BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2025
http://excel.fit.vutbr.cz

Improvement of active network monitoring with explainable diagnostics

Dias Assatulla

Abstract

Modern network monitoring systems effectively detect errors but fail to provide meaningful interpretations, leaving specialists alone with technical logs. This work explores the integration of explainable diagnostics into active network monitoring using large language models (LLMs). The goal is to develop a system that detects issues and delivers human-readable explanations. A custom monitoring system was developed and integrated with an LLM, using prompt engineering technique to generate structured explanations. The models successfully interpret log errors and provide additional context. An online survey involving 13 participants confirmed the usefulness of this approach: the system accurately interprets logs and makes them more understandable for non-experts. As a result, it reduces the time required for incident analysis and eases the workload of system administrators, especially in critical situations where immediate context is needed. This work demonstrates a practical application of LLMs in IT infrastructure and shows potential as a valuable addition to existing monitoring systems.

*xassat00@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Modern networks are highly complex, layered systems that must ensure uninterrupted access to crucial applications and services. It is essential to continuously monitor the network infrastructure to quickly identify and prevent potential problems. Monitoring critical services can prevent an emergency from occurring and, if they do occur, minimize their consequences by responding rapidly to faults.

Active network monitoring enables network administrators to constantly evaluate network health by detecting issues before impacting users, using synthetic data and real-time component checks [1]. The example topology of this method is shown in Figure 1.

However, while this solution can identify network problems, it often does not provide a detailed and understandable explanation to answer the following questions: "*What does this error mean? Why did it occur? What might be the underlying cause of the error?*" and the like. Existing active monitoring tools focus primarily on fault detection, but lack advanced diagnostic capabilities. They typically provide raw logs or alerts without contextualized explanations.

This work explores methods for improving active network monitoring by integrating explainable diagnostics through the use of Large Language Models (LLMs). The aim is to detect network problems and generate structured and human-readable explanations that help identify the cause and provide potential recommendations to solve them.

2. Explainable diagnostics

When analyzing error logs, it is common to see error codes and short messages of 2-3 words related to these codes. For example, entries like '*404 Not Found*', '*501 Syntax error in parameters or arguments*', '*403 Forbidden*', or '*521 Server does not accept mail*' often appear in the server logs and indicate different types of issues that need to be addressed. Such messages often require further investigation to understand the underlying issue.

Explainability is an approach in which the system not only detects and records errors, but also explains their causes and provides recommendations for resolution. In traditional network monitoring systems, diagnostics is often limited to logging events and error codes, requiring network administrators to manually analyze logs and find the cause of problems, a process that typically requires expert knowledge and significant experience with the networks architecture and behavior. Such a process can be labor intensive, especially if the error occurs irregularly or is related to multiple factors.

2.1 Application LLM to diagnostics

To obtain the most structured and accurate LLM responses, the prompt engineering technique is used [2]. This approach consists in generating special queries (prompts) that guide the model to the desired format and content of the response. A well-formulated prompt allows the right results from the model to be obtained faster and more accurately. When creating a prompt, additional context, phrases, and clarifications are provided to help the model understand the task conditions. Examples of input data and desired output responses can also be provided. Based on this prompt, the model analyzes the input data and generates a response (Figure 2). The more precise and detailed the query, the higher the probability that the answer will fully solve the problem.

Taking all this into account, the following requirements are formulated for the prompt:

- It should give a brief explanation of what happened.
- It should classify the problem (or indicate that everything is working fine).
- It should identify the root cause of the error (if any).
- It should recommend corrective action (or note that no action is required if everything works normally).

2.2 Design of the diagnostic system

For this purpose, a custom monitoring system was developed and integrated with several LLMs (Gemini [3], Nous Hermes 2 [4], LLaMa [5]) of choice to allow the interpretation of log data if there is an error in the logs (as shown in Figure 3).

If Gemini is used as the model, the interaction takes place through the API. If LLaMA or Nous Hermes 2 is used, the log processing is carried out locally, using the internal resources of the device. The GPT4All tool is used to work with locally deployed models. GPT4All is an open source ecosystem for working with local LLMs developed by Nomic AI [6].

3. Evaluation of scenarios and results

The work involved experiments in three main scenarios:

- Interpretation of the network log.
- Temperature effect on model behavior.
- Consistency of responses.

A methodology based on expert evaluation was used to assess the quality of the responses generated by the LLM models. The goal was to determine how helpful the responses from the models are, how correctly they interpret the contents of the logs (including whether they correctly identify the presence or absence of an error), and which model performs best in different scenarios.

An example of a log file containing service errors is provided in Listing 1. In this example, the SMTP service authentication failed and the host was unreachable during the Connectivity test. Listing 2 presents an interpretation of the network log scenario, where the Gemini model provides a structured response to the errors in Listing 1, along with recommended actions.

4. Conclusions

Response option	Most selected (votes, %)
1 – Not useful at all	0 votes (0%)
2 – Slightly useful	0 votes (0%)
3 – Neutral	1 votes (7.7%)
4 – Quite useful	6 votes (46.2%)
5 – Very useful	6 votes (46.2%)

Table 1. Usefulness of using LLMs for interpreting monitoring results

A total of 13 people participated in the online survey. They evaluated the responses of the LLMs, compared them with each other, and selected which model performed better in various situations. In the end, the participants also answered a question about how useful they consider the implementation of LLMs to work with the monitoring results (see Table 1).

One of the possible deployment scenarios for the proposed system is integration as a chat-based diagnostic assistant. The LLM could be implemented as a bot agent in team communication platforms (e.g., Slack, Microsoft Teams), automatically posting summary updates every few hours or immediately notifying the team in case of critical failures with context.

Another improvement is anonymizing logs before sending them to cloud-based LLMs, such as Gemini. Since this model operates through an external API, the question of protecting confidential or sensitive data arises. For example, fields containing IP addresses, domain names, or hostnames could be automatically replaced with neutral values, which would allow the use of cloud models without the risk of information leakage.

In general, the work demonstrates the potential of using LLMs for network monitoring diagnostics. The results obtained and the prototype implemented showed that the application of LLMs can significantly increase the informativeness and clarity of the diagnostic results.

Acknowledgements

I would like to thank my supervisor doc. Ing. Petr Matoušek Ph.D., M.A. for his guidance, suggestions, and help while working on this project.

This work is part of the INVENTOR research project, which focuses on providing network administrators with tools to monitor cloud applications using active network techniques [7].

References

[1] Shanika Wickramasinghe. Active vs. passive monitoring: What's the difference? online, 2023.

[2] Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. Unleashing the potential of prompt engineering in large language models: a comprehensive review, 2024.

[3] Google DeepMind. Gemini developer api. online.

[4] NousResearch. Nous hermes 2 - mistral 7b - dpo. online.

[5] Meta AI. Meta llama 3 8b instruct. online.

[6] Yuvanesh Anand, Zach Nussbaum, Brandon Duderstadt, Benjamin Schmidt, and Andriy Mulyar. Gpt4all: Training an assistant-style chatbot with large scale data distillation from gpt-3.5-turbo. online, 2023.

[7] Ondřej Ryšavý, Marek Kuchynka, Petr Matoušek, Nelson Makau Mutua, Libor Polčák, and Jan Polišenský. Combined passive and active network monitoring. online, 2024. Code: FW10010040, Agency: Technologická agentura ČR.