed, then used in the log generation tool to generate realistic and diverse log data. The final approach ensured that the data set was pertinent to real-world scenarios, yet it could not claim to be completely exhaustive. These logs included information related to network activities, system events, configuration changes, kernel logs, boot logs and potential fault conditions. Logs were in varying levels of detail.

Data restructuring and template Extraction:

Logs were extracted and categorised to remain with unique logs for templating.

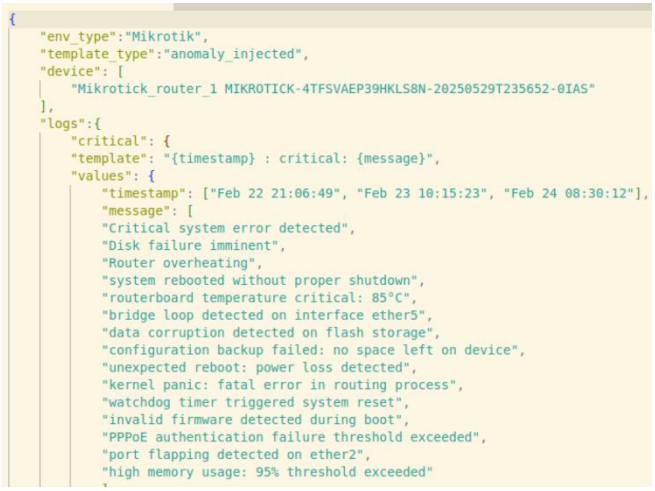


Figure1 Log Templating for Log Generator

Figure1 shows a portion of a log template that was used to generate logs. Numbers were dynamically randomised to simulate a real-world scenario

Data synthesis:

A custom log generator was developed to simulate both normal and anomaly scenarios based on extracted templates.

<125>May 27 06:44:02 windows_device_1 WINDOWS-N37CK9HKQ2S2BHR7-20250512T021200-AQ45: Error MEMORY Uncorrectable ECC error on DIMM DIMM_C1. System stability at risk

Figure 2 Generated Log in RFC3164 Format

Data quality assessment:

The processed data was evaluated for completeness, ensuring all critical fields were extracted, and for consistency validating so that templates and generated logs maintained uniform structure and imbalance meaning real anomalies are rare. This was managed through a controlled log generation, by using the probabilistic technique where a probability value for an anomaly to happen is set almost equal to balance the training set. It was set at very low for the purpose of testing.

Data preparation

This stage comprised the removal of irrelevant log entries, for example, keep system alive messages. Malformed logs were excluded, and time stamps were normalised to a standard format. Timestamp, host, message and text based features were extracted using natural language processing techniques. Logs were labelled automatically through the log generator as "normal" for normal logs

Data Modelling

The primary modelling goal was anomaly detection and pattern analysis in relation to anomalies, Modelling involved the following stages, Log parsing, feature extraction, and model training

Log parsing

This step involved splitting a log message into columns through regex matching with reference to the standard syslog format RFC3164 and RFC5424.

Custom Tokeniser: This step involved domain-specific tokenisation to pre-process log messages by extracting critical tokens and semantic units which corresponded to certain network activities in order to optimise feature extraction.

Phrase*r*: detects and groups multi word expressions into single tokens and helps improve downstream tasks.

TF-IDF Vectorizer: Log messages are transformed into numerical feature vectors by employing Term Frequency-Inverse Document Frequency (TF-IDF) method which allows the system to determine the relevance of certain terms within logs in relation to a specific data *subset.*

Word2Vec Embedding: this is a neural network based technique that learns vector representation of words. Words with similar meanings end up having similar vector representations in the embedding space, the architecture used is Skip-gram which predicts the surrounding context when given a word.

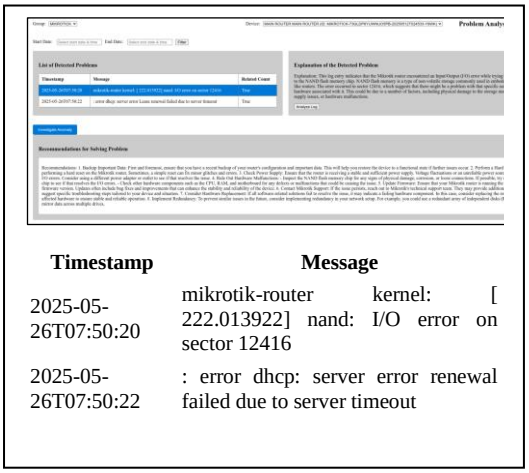
Feature Normalisation: The vectors are scaled uniformly to all features to improve functionality and stability of the downstream machine learning model.

Gradient Boosting Model (XGBoost Classifier): The model used as the core classifier for anomaly detection is an XGBoost–based Gradient Boosting Model.

IV.Results

The aim of the research was to design an intelligent system capable of proactive monitoring of network devices by performing log analysis.

The receiving of logs from networked devices was accomplished by implementing a syslog server within the system, whilst prediction of failures was achieved through log analysis- machine learning algorithms involving TF-IDF vectorisation, Word2Vec embedding, normalisation. Finally, training the XGBoost classifier model was used to determine whether the logs were abnormal or not, where, abnormal logs indicate issues requiring immediate attention. Finding probable causes and recommendations was done using OpenAI or Cohere upon detection of anomalies. The model queries OpenAI or Cohere language model to uncover probable causes from the logs and give recommendations on an action towards the detected problem.



Timestamp	Message
2025-05-26T07:50:20	mikrotik-router kernel: [222.013922] nand: I/O error on sector 12416
2025-05-26T07:50:22	: error dhcp: server error renewal failed due to server timeout

Figure 3 Analysis of Anomaly

Figure 3 is the system analysing a detected anomaly and giving an explanation of what the anomaly means, together with the recommendation of what actions should be done to mitigate or resolve the detected problem.

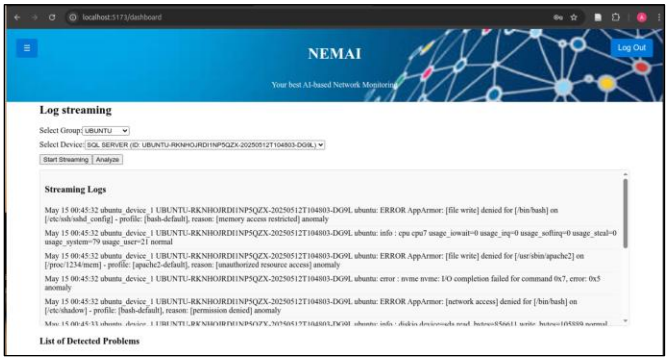


Figure 4 Log Streaming

Figure 4 shows the real time streaming of logs for a selected device. This helps to know if the device is actually connected and sending its logs to the system . Devices are assigned unique identities (ID) when they are added to the system, and this unique ID when configured in the targeted device will be received together with the logs.

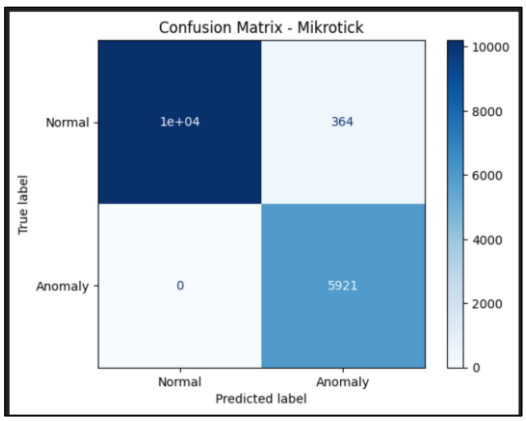


Figure 5 Model Performance

Figure 5 is a confusion matrix for the model obtained during testing. Anomalies that were detected as anomalies are 5921,anomalies that were detected as normal are 364 and normal that was detected as normal are 10 000, normal that were detected as anomaly were 0.

Even with diversifying the log data, the dataset remained non-exhaustive. There was a lack of logs coming from certain device types or from rare failure scenarios, and this might hinder the model's ability to generalise well in the much broader real world.

The researchers focused on common network events and device types which becomes a limit when considering rare or device specific failures and, they acknowledge that there is need for a deep intrusion into this scenario where we expand data collection to include a wider range of device models and network conditions

The researchers acknowledge that human in the loop evaluation was not done in this study, alternatively the researcher carried out a qualitative analysis of the LLM outputs by manually assessing the clarity and relevance of the generated explanations and recommendations serving as the initial validation of the systems usefulness, in future work we plan to include experts input evaluation to quantify satisfaction providing real relevance of the systems real world impact.

Adoption of transformer-based models: Considering models like BERT or Distil BERT could help better understand log messages at a semantic level. These models take into account word order and context, making them more effective at identifying unusual patterns—especially in complex or structured logs. Context-sensitive anomaly detection: Instead of treating each log entry in isolation, developing methods that analyse sequences of events (such as with LSTMs or Transformer-based encoders) could uncover anomalies that only appear when considering the flow of previous actions. Taking into account context meaning can improve accuracy. This learning approach can refine the model and reduce false positives with continued use.

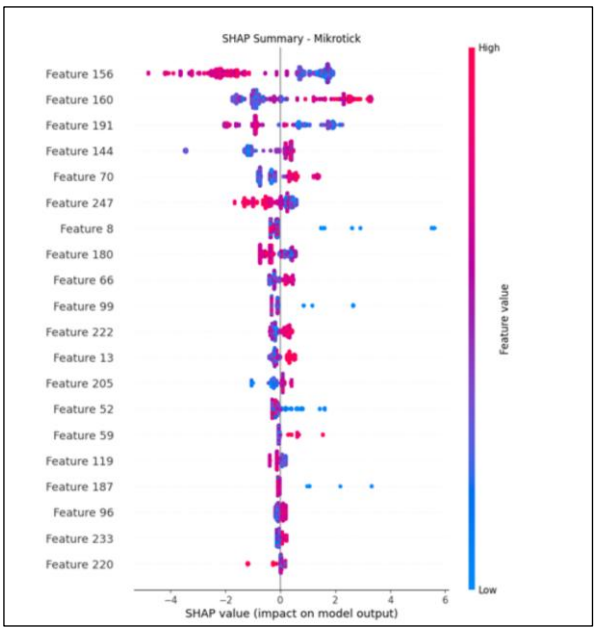


Figure 6 SHAP Summary

Figure 6 shows the impact of each feature on the model. Y-axis is the feature name and X-axis is the SHAP value, red means high feature value and blue means low feature value, the positive pushes prediction towards anomaly while the negative pushes prediction towards normal, those close to zero

have low impact on the prediction which features a dominance in the prediction process. Features 156 and 160 have the highest impact on predictions.

V.Discussions and Future Work

This research focused on the importance and need for proactive network monitoring through ML and NLP driven log analysis. Traditional reactive approaches are insufficient because of delayed ETRs leading to financial losses and business operation distraction. This gap was addressed by introducing a proactive network monitoring tool based on anomaly detection and pattern analysis through semantic understanding. The use of custom generated log data, which was extracted from real world networks, made sure that the training set was close to real world data though there is no claim that the dataset is exhaustive of the different logs.

The current implementation, inference is only done upon user request and not continuously, the system's real time component is on log streaming and only when a user selects to monitor the device. On latency, the major factor influencing latency is on the network infrastructure itself that is looking at bandwidth, routing efficiency, among other network concepts and also device performance, the model's inference time is minimal relative to delays in the network since they are invoked asynchronously.

The researchers used synthetic generated logs from authentic network templates to approximate realistic behaviour.

Proprietary models: The use of commercial APIs does introduce operational costs, in this project the LLM component is only triggered when an anomaly is found /detected and an explanation is required, hence reducing API calls and operational costs.

Reproducibility: The dependence on proprietary models can limit reproducibility due to changes in API behaviour or access policies, to reduce its effects, all API configurations were documented that is, the models, versions and prompts used in the study, and, in addition to this the system supports interchangeability with open source LLMS given all is in accordance to be able to use them for future offline operation.

VI.Conclusion

In conclusion, the project demonstrates a practical approach to intelligent network monitoring using machine learning, natural language and real-time log analysis.

References

[1] F. Liu, B. Farkiani, and P. Crowley, "Llms for Computer Networking Operations & Management: A Survey on Applications, Key Techniques, and Opportunities," 2024, *SSRN*. doi: 10.2139/ssrn.5074973.

[2] D. L. Vajda, T. V. Do, T. Bérczes, and K. Farkas, "Machine learning-based real-time anomaly detection using data pre-processing in the telemetry of server farms," *Sci. Rep.*, vol. 14, no. 1, p. 23288, Oct. 2024, doi: 10.1038/s41598-024-72982-z.

[3] S. Mukherjee, "Machine Learning Methodologies for Beyond 5G and 6G Heterogeneous Networks: Prediction, Automation, and Performance Analysis," 2024.

[4] F. Liu, B. Farkiani, and P. Crowley, "A Survey on Large Language Models for Network Operations & Management: Applications, Techniques, and Opportunities," Dec. 10, 2024, *Preprints*. doi: 10.36227/techrxiv.173386065.57486944/v1.

[5] F. Wang, Y. Jiang, R. Zhang, A. Wei, J. Xie, and X. Pang, "A Survey of Deep Anomaly Detection in Multivariate Time Series: Taxonomy, Applications, and Directions," *Sensors*, vol. 25, no. 1, p. 190, Jan. 2025, doi: 10.3390/s25010190.

[6] F. Vitale, F. De Vita, N. Mazzocca, and D. Bruneo, "A Process Mining-based unsupervised Anomaly Detection technique for the Industrial Internet of Things," *Internet Things*, vol. 24, p. 100993, Dec. 2023, doi: 10.1016/j.iot.2023.100993.

[7] A. H. Shah, D. Pasha, E. H. Zadeh, and S. Konur, "Automated Log Analysis and Anomaly Detection Using Machine Learning," in *Frontiers in Artificial Intelligence and Applications*, A. J. Tallón-Ballesteros, Ed., IOS Press, 2022. doi: 10.3233/FAIA220378.

[8] C. Cao, F. Wang, L. Lindley, and Z. Wang, "Managing Linux servers with LLM-based AI agents: An empirical evaluation with GPT4," *Mach. Learn. Appl.*, vol. 17, p. 100570, Sep. 2024, doi: 10.1016/j.mlwa.2024.100570.

[9] J. Cândido, M. Aniche, and A. Van Deursen, "Log-based software monitoring: a systematic mapping study," *PeerJ Comput. Sci.*, vol. 7, p. e489, May 2021, doi: 10.7717/peerj-cs.489.

[10] J. Cândido, M. Aniche, and A. Van Deursen, "Log-based software monitoring: a systematic mapping study," *PeerJ Comput. Sci.*, vol. 7, p. e489, May 2021, doi: 10.7717/peerj-cs.489.

[11] Y. Zhang, "Design and Implementation of a Computer Network Log Analysis System Based on Big Data Analytics," *Adv. Comput. Signals Syst.*, vol. 8, no. 6, 2024, doi: 10.23977/acss.2024.080607.

[12] S, Partovian, A. Bucaioni, F. Flammini, & J. Thornadtsson. (2023). 'Analysis of log files to enable smart-troubleshooting in industry 4.0: a systematic mapping study'. IEEE Access, 12, 147640-147658.

[13] S. H. Kendyala, (2023). High Availability Strategies for Identity Access Management Systems in Large Enterprises. Available at SSRN 5074869.