BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INFORMATION SYSTEMS
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

IMPROVEMENT OF ACTIVE NETWORK MONITORING WITH EXPLAINABLE DIAGNOSTICS

ROZŠÍŘENÍ AKTIVNÍHO SÍŤOVÉHO MONITOROVÁNÍ O VYSVĚTLITELNOU DIAGNOSTIKU

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR **DIAS ASSATULLA**
AUTOR PRÁCE

SUPERVISOR **doc. Ing. PETR MATOUŠEK, Ph.D., M.A.**
VEDOUCÍ PRÁCE

BRNO 2024

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Bachelor's Thesis Assignment

162648

Institut: Department of Information Systems (DIFS)
Student: **Assatulla Dias**
Programme: Information Technology
Title: **Improvement of active network monitoring with explainable diagnostics**
Category: Networking
Academic year: 2024/25

Assignment:

1. Describe active monitoring techniques for network services.
2. Propose and implement an active monitoring system for selected network services. Evaluate its performance.
3. Extend the implemented system with explainable diagnostics using Large Language Models (LLM).
4. Propose some use cases for enhanced monitoring with explainable diagnostics. Test and evaluate their performance and contribution.
5. Compare your results with other existing solutions and discuss their possible deployment.

Literature:

- Chang Liu, Xiaohui Xie, Xinggong Zhang, Yong Cui: Large Language Models for Networking: Workflow, Advances and Challenges, 2024, arXiv:2404.12901.
- Lewis Tunstall, Leandro von Werra, Thomas Wolf. Natural Language Processing with Transformers. Building Language Applications with Hugging Face. Revised version. O'Reilly, 2022.
- Y. Huang *et al.*, "Large Language Models for Networking: Applications, Enabling Techniques, and Challenges," in *IEEE Network*, doi: 10.1109/MNET.2024.3435752.
- Sathiya Kumaran Mani et al. 2023. Enhancing Network Management Using Code Generated by Large Language Models. In Proceedings of the 22nd ACM Workshop on Hot Topics in Networks (HotNets '23). ACM, New York, NY, USA, 196–204.
- Omran Ayoub, Nicola Di Cicco, Fatima Ezzeddine, Federica Bruschetta, Roberto Rubino, Massimo Nardecchia, Michele Milano, Francesco Musumeci, Claudio Passera, Massimo Tornatore, Explainable Artificial Intelligence in communication networks: A use case for failure identification in microwave networks, Computer Networks, Volume 219, 2022, 109466, ISSN 1389-1286.
- Ł. Tulczyjew, K. Jarrah, C. Abondo, D. Bennett and N. Weill, "LLMcap: Large Language Model for Unsupervised PCAP Failure Detection," *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*, Denver, CO, USA, 2024, pp. 1559-1565, doi: 10.1109 ICCWorkshops59551.2024.10615433.

Requirements for the semestral defence:
Points 1-3.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor: **Matoušek Petr, doc. Ing., Ph.D., M.A.**
Head of Department: Kolář Dušan, doc. Dr. Ing.
Beginning of work: 1.11.2024
Submission deadline: 14.5.2025
Approval date: 22.10.2024

Abstract

This thesis focuses on improving active network monitoring by integrating explainable diagnostics using large language models (LLMs). A system was developed to monitor services such as SMTP, SSH, FTP, and host availability using ICMP. It generates structured logs for each service, which are sent to an LLM for analysis if an error is detected. Gemini, Nous Hermes 2, and LLaMA models were selected, and through prompt engineering, these models produce human-readable explanations detailing error causes and providing recommendations. The approach was evaluated in several scenarios and demonstrated practical usefulness. An online survey was conducted to assess the quality of the generated outputs, and the results confirmed that the explanations from the LLMs were perceived as clear, helpful, and relevant in most cases.

Abstrakt

Tato práce se zaměřuje na zlepšení aktivního monitorování sítě o vysvětlitelnou diagnostiku s využitím rozsáhlých jazykových modelů (LLM). Byl vyvinut systém pro monitorování služeb, jako jsou SMTP, SSH, FTP a dostupnost hostů pomocí ICMP. Pro každou službu jsou generovány strukturované logy, které jsou v případě detekce chyby odeslány k analýze LLM modelem. Byly vybrány modely Gemini, Nous Hermes 2 a LLaMA a pomocí technice prompt engineering tyto modely vytvářejí srozumitelná vysvětlení, která popisují příčiny chyb a poskytují doporučení. Tento přístup byl hodnocen v několika scénářích a prokázal praktickou užitečnost. Pro posouzení kvality vytvořených výstupů byl proveden online dotazník, jehož výsledky potvrdily, že vysvětlení z LLM byla ve většině případů vnímána jako jasná, užitečná a relevantní.

Keywords

Large Language Model, Prompt Engineering, Network Monitoring, Active monitoring.

Klíčová slova

velký jazykový model, promptní inženýrství, monitorování sítě, aktivní monitorování.

Reference

ASSATULLA, Dias. *Improvement of active network monitoring with explainable diagnostics*. Brno, 2024. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor doc. Ing. Petr Matoušek, Ph.D., M.A.

Rozšířený abstrakt

Cílem této práce je vylepšit aktivní síťový monitoring přidáním vysvětlitelné diagnostiky, aby se nejen zaznamenávaly chyby, ale také srozumitelně vysvětlovaly jejich příčiny a možné doporučení k jejich odstranění. Tento přístup zkracuje čas potřebný k analýze logů a usnadňuje rozhodování správcům sítě.

V rámci práce byl vyvinut vlastní monitorovací systém, který ověřuje dostupnost a správnou funkci služeb jako SMTP, SSH, FTP a ICMP. Výsledky se ukládají ve formátu JSON. Pokud je v logu detekována chyba (například hostitel není dostupný, nebo SMTP autentizace selhala), tento log je předán jazykovému modelu k analýze. Model následně vytvoří lidsky srozumitelné vysvětlení, které obsahuje: stručný popis incidentu, jeho klasifikaci, předpokládanou příčinu a doporučení pro vyřešení problému.

Pro analýzu byly vybrány tři různé LLM modely. Prvním je Gemini — uzavřený model od společnosti Google, druhým je LLaMa — částečně otevřený model od společnosti Meta, a třetím Nous Hermes 2, vytvořený skupinou nadšenců. Tímto způsobem se hodnotila práce 3 odlišných LLM modelů. Pro zajištění jednotnosti a přesnosti odpovědí byla použita technika prompt engineering. Speciálně vytvořený prompt umožnil získat přesnější, relevantnější a kvalitnější odpovědi od jazykových modelů, minimalizující pravděpodobnost získání vágních, nesprávných nebo nežádoucích výsledků. Systém byl testován v několika scénářích:

- hodnocení přesnosti interpretace logů,
- vliv parametru teploty na odpovědi modelů,
- stabilita generovaných výstupů při opětovném zadání stejného logu.

Pro zhodnocení užitečnosti odpovědí byl proveden dotazník: účastníkům byly předloženy různé interpretace logů vygenerované jednotlivými modely a měli posoudit jejich užitečnost a správnost. Výsledky ukázaly, že všechny tři modely jsou schopny generovat užitečná vysvětlení, zejména v typických případech chyb. Cloudový model Gemini se ukázal jako nejstabilnější v situacích, kdy operace proběhla úspěšně, ale má vysokou latenci.

Na základě provedených experimentů lze usoudit, že použití LLM v síťovém monitoringu může výrazně zlepšit diagnostický proces a zjednodušit práci síťových administrátorů. Je však také nutné zohlednit faktory, jako jsou:

- úroveň důvěrnosti dat,
- dostupnost výpočetních zdrojů,
- a charakter řešených úloh.

V perspektivě se takové systémy mohou stát součástí hybridních monitorovacích řešení nebo bot-agentů, kteří budou upozorňovat administrátory v kritických situacích s již připraveným kontextem, nebo poskytovat nějaké shrnutí.

Improvement of active network monitoring with explainable diagnostics

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of doc. Ing. Petr Matoušek Ph.D., M.A. I have listed all the literary sources, publications, and other sources which were used during the preparation of this thesis.

.....
Dias Assatulla
May 2, 2025

Acknowledgements

A huge thanks to my family who continue to believe in me.

Contents

Chapter 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.

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. The concept of explainable diagnostics offers an approach in which the system not only identifies and reports problems, but also explains the error and its causes. 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 thesis 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. For this purpose, a custom monitoring system was developed and integrated with several LLMs (Gemini, Nous Hermes 2, LLaMa) of choice to allow the interpretation of log data if an error exists in the logs. To ensure consistent and accurate responses from the models, the *prompt engineering* technique was used to define the structure and format of the generated explanations.

Chapter 2 provides a fundamental overview of network monitoring, explaining its purpose and significance. It introduces both active and passive monitoring methods, discussing their characteristics and applications in network management. Chapter 3 describes the design and implementation of an active monitoring system that is capable of collecting network logs. The primary purpose of this system is to generate structured network logs, which will then be processed by an LLM to produce human-readable explanations and actionable diagnostics. Chapter 4 provides an overview of the history of language models, introduces the fundamental concepts of transformer architecture, and discusses the classification of models into open source and proprietary categories. In addition, it presents the specific LLMs chosen for further experimentation. Chapter 5 describes the approach used to integrate LLMs into the network monitoring system, focusing on how they process collected

logs to generate explanations and diagnostics. Chapter 6 focuses on experiments in which the model receives a log to produce a diagnostic text.

Chapter 7 describes the results achieved, discusses the data obtained, touches on the problems encountered during the work, and is the conclusion of this thesis.

This thesis is part of the INVENTOR research project, which focuses on providing network administrators with tools to monitor cloud applications using active network techniques [1].

Chapter 2

Computer network monitoring

This chapter introduces the basics of network monitoring and its importance in maintaining stable and secure network operations. It discusses the key tasks of monitoring and its role in preventing and troubleshooting. It also present the basic concept of two monitoring approaches: active and passive. Their key differences as well as the tools used in each approach will be presented.

2.1 About network monitoring

The process of constantly monitoring a computer network to ensure its optimal operation, security, and performance is called network monitoring. Detecting and solving issues proactively is an effective way to avoid significant breakdowns. Monitoring improves the availability of services and maximizes the utilization of network resources, positively influencing overall performance. The objects of monitoring can be various network elements: from physical devices (servers, routers, switches) to logical elements (interface status, bandwidth utilization, device performance, applications, services, etc.). Logical elements are often defined as *metrics* because their data is measurable (CPU/Memory utilization in %, Storage usage in GigaBytes, etc.). Such data is primarily used for analysis and troubleshooting, since the physical device can be in UP state, but the network may operate slowly due to high load on communication channels, delays, or packet loss.

2.1.1 Main goals of the network monitoring

The *FCAPS* model within `ISO/IEC 10040` divides network monitoring into five areas which include *Failure Management*, *Configuration Management*, *Account Management*, *Performance Management*, and *Security Management* all of which work together to support network operations. Each of the network areas is important for network efficiency and security.

The main goal of *Failure Management* is to detect network problems and then determine their source in order to fix them in order to reduce service outages. This includes:

- Monitoring network devices and links at all times.
- Creating alert systems which notify staff of network failures.
- Logging of events and faults for future use.
- Troubleshooting network problems and restoring its normal functioning.

Configuration Management aims to maintain and control network configurations to ensure stability and consistency. This is achieved by:

- Gathering and saving configuration details from network devices.
- Documentation of changes made to device configurations.
- Automating backup and recovery configurations.

Account Management aims to collect, analyze, and use information about network traffic. It includes:

- Tracking the usage of network resources.
- To identify slow or insecure areas in the network from traffic.
- Creating reports on the performance of the network.

Performance Management monitors on link utilization, packet latency, and error rates among other network performance metrics. It focuses on:

- Identifying areas of congestion in the network and improving its performance.
- Predicting on how much load the network will have and subsequent network scaling planning.

Security Management focuses on monitoring security events, such as unauthorized access attempts or network attacks. This also includes:

- Performing security audits and vulnerability assessments.
- Using encryption and other secure communication protocols.
- Ensuring data confidentiality and integrity.

Following *FCAPS* model allows organizations to structure their approach to network monitoring, improving network stability, resilience, scalability, availability and security. The next section explains the basic concepts, operation, and advantages of active and passive network monitoring approaches.

2.2 Passive network monitoring

The distinctive feature of this monitoring approach is that it observes network traffic without actively interfering with it [2]. Unlike active monitoring, which is a proactive approach that interacts with the network by sending synthetic data over it, passive monitoring silently listens to the existing data flow for further processing. This can be achieved by using methods such as *Packet capturing*, *Flow monitoring*, and *Logging*. A brief description of each method is given in Table 2.1.

Method	Description	Tools
Packet capturing	Capturing network packets for detailed analysis, troubleshooting, and threat detection.	network TAPs, SPAN ports, tcpdump, wireshark
Flow monitoring	Collects, aggregates, and analyzes information about data flows through the network.	NetFlow, sFlow
Logging	Storing and logging events in the network, such as information actions, and the state of the system, application, or users.	Syslog, SNMP

Table 2.1: Passive monitoring methods.

An example topology is shown in Figure 2.1, which involves the use of port mirroring to capture copies of the traffic. The *red lines* show the ports from which data should be captured, and the *green lines* indicate the ports to which the data should be directed. TAPs [3] or SPAN ports function to replicate network traffic and direct it to a designated port. The copied data is subsequently sent to Data Storage for retention. Popular tools for analyzing network packets include Wireshark[1] and tcpdump[2].

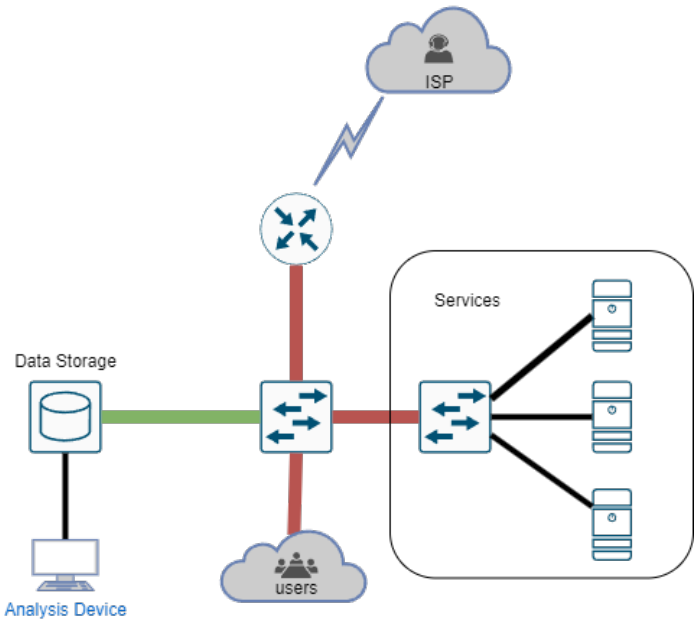

Figure 2.1: Passive network monitoring example topology. SPAN ports.

Both TAPs and SPAN ports perform the same function; the difference is that the TAP network is a separate device in the network, and in the case of SPAN, a separate port is allocated on the switch. According to this white paper [3], the TAP device does not affect network performance as it replicates data at the physical layer and sends them to separate monitoring ports, unlike the SPAN port, which is a software-built operation that can cause additional load on the switch/router, increasing network latency.

[1] https://www.wireshark.org
[2] https://www.tcpdump.org

The advantage of this approach is that it analyzes the actual network traffic that passes through the network and that there is no generation of additional traffic. This allows us to identify *problem areas* and *bottlenecks* in the network. Based on these data, targeted improvements can be applied to degraded elements, and this information can be used to better plan future scaling and resource optimization.

However, passive network monitoring has a number of disadvantages. It is not capable of reacting to real-time events, but it only records the fact that a problem has occurred without interfering with its solution. In addition, it is sometimes impossible to get a complete picture when analyzing traffic because most popular network services use encryption, which in turn limits the ability to identify threats hidden in encrypted traffic. There is also a need to store huge amounts of data and analyze it later, which requires significant financial expenditures.

In addition to the previously described tools, *IDS (Intrusion Detection System)* and *NetFlow* [4] can also be used in passive monitoring. IDS is designed to analyze network traffic and detect suspicious activity [5], while NetFlow is used to collect and analyze network traffic flow statistics to optimize network performance.

In summary, passive monitoring evaluates user activity in applications and services by analyzing actual data, giving insights on service utilization, and helping optimize network performance.

2.3 Active network monitoring

While passive monitoring analyzes and collects data on events that have already occurred, active monitoring focuses on proactively detecting issues. It does this by sending synthetic data or test requests through the network, allowing faster identification of potential problems before they escalate [2].

Synthetic data are used here to test specific applications or services. What is meant here by synthetic is that the data are created specifically for testing rather than arising from actual user sessions or applications. Such data is used to measure various network parameters. Network administrators have more control over these data, as they decide what sizes, number of requests, frequency, and so on will be sent.

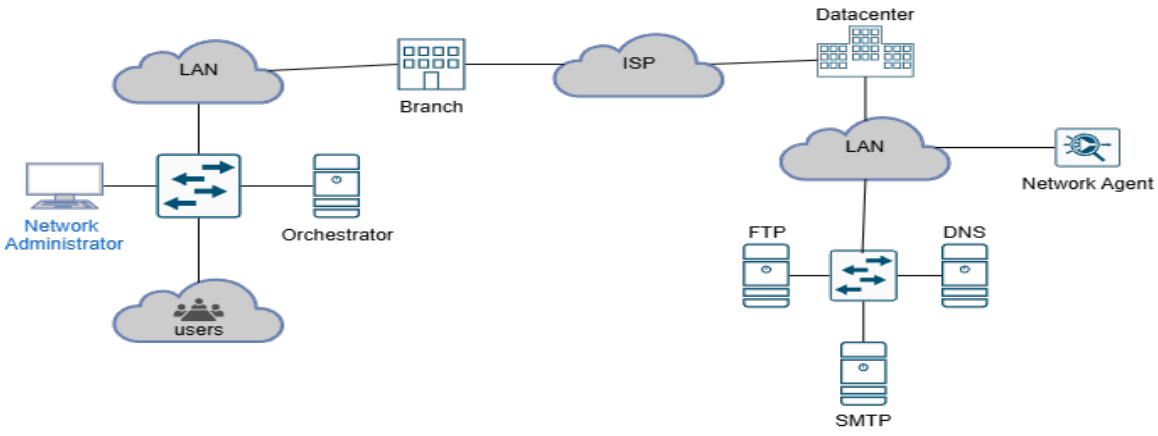

Figure 2.2: Active network monitoring example topology.

In Figure 2.2 an example topology is shown with included services that require monitoring, network agents, and a dedicated orchestrator server. Network agents use some specially

created test or monitoring software and generally perform specific tasks on the network on behalf of an administrator or other application (in the case of software). These agents are often located in different parts of the network segment, act autonomously, and send their results to the orchestrator. Orchestrator, as the name implies, manages all network agents: it configures when they should start executing commands and stores the results with it.

This approach generally includes many tools that help ensure the stability and performance of networks and applications. One of the simplest and most commonly used is Ping[3]. It sends ICMP [6] requests to the target host, measures latency, and checks host availability and response time. When it is important to measure the maximum throughput of the network, the utility iperf[4] is used. This tool generates traffic between nodes to assess speed test. For network diagnostics and scanning, nmap[5] is often used. Although it is mostly used for security purposes, it can also be helpful in active monitoring: checking open ports, determining the number of active nodes in a subnet, and analyzing different scenarios. For monitoring web applications, there is a simple curl[6] utility. It sends web requests to specified nodes, checking their availability, response speed, and accuracy.

Such tools are an excellent foundation for active network monitoring, providing basic information on network availability and performance. However, this may not be enough for complex infrastructure and specific applications.

Imagine a system where multiple microservices communicate with each other through internal APIs. A `ping` can confirm that each microservice is reachable over the network, and a `curl` can confirm that an individual endpoint returns a `200 OK` code. However, if the interaction between two services is broken, for example, due to incompatibility of the API versions, neither `ping` nor a simple `curl` query will reveal it.

In these situations, *special tests* are necessary to simulate real interaction scenarios between services. For example, a function call of one microservice that requires another response allows one to check the availability and correctness of business logic and compatibility. These tests reveal errors in code, configuration, or integration logic that are not visible when using standard utilities. Simulating user actions as part of these tests helps detect potential failures before they affect real users and business processes. This approach also allows for the identification of problems before users notice them.

Despite the advantages of flexibility, it should be noted that using generated data to test a service or application is an additional load on the network, and the results may only sometimes be accurate and may have errors.

2.4 Conclusion

This chapter reviews the basic concepts of network monitoring and active and passive approaches. The main distinguishing features of each approach are described, as well as their specific purposes and pros and cons. A brief distinction between the two approaches is described in Table 2.2. The next chapter will describe the implementation of a service-specific monitoring system using active network monitoring.

[3] https://linux.die.net/man/8/ping
[4] https://iperf.fr
[5] https://nmap.org
[6] https://curl.se

	Active monitoring	Passive monitoring
Usage	availability monitoring, network performance, SLA control.	analysing incidents, gathering traffic statistics
Principle of operation	Sends requests to devices on the network for information.	Listens to traffic without interfering with it.
Data type	Synthetic data	Real data
Proactivity	Very proactive, can detect problems before they escalate.	Less proactive, reacts to events that have already happened.
Load on network	Generates additional traffic	Minimal, as it does not generate additional traffic.
Tools	nmap, ping, curl, iperf, etc.	Wireshark, tcpdump, IDS, NetFlow, etc.

Table 2.2: Summary: active vs. passive network monitoring

Chapter 3

Active monitoring system

This chapter describes the design of a system to monitor specific network services. It looks at services to be monitored and what is being analyzed to ensure they are working correctly. It also gives an overview of the tools used and how the monitoring results are stored. In addition, an experiment will be carried out to test the system to ensure that it is functioning correctly and that the data obtained are reliable.

3.1 System design and services

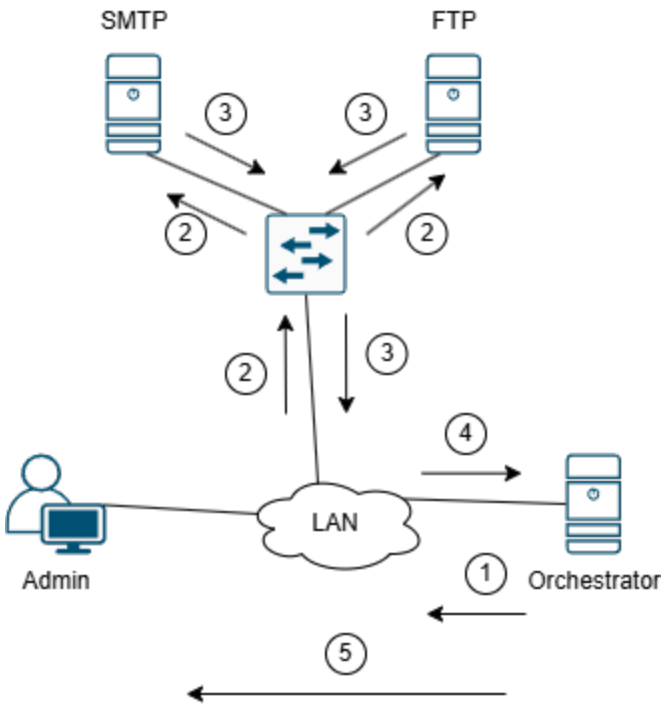

Figure 3.1: Logical design of monitoring system.

Starting with the design specified in Figure 3.1, there is an orchestrator that runs the tests and stores their results, a server or servers that run the network services, and a network

administrator who receives the results from the orchestrator and has access to make changes to the orchestrator.

3.1.1 Description of functionality:

1. The orchestrator runs tests that verify the operation of certain services by sending requests.
2. These requests go through internal routing to the subnet where the services are located.
3. The services respond to requests as if they were sent by an ordinary user.
4. The orchestrator receives these responses, processes them, and stores results in a specific format.
5. The orchestrator sends a copy of the results to the administrator.

The peculiarity of this design lies in the simplicity of its implementation, as well as its intuitive comprehensibility.

For this thesis, the following network services were chosen: FTP [7], SSH [8], SMTP [9], and ICMP [6]. The latter, although not a traditional network service, is used to actively monitor server availability. The availability check with the ICMP protocol allows you to quickly determine the status and availability of the server, making it a useful monitoring tool. The important details for understanding these services are shown in Table 3.1.

Network Service	TCP/UDP	Port	Usage
SMTP	TCP	25, 587, 465	Transfers mail over the network
FTP	TCP	20, 21	Transfers file over the network
SSH	TCP	22	Secure connection to access a computer
ICMP	-	-	To check device availability

Table 3.1: Network services to be tested.

3.1.2 Parameters tested for each service

With active monitoring, it is possible to check the specific parameters of each service, allowing for more accurate data on the state of the system. This approach to monitoring helps focus on key metrics that have the most significant impact on the performance of services. The following are the parameters for each service that will be checked:

- SMTP (Simple Mail Transfer Protocol):
  - *Welcome message:* Confirms server readiness for communication.
  - *Availability:* Checks server connection and response.
  - *Authentication:* Verifies credentials (username and password).
  - *Send email:* Ensures that the service can send emails.
  - *Session termination:* Confirms proper session closure.
- FTP (File Transfer Protocol):

  - *Welcome message:* Indicates the readiness of the server for file transfer.
  - *Availability:* Checks server connection and response.
  - *Authentication:* Validates user credentials.
  - *Session termination:* Confirms proper session closure.

- SSH (Secure Shell):
  - *Availability:* Confirms the server's ability to establish an SSH connection.
  - *SSH banner:* Retrieves server information and security details.

- ICMP (Internet Control Message Protocol):
  - *Availability:* Verifies server reachability through echo requests.
  - *Latency:* Measures the round-trip time of packets.
  - *Jitter:* Evaluates the variability in packet delay.
  - *Packet Loss:* Assesses the reliability of the connection.

Analyzing these parameters will help assess the correctness of service operations and identify potential bottlenecks in their functioning.

3.2 Implementation

The practical part of this thesis is the implementation of selected tests written in Python[1] programming language. The list of libraries that were used is described in Table 3.2. Most libraries are built-in modules, so their version is not usually shown. Except for `icmplib`, in this thesis, version `3.0.4` was used. All results are stored in JSON[2] format, which makes it easier to process and analyze them further.

Library	Purpose
json	for storing test results
yaml	used for configuration file
subprocess	used to run other Python scripts
datetime	is used for manipulation and formatting of the date and time
time	for measuring time delays
icmplib	send ICMP echo messages
getpass	used for authentication process
email	used in creating and parsing email messages using SMTP
socket	to work with BSD socket interface
ftplib	interaction with FTP servers
smtplib	interaction with SMTP servers

Table 3.2: Python libraries.

[1] https://www.python.org
[2] https://www.json.org/json-en.html

The implementation scheme is shown in Figure 3.2. The main component of the system is the `orchestrator.py` script, which loads the necessary configuration parameters from the `config.yaml` file. These parameters are then passed to each individual service test.

The service tests, using the received parameters, send the appropriate commands to the target services and get the results. The results are then returned to the main script, which saves them in a file named `log_datetime_HH:MM:SS.json`.

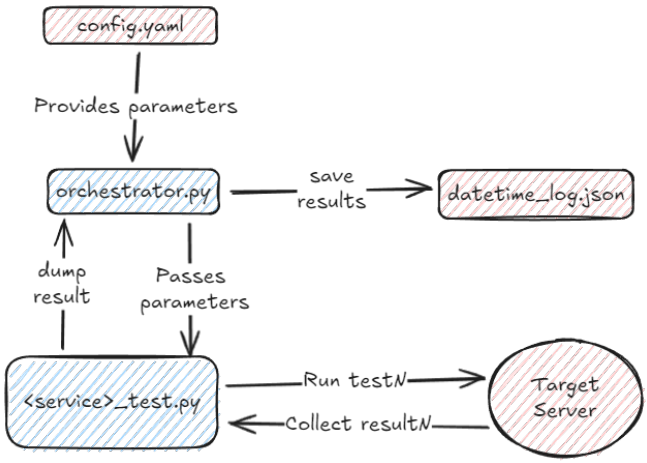

Figure 3.2: Logical design of monitoring system.

3.2.1 Expected input values

In the testing of network services, a single configuration file `config.yaml` is used to provide all necessary parameters for the execution of the test. The main script reads this file and passes only the required parameters to individual test scripts. An example of a configuration file that contains the input values for each service is shown in Listing 3.1.

```yaml
services:
- name: SMTP
  ip: localhost
  port: 465
  username: smtpuser
  password: pwd // Plaintext password: will be encrypted after first run
  from: smtpuser@example.local
  to: smtpuser@example.local
- name: FTP
  ip: localhost
  port: 21
  username: user
  password: ENC(pwd) // Encrypted password: ENC(...) means encrypted
- name: SSH
  ip: localhost
  port: 22
- name: Ping
  ip: localhost
  count: 5
  timeout: 1
  interval: 0.2
```

Listing 3.1: Input values example.

Thus, all necessary parameters are stored in one place and, if any changes need to be made, they are made in this file. The value of each parameter, its description, as well as characteristics such as data type (string, integer, etc.), are specified in Table 3.3.

Service	Parameter	Type	Description
SMTP	ip	string	Target IPv4/IPv6 address or hostname
	port	int	Connection port
	username	string	Username for login
	password	string	Password for login (plaintext)
	from	string	Sender's email address
	to	string	Recipient's email address
FTP	ip	string	Target IPv4/IPv6 address or hostname
	port	int	Connection port
	username	string	Username for login
	password	string	Password for login (plaintext)
SSH	ip	string	Target IPv4/IPv6 address or hostname
	port	int	Connection port
Ping	ip	string	Target IPv4/IPv6 address or hostname
	count	int	Number of packets to send
	timeout	int/float	Response timeout (in seconds)
	interval	int/float	Interval between packets (in seconds)

Table 3.3: Configuration file input values.

3.2.2 Expected output values

The result of all tests is the generation of a single JSON file. This file contains an array, each element of which is a JSON object with data from the specified service. Each object contains common data that are present for every service, such as the service name, test status, test date, etc. A more detailed list of common fields is provided in Table 3.4. The expected output values for each service are described in the following tables. 3.5, 3.6, 3.7, and 3.8.

Field	Type	Description
service	string	Name of the tested service (e.g., SMTP, SSH, FTP, ICMP).
status	string	Test result (e.g., complete, error).
datetime	string	Test execution timestamp in ISO 8601 format.
target_host	string	Hostname or IP address of the target.
ip_address	string	Resolved IP address (appears if `target_host` is a domain name). Supports IPv4 and IPv6.
context	JSON object	Detailed results of actions performed (for SMTP, SSH, and FTP).
response_time_ms	float	Service response time in milliseconds (for SMTP, SSH, and FTP).

Table 3.4: Common fields in expected output values.

Field	Type	Description
operation	string	General structure for SMTP operations.
operation.status	boolean	Status of the operation.
operation.smtp_code	string	SMTP response code.
operation.smtp_msg	string	SMTP response message.
EHLO	string	Supported SMTP features.
NOOP	operation	NO-Operation command results.
AUTHENTICATE	operation	Authentication command results.
SEND EMAIL	operation	Email transfer command results.
QUIT	operation	Session closure command response.

Table 3.5: Expected output values for SMTP service.

Field	Type	Description
is_alive	boolean	Host reachability status.
latency_avg_ms	float	Average latency in milliseconds.
latency_min_ms	float	Minimum latency recorded in milliseconds.
latency_max_ms	float	Maximum latency recorded in milliseconds.
jitter_ms	float	Variation in latency (jitter) in milliseconds.
packet_loss	float	Percentage of packets lost during the test.

Table 3.6: Expected output values for ICMP service.

Field	Type	Description
welcome_msg	string	Welcome message from the FTP server.
login_msg	string	Response message for login attempt.
list_msg	string	Response for directory listing command.
quit_msg	string	Response for the QUIT command.

Table 3.7: Expected output values for FTP service.

Field	Type	Description
banner	string	SSH server banner message.

Table 3.8: Expected output values for SSH service.

3.3 Experiments

This section aims to test the accuracy of scripts developed to monitor the network services listed in Table 3.1.

3.3.1 Environment description

`Windows Subsystem for Linux` (WSL) is used as a test environment for the experiment, where the `22.04.5 LTS` version of the Ubuntu operating system is installed. Scripts will be run in this environment, and real servers will be used as target servers. Two of these

servers are university hosted servers with already installed services, allowing experiments to be conducted under conditions as close to real-world operation as possible.

The following network services were tested during the experiment:

- *ICMP* — connectivity was tested using the university server `kazi.fit.vutbr.cz`.
- *SMTP* — the university server `eva.fit.vutbr.cz` was used for the testing.
- *SSH* — testing was also conducted on the university server `kazi.fit.vutbr.cz`.
- *FTP* — since no suitable external FTP server with authentication support was found, a local server was set up using `pyftpdlib`, allowing for authentication testing.

3.3.2 SMTP test

The script responsible for monitoring the SMTP service is named `smtp_test.py`. It checks the functionality of the service using the `smtplib` library. The communication process is shown in Figure 3.3.

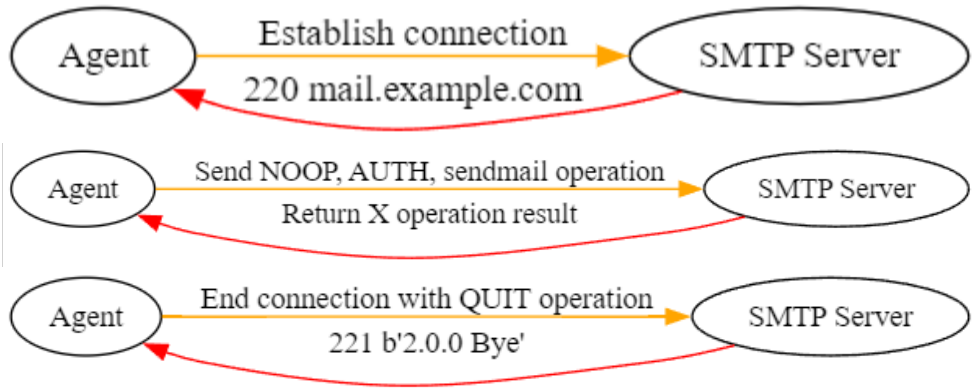

Figure 3.3: SMTP test communication process. Sequence from top to bottom.

This test sequentially checks the functionality of the SMTP service. First, a connection to the server is established, after which the `NOOP` command is sent to check if the service is operational. Then authentication is performed, after which the script will try to send a test email. The final stage is the correct termination of the session using the `QUIT` command.

The results of executing all these commands will be collected and used to evaluate service performance.

Input example

```
services:
- name: SMTP
  ip: eva.fit.vutbr.cz
  port: 587
  username: xassat00@stud.fit.vutbr.cz
  password: ***
  from: xassat00@stud.fit.vutbr.cz
  to: xassat00@stud.fit.vutbr.cz
```

Listing 3.2: SMTP input values.

Output example


```
{
    "service": "SMTP",
    "status": "complete",
    "datetime": "2025-02-23T19:13:17.556543",
    "target_host": "eva.fit.vutbr.cz",
    "port": 587,
    "ip_address": "2001:67c:1220:8b0::93e5:b00e",
    "context": {
        "EHLO": "250 eva.fit.vutbr.cz Hello [IPv6:2001:67c:1220:a14:0:0:0:1009], pleased to
            meet you\nENHANCEDSTATUSCODES\nPIPELINING\nEXPN\nVERB\n8BITMIME\nSIZE 33554432\
            nDSN\nAUTH PLAIN LOGIN\nDELIVERBY\nHELP",
        "NOOP": {
            "status": true,
            "smtp_code": "250",
            "smtp_msg": "b'2.0.0 OK'"
        },
        "AUTHENTICATE": {
            "status": true,
            "smtp_code": "235",
            "smtp_msg": "b'2.0.0 OK Authenticated'"
        },
        "SEND EMAIL": {
            "status": true
        },
        "QUIT": {
            "status": true,
            "smtp_code": "221",
            "smtp_msg": "b'2.0.0 eva.fit.vutbr.cz closing connection'"
        }
    },
    "response_time_ms": 254.0
}
```


Listing 3.3: SMTP output values.

3.3.3 SSH test

The `socket` library is used to test the functionality of the SSH service. The Python script is named `ssh_test.py`.The communication process is shown in Figure 3.4.

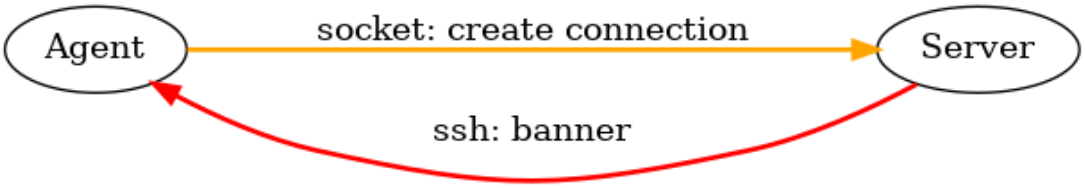

Figure 3.4: SSH test communication process.

This test focuses on checking the availability of the SSH service. When a connection to the server is established, the server sends a banner. The SSH banner contains information about the service version and confirms that the server is ready to process requests. Additional operations are not required for this test, as the main task is to confirm the availability of the service.

Input example

```yaml
services:
- name: SSH
  ip: kazi.fit.vutbr.cz
  port: 22
```

Listing 3.4: SSH input values.

Output example

```json
{
    "service": "SSH",
    "status": "complete",
    "datetime": "2025-02-23T19:13:20.367681",
    "target_host": "kazi.fit.vutbr.cz",
    "port": 22,
    "ip_address": "2001:67c:1220:808::93e5:80c",
    "response_time_ms": 27.0,
    "context": {
        "banner": "SSH-2.0-OpenSSH_9.9-hpn14v15"
    }
}
```

Listing 3.5: SSH output values.

3.3.4 FTP test

The `ftplib` library, which is a built-in Python module, allows the establishment of an FTP connection and provides all the necessary functionality to check the FTP service. The script for this goal is named `ftp_test.py`. The communication process is shown in Figure 3.5.

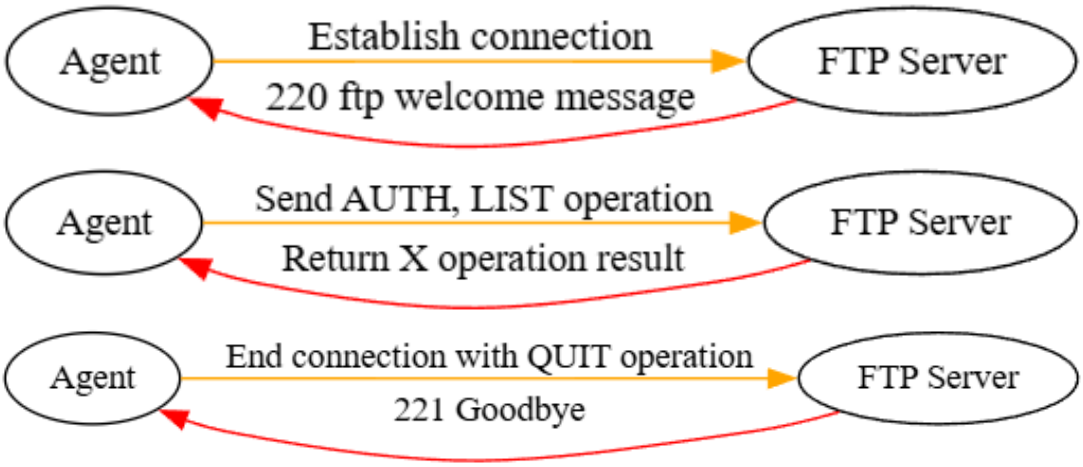

Figure 3.5: FTP test communication process. Sequence from top to bottom.

Input example

```yaml
services:
- name: FTP
  ip: localhost
  port: 21
  username: smtpuser
  password: ***
```

Listing 3.6: FTP input values.

Output example

```
{
    "service": "FTP",
    "status": "complete",
    "datetime": "2025-02-23T19:13:22.681346",
    "target_host": "localhost",
    "port": 21,
    "context": {
        "welcome_msg": "220 pyftpdlib 2.0.1 ready.",
        "login_msg": "230",
        "list_msg": "226",
        "quit_msg": "221"
    }
}
```

Listing 3.7: FTP output values.

3.3.5 Availability test

This test monitors the connectivity of the target and measures latency, jitter, and packet loss using the `icmplib` library. The script is named `ping_test.py`. The communication process is shown in Figure 3.6.

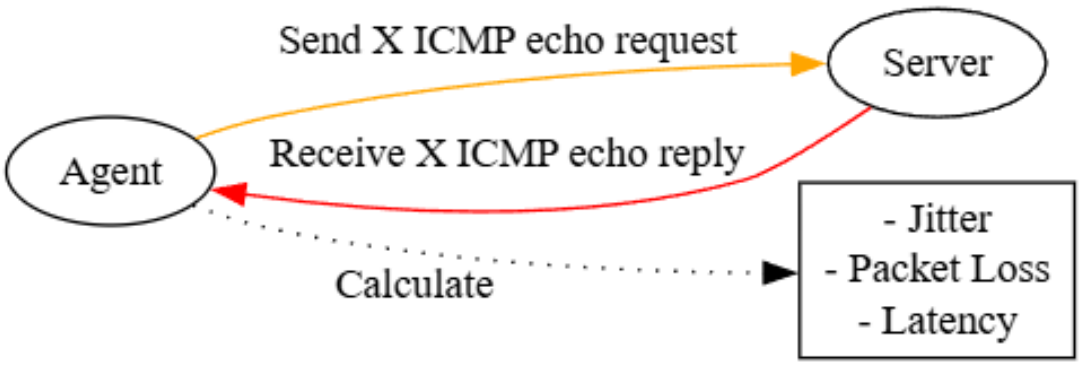

Figure 3.6: ICMP test communication process.

Using the `ping` function of the library `icmplib`, the script sends 5 ICMP echo requests to the target with a 0.2 second interval between each request based on input values defined in Listing 3.8. The timeout for each request is set to 2 seconds. These parameters allow for precise measurement of network responsiveness and reliability.

Input example

```yaml
services:
- name: Ping
  ip: kazi.fit.vutbr.cz
  count: 5
  timeout: 1
  interval: 0.2
```

Listing 3.8: ICMP input values.

Output example

```json
{
    "service": "Connectivity Test",
    "status": "complete",
    "datetime": "2025-02-23T19:13:23.534229",
    "target_host": "kazi.fit.vutbr.cz",
    "ip_address": "147.229.8.12",
    "is_alive": true,
    "latency_avg_ms": 20.584,
    "latency_min_ms": 19.129,
    "latency_max_ms": 21.709,
    "jitter_ms": 1.014,
    "packet_loss": 0.0
}
```

Listing 3.9: ICMP output values.

3.4 Summary

This chapter explained the design of a monitoring system that checks the state of network services using Python scripts and stores the results in JSON format. All tests and their tasks were also described. Experiments was performed to demonstrate the functionality of the tests in action. The next chapter explains the basic concept of LLMs, their differences, and the model that is chosen to improve the current system.

Chapter 4

Large Language Models

This chapter discusses the fundamental concepts of LLM, including its history and architectural characteristics. In addition, the existing models, their classification, and differences are analyzed. The focus is on the difference between open and closed models. Finally, the selected models and the reasons for their choice will be reviewed.

4.1 A brief history of language models

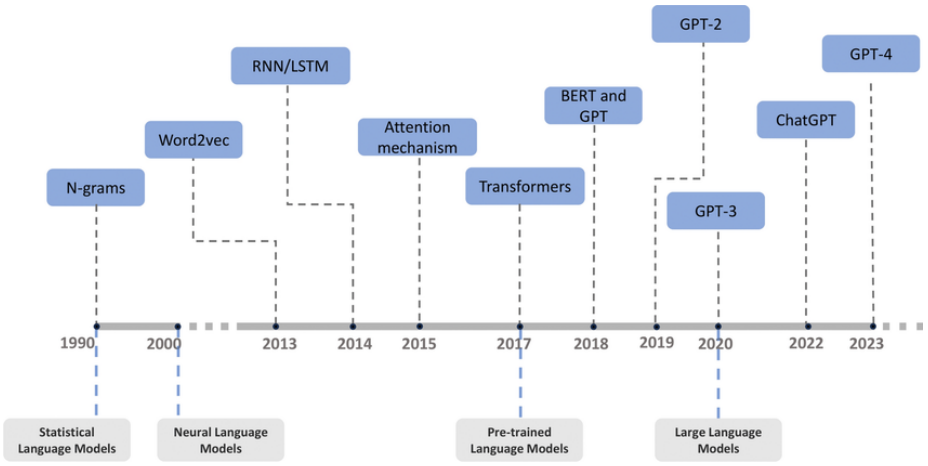

Figure 4.1: History timeline of language models. Image taken [10].

The history of language models began with the first attempts to train computers to understand and generate natural language, back when computing power and data volumes were very limited. As shown in Figure 4.1, in the early 1990s, approaches were based on statistical methods. They determine the likelihood of word occurrences based on examples from text. These models were often shallow and lacked the ability to generalize, which is characteristic of modern approaches [11].

In 2001, Canadian researcher Yoshua Bengio and his colleagues proposed the first *Neural Language Model* (NLM) [12]. They developed a multilayer neural network capable of predicting the next word in a text based on its context. The full version of their work was published in 2003 in the paper *A Neural Probabilistic Language Model* [13], where they showed that neural networks can significantly improve the quality of predictions over tra-

ditional statistical methods. However, such language models do not account for long-range dependencies between words. They could generate grammatically correct text, but it was not very meaningful or did not fit the overall context.

One of the first solutions to the problem of accounting for long-range dependencies in text was the use of *Recurrent Neural Networks* (RNNs). A special place was given to their improved variants, *Long Short-Term Memory* (LSTM), proposed by *Sepp Hochreiter* and *Jürgen Schmidhuber* in 1997 [14]. LSTM blocks can *remember* important information over a long period of time, making them more efficient in sequence processing and providing a basis for building more advanced language models.

A key breakthrough came in 2017 when Google introduced [15] the *Transformers* architecture, a new type of neural network that effectively takes context into account using an attention mechanism. This allowed for more efficient processing of long-range dependencies in text, simplified parallelization of computation, and made it easier to train the model on big data. It is this architecture that formed the basis of modern Large Language Models. The most recognized instances include Google's *Bidirectional Encoder Representations from Transformers* (BERT) [16] and the series of LLM models developed by OpenAI known as *Generative Pre-trained Transformers* (GPT) [17].

4.2 Architecture of transformer

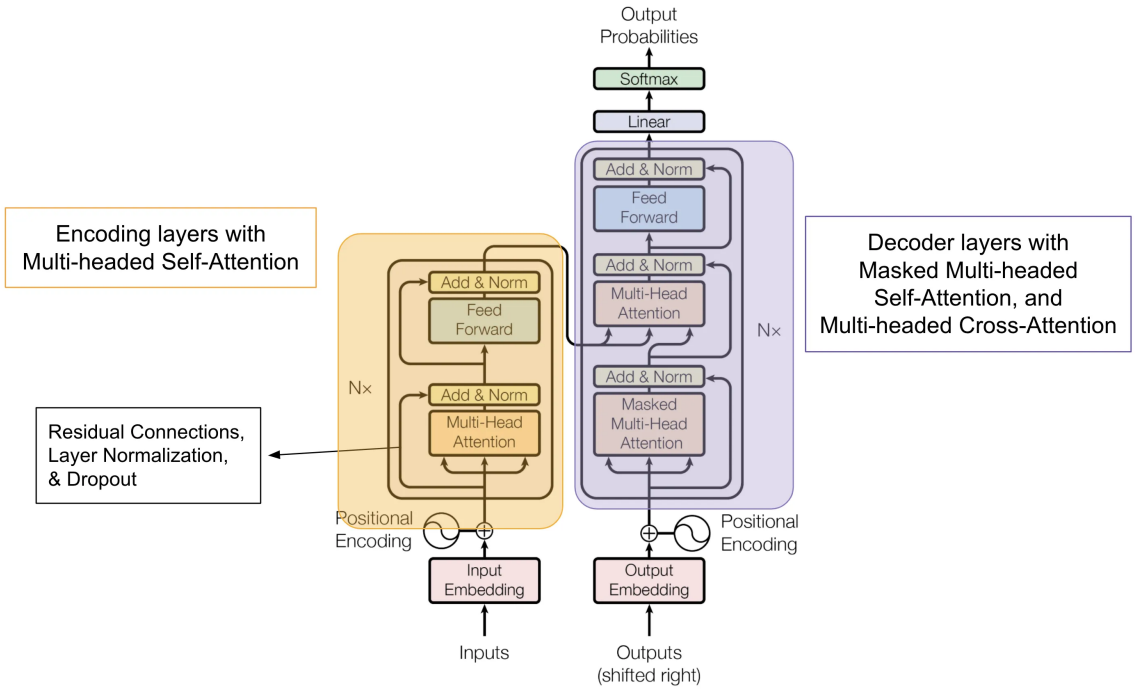

Figure 4.2: Architecture of Transformer. Image taken from [18]

The architecture of the current popular LLMs is based on transformers. The main idea is the mechanism of *self-attention*, which allows each word in a sentence to consider all other words and determine which ones are most important. For this purpose, each word is represented as three vectors:

- Query,
- Key,
- and Value.

The model compares the *query* with the *keys*, calculates importance weights, and applies them to the *values*, allowing it to find relationships between words. However, self-attention is only one part of the architecture. The complete transformer consists of an *encoder* and a *decoder* (Figure 4.2).

The encoder is responsible for processing the input sequence and consists of N layers. Each layer includes:

- Multi-Head Self-Attention: a mechanism that allows the model to consider different aspects of context by applying several attention heads simultaneously.
- Feed-Forward Network (FFN): two linear projections with a nonlinearity (ReLU), applied to each word individually.
- Layer Normalization and Residual Connections: stabilize training and prevent the vanishing gradient problem.

The decoder also consists of N layers, but in addition to the encoder components, it contains:

- Encoder-Decoder Attention: allows the decoder to incorporate information from the input text.
- Masked Multi-Head Self-Attention: prevents the model from looking into the future when generating a sequence.

Since the transformer does not use recurrent or convolutional networks, it adds special *positional encodings* that enable the model to account for the order of words in a sequence. The transformer is trained using *gradient descent* [19] and the *Adam* optimizer [20], while *cross-entropy loss* [21] is used for language tasks.

4.3 Open and closed LLM models

The number of available LLMs has increased significantly in recent years. They vary in size, scope, and availability. Some are designed for scientific research, some are developed for commercial purposes, and others are created specifically for highly specialized tasks. Despite this diversity, all models can be roughly divided into two main categories: open and closed.

Open models are LLMs whose source code and, in some cases, trained weights are freely available. They allow researchers, developers, and organizations to use them without closed model licensing and commercial constraints. Closed models are models whose development and training are conducted by private companies and whose use is governed by commercial terms. Tables 4.1 and 4.2 detail the advantages and disadvantages of open and closed models.

Advantages of Open Models	**Advantages of Closed Models**
Transparency: Open source code allows analyzing algorithms, identifying biases, and improving the model.	*High performance:* Closed models have better accuracy and efficiency because suppliers have more resources for training and optimization.
Flexibility: Models can be further trained on specific data and adapted to particular tasks.	*Support:* Users can expect official support from developers.
Affordability: Using such models is often free or less expensive than commercial solutions.	*Ease of use:* Closed models are often provided as cloud services with a user-friendly interface that allows them to be integrated into applications.
Science advancement: Open models facilitate research, knowledge sharing, and the development of new AI technologies.	*Less security risks:* Because the code is closed, the risks of compromise are reduced.

Table 4.1: Advantages of open and closed models.

Disadvantages of Open Models	**Disadvantages of Closed Models**
Equipment: Large open models often require significant computing power.	*Transparency:* It is difficult to understand how the model works, the data used to train it, and possible biases.
Support: Expert support is not always available, as it depends on community activity rather than official services.	*Limited flexibility:* Usability is determined by the functionality offered by the suppliers.
The boundary of openness: Even open models may have licenses that restrict certain uses or limit access to full source codes or trained weights.	*Cost:* Prices may vary depending on the purpose of use. For example, accessing a model via an API may have a different price than using a paid subscription.

Table 4.2: Disadvantages of open and closed models.

Choosing between an open and a closed model is driven by specific objectives. Open models excel in environments focused on research, experimentation, and customization. In contrast, closed models are the ideal choice for commercial solutions that demand stability, high performance, and reliable support.

In addition, there is another equally important classification: language models are divided into Large Language Models (LLM) and Small Language Models (SLM). They differ not only in the amount of training data, as LLMs are trained on much more significant amounts of data than SLMs. They also differ in model size and complexity, level of context understanding and domain specialization, resource consumption, bias, inference speed, and the types of datasets used [22]. The main aspects are outlined in Table 4.3.

Key	Large Language Models	Small Language Models
Model size and complexity	Have significantly more parameters and complex architectures, allowing them to better capture language nuances.	Have fewer parameters and simpler architectures, making them more efficient for specific tasks.
Contextual understanding	Possess deeper contextual understanding, enabling generation of coherent and logical text.	Limited contextual understanding, which may result in less coherent or accurate output.
Domain specialization	Trained on a wide range of data, making them versatile across domains.	Often specialized in specific domains, increasing efficiency in narrow tasks.
Resource consumption	Require substantial computational resources for training and deployment.	Demand fewer computational resources, making them suitable for resource-constrained devices.
Bias tendency	May reflect societal biases due to training on large-scale internet data.	Less prone to bias due to smaller and more controlled training datasets.
Inference speed	Generally slower due to larger model size and complexity.	Typically faster due to smaller size and simpler architecture.
Types of datasets used	Utilize large-scale and diverse datasets.	Usually trained on smaller, more domain-specific datasets.

Table 4.3: Comparison between Large and Small Language Models

For comparison, one of the most popular LLM, ChatGPT-3, contains 175 billion parameters [23], while, for example, the open source Gemma-2B model from Google's Gemma family includes 2 billion parameters [24]. However, more parameters do not always mean better performance. The quality of a model depends not only on its size but also on its architecture, training corpus, and optimization techniques. Due to their compactness, many SLMs can be further trained on specialized domain-specific datasets to solve specific problems, consuming less computational resources than LLMs.

4.3.1 About privacy and sensitive data

When LLMs are used in monitoring systems, privacy concerns must be addressed, particularly during the analysis of production environment logs. Such logs may contain sensitive information such as internal IP addresses, hostnames, technical errors, file paths, or user-related data. Transferring these data to third-party services, such as cloud LLMs via APIs, potentially carries the risk of information leakage.

To prevent possible leakage of sensitive data, a number of measures can be applied: anonymizing logs in advance (removing or masking IP addresses, hostnames, and other identifiers), using locally deployed LLMs running without connection to external services, and limiting analysis to test or preprepared logs only.

4.4 Chosen models

Three state-of-the-art developments have been selected in the field of Large Language Models: Gemini, LLaMA, and the Nous Hermes 2 on Mistral platform. A comparative charac-

terization of each model is shown in Table 4.4. All data about the characteristics of LLaMa and Nous Hermes were taken from the popular source LLM Explorer [25]. Gemini closed model data was obtained from the official Google Cloud pages [26].

Characteristic	**Gemini 2.0 Flash**	**Meta Llama 3 8B Instruct**	**Nous Hermes 2 Mistral 7B DPO**
Required VRAM	Uknown (API-based)	16.1 GB	14.4 GB
Maintainer	Google DeepMind	meta-llama	NousResearch
Context Length	1 million (Flash)	8,192	32,768
Tokenizer Class	Proprietary	PreTrainedTokenizerFast	LlamaTokenizer
License	Proprietary	llama3	apache-2.0
Model Size	Unknown	8B	7B
Model Architecture	Custom (Gemini)	LlamaForCausalLM	MistralForCausalLM

Table 4.4: Comparison of LLM Models

4.4.1 Gemini

Gemini is a large proprietary model developed by Google DeepMind. The main reason for its selection is the availability of a free level using API, which makes it convenient for experimental use within the thesis [27]. In addition, Gemini is a powerful and modern model, close in quality to ChatGPT, which makes it an ideal representative of a closed-source LLM model.

4.4.2 LLaMa

LLaMA (Large Language Model Meta AI) is an openly available model developed by a large technology corporation Meta. In this thesis, the GGUF format version of LLaMA was used, which allows the model to run efficiently on regular CPU computers, without the need for powerful graphics [28]. This makes it accessible for local experiments and provides full control over data processing.

4.4.3 Nous Hermes 2

Nous Hermes 2 is an open source model based on the Mistral architecture and developed by the independent research group Nous Research[1]. As in the case of LLaMA, a GGUF-formatted version is used to run locally without a GPU [29]. The choice of this model is due to the interest in alternatives from small independent developers, allowing to compare their approaches with solutions from large companies.

4.5 Summary

This chapter provided a brief history of language models, reviewed the concept of Transformers architecture, analyzed the key differences between open and closed models, and provided a reason for choosing specific models. The choice of models includes:

[1] https://nousresearch.com

- One closed source cloud-based model (Gemini).
- Two open source local models in GGUF format (LLaMA and Nous Hermes).
- One model from a corporate source (Meta) and one from an independent group (Nous Research).

The next chapter will discuss the tools used to enhance the explainability of network logs.

Chapter 5

Explainable Diagnostics

In this chapter, the main idea of this paper will be discussed. The concept of explainable diagnostic will be defined, its necessity will be explained, and an overview of the tools previously used for this purpose will be given. In addition, the updated design of the monitoring system (Chapter 3) will be presented, the tools used for it will be described, the input data used will be discussed and the final result will be demonstrated.

5.1 What is explainability?

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.

Modern diagnostic techniques strive for greater interpretability, allowing automated systems not just to identify failures, but also to explain them in an understandable way. This is particularly important in large network infrastructures, where the complexity of services makes it difficult to quickly analyze problems. The use of Large Language Models opens up new possibilities for explainable diagnostics, as such models are able to analyze data, identify patterns, and formulate conclusions in an easy-to-understand format.

5.1.1 Methods used for diagnostics

A monitoring system is a tool whose primary task is to collect and record data on the state of a network, services, or equipment. When certain events occur, such as failures or test runs, the results are recorded as structured data, for example, logs containing metrics. This process can be implemented using both passive methods (such as NetFlow) and active approaches, described in Chapter 3 of this thesis. Despite the differences in approaches, all systems share the same goal, obtaining suitable structured data for further analysis.

However, this data itself does not possess independent diagnostic value. Even if the monitoring system records an incident or anomaly, the subsequent analysis and decision making require the participation of an expert. Only someone with the appropriate qualifications can correctly interpret metrics, logs, and other fields.

Using LLM allows logs to be analyzed in a more flexible format, identifying the fact of an error and potential root causes. This is achieved through the ability of models to interpret data and build logical relationships. The application of explainable diagnostics in network monitoring involves three key steps:

- Log analysis: processing of data received from network services.
- Event classification: determining whether an event is normal or error-related.
- Generating an explanation: describing the cause of the error and recommendations for recovery.

5.1.2 Application LLM to diagnostics

To get the most structured and accurate answers from LLM, the *prompt engineering* method is used [30]. This approach consists in generating special queries (prompts) that guide the model to the desired format and content of the response. The key elements of a successful query are the precision, logical meaning, consistency, and completeness of the description. The structure also plays an important role: for example, in the use of lists, clarifying questions, and examples. There are other approaches for this solution: fine-tuning and Retrieval-Augmented Generation (RAG) [31].

Fine-tuning is used when the model initially does not understand the task format or demonstrates a poor quality of response to domain-specific tasks [32]. However, this approach requires significant computational resources, an extensive training corpus, and a deep understanding of the model architecture. In addition, if the structure of the input data (e.g. log format) changes, the model will need to be trained again. This reduces the flexibility of the system and increases the cost of maintaining it. In the context of this thesis, where the structure of logs may change (e.g. when switching from JSON to YAML), this approach is redundant and impractical.

An alternative is the RAG (Retrieval-Augmented Generation) method, where the model uses an external data store (e.g., a log or external database) to generate a response [33]. This approach allows combining the LLM model with up-to-date information from external sources. However, implementing an RAG system requires the building of a storage, indexing, and retrieval infrastructure, the establishment of an embedding mechanism, and the linking of the resulting data to generation, which makes it very complex.

Due to the above, prompt engineering was chosen in this thesis as the most appropriate, flexible, and easy-to-implement way to interact with the LLM, allowing for the achievement of structured and accurate answers without the need to modify the model itself.

A well-formulated prompt allows the right results from the model to be obtained faster and more accurately. The better the query is formulated, the fewer iterations are required to refine the answer, thus reducing computational resources and time. With prompt engineering, it is possible to *train on the fly* or adapt the model to solve specific problems without changing the model architecture.

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 be also provided. Based on this prompt, the model analyzes the input data and generates a response. The more precise and detailed the query, the higher the probability that the answer will fully solve the problem.

Proper wording helps the model focus on the essence of the query, avoiding ambiguities and unwanted interpretations. The better the model understands the problem, the more valuable and accurate the answer will be.

Modern LLM models, such as Gemini, LLaMA, and Nous Hermes 2, are highly generative in their response. The same query formulated slightly differently or identically can lead to differently shaped and structured answers. Without clear instructions, the model can produce vague, insufficiently detailed, or redundant answers, making it difficult to further process the data. Prompt engineering solves this problem by giving the model a strict response structure.

When analyzing error logs, it is common to see error codes and short messages of 2-3 words relating 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 server logs and indicate different types of issues that need to be addressed. However, such messages do not always give a clear picture of what the error refers to, what it means, and what caused it. Error codes and brief messages often require further investigation to understand the underlying issue. To eliminate this uncertainty and make diagnostics more understandable, the prompt should be generated with the following requirements in mind:

- 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).

This approach ensures the predictability of responses, reduces their variability, and preserves the specified structure. This is especially important when analyzing logs, as it provides the same output format, which simplifies subsequent processing, automated analysis, and data aggregation.

5.2 Design and implementation of diagnostic system

The design of the explainable diagnostic system is shown in Figure 5.1. The main script responsible for generating and analyzing log files is called `diagnosis_system.py`. At the beginning, it loads the necessary configuration files from `diagnostic_config.yaml`, which specifies information about which *model* to use, as well as the *location of the folder* with log files.

The glob library is used to search for all files with the `.json` extension in the specified directory. Then using the `os.path.getctime` function, which returns the file creation time, the file with the latest creation date is determined. Thus, the most recent file is automatically selected from the list of logs and used for analysis. An example of a configuration file is shown in Listing 5.1. In this case, the *Gemini* model will analyze the last JSON file stored in the `./logs/` folder.

```
model_config:
  model: gemini # can be gemini, mistral, or llama
  log_dir: "./logs/"
```

Listing 5.1: Diagnostic configuration file.

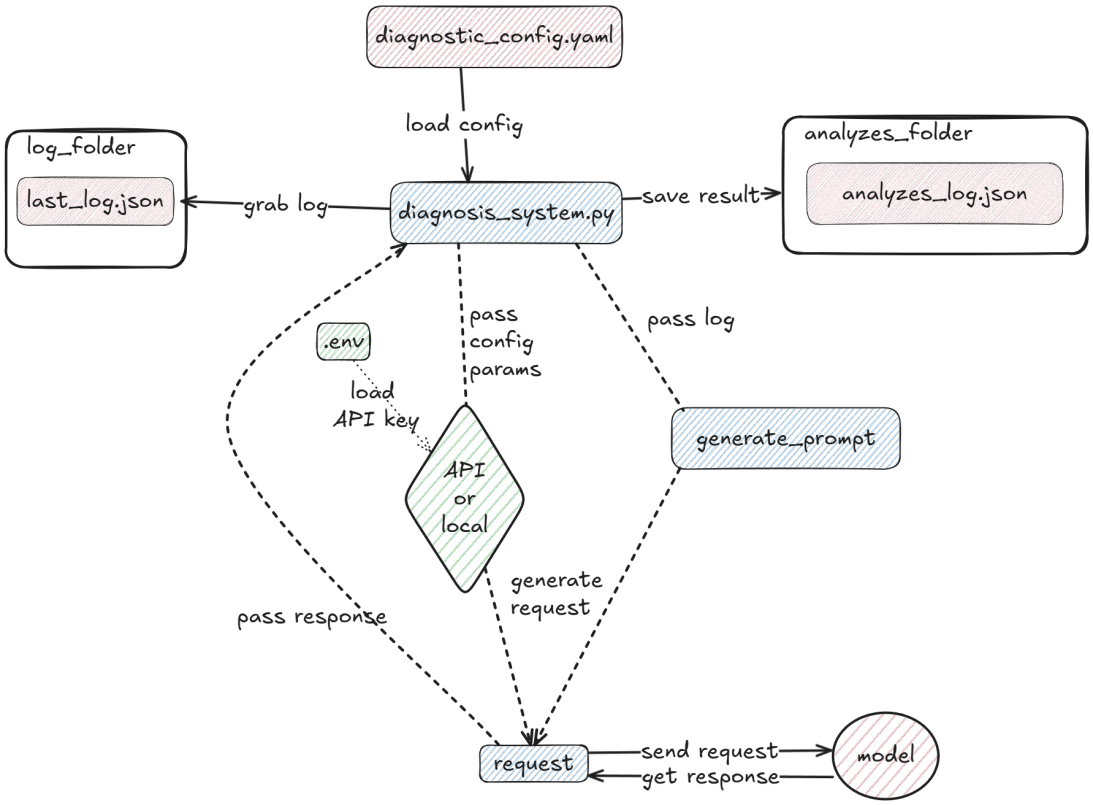

Figure 5.1: The explainable diagnostic system design.

Before passing the log file for analysis, a preliminary check of its contents is performed. If the log file contains at least one record with the service status `error`, such a log is considered potentially problematic and is sent to LLM for analysis. Otherwise, the file is classified as correct and not analyzed because there are no signs of errors. Listing 5.2 presents an example of a log file that requires analysis and Listing 5.3 that does not require analysis.

```
[
    {
        "service": "SMTP", # record 1
        "status": "complete",
        <omitted>
    },
    {
        "service": "SSH", # record 2
        "status": "error", # this log file needs to be analyzed!
        <omitted>
    },
    <omitted>
]
```

Listing 5.2: Incorrect log example.

Once the log file and the model to be used to analyze it have been defined, a specially created prompt with clear instructions is generated. This prompt is shown in the Listing 5.6. It was based on the previous Section 5.1.2, which discussed the requirements for the

responses generated by the LLM. The main focus was to ensure consistency in the model responses in order to avoid unnecessary variability and to achieve stability in the wording. Such formulation clearly defines the role of the model, which will prevent it from focusing on other areas. In addition, by structuring the request, the model response will be consistent and easy to understand. Clear bullet points with expected response formats help avoid ambiguity and ensure that information is presented in logical order. The defined example classifications avoid overly generic categories such as `Error` or `Problem`, and the instruction to contact the administrator in case of doubt adds a level of reliability.

```
[
    {
        "service": "SMTP", # record 1
        "status": "complete",
        <omitted>
    },
    {
        "service": "SSH", # record 2
        "status": "complete", # regular log, skip analysis process
        <omitted>
    },
    <omitted>
]
```

Listing 5.3: Correct log example.

The log file to be analyzed does not undergo any pre-processing or transformation: its contents are inserted *unchanged* into the prompt passed to the LLM. This approach preserves the original context and structure of the data, which is particularly important for detailed analyzes.

The prompt with the log is then passed to the selected model to be analyzed. If *Gemini* is used as the model, the interaction takes place through the API. The API key required for authentication is retrieved from a `.env` file using the `dotenv` library, which ensures secure and convenient access to sensitive configuration data. 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 to work with local LLMs developed by Nomic AI [34]. It provides a user-friendly interface to download, run, and interact with LLMs without cloud access. In this implementation, it uses the method for deploying LLMs using the Python SDK as described in the documentation [35].

It should be noted that the structure of a prompt with logs depends on the type of model used, whether local or API-based. When interacting with the Gemini model, which has high computational power and the ability to analyze data in a structured way, it is possible to send the entire log with a single request. In this case, the model will process each log item separately but present the results within a single response, keeping the same format for all records.

However, difficulties may arise when working with local models due to excessive generalization of information. For example, if a log file contains three correct and one incorrect records (see Listing 5.2 as an example), the model may produce a single result for all log records, meaning it does not provide an explanation for each individual record, but rather gives one response for all, potentially missing the erroneous log. To avoid distortion, another method is used: each log record is added to the prompt separately, so as many queries as there are records in the log are executed. This method allows the model to analyze each

record separately, reducing the risk of ignoring errors and making conclusions more accurate and transparent.

The following parameters are configured when sending a request to the local LLM: `max_tokens`, `temperature`, `top_k`, `top_p`, and `streaming` (Listing 5.4).

```
with model.chat_session():
    response = model.generate(
        prompt=prompt_data,
        max_tokens=1024,
        temp=0.2,
        top_k=10,
        top_p=0.9,
        streaming=False,
        )
```

Listing 5.4: Local LLM parameter values.

1. The `max_tokens` parameter limits the maximum number of tokens that the model can generate. A value of 1024 ensures that the model has enough capacity to provide detailed and structured explanations without prematurely truncating the generated response. Increasing this value further was unnecessary, as responses typically never approached this limit, and higher values would increase computational overhead without added benefit.

2. `Temperature` controls the level of randomness in the response of the model. The higher the value, the more creative and unpredictable the answers become. In this case, the parameter is set to 0.2, since the goal of this system is to produce reliable and repeatable diagnostic explanations, low randomness is desirable.

3. `top_k` limits the number of most likely word choices that the model can select at each generation step. Setting this to 10 means that the model only considers the ten most likely tokens at each step, significantly reducing the risk of irrelevant or unexpected token selection, while still allowing slight variation to avoid overly repetitive answers.

4. `top_p` (also known as nucleus sampling [36]) controls which words will be used in the answer, selecting only those whose total probability is 90%. This balances accuracy and diversity, allowing the model to account for rare but meaningful words if they logically fit into the context.

5. The `streaming` parameter controls whether responses are returned incrementally (word-by-word) or as a complete output. For the analysis of structured logs, returning a complete output (`False`) is preferable because the log analysis requires consistent and uninterrupted text structures.

According to the source cited [37], the *Gemini* model *temperature* parameter default value is set to `1.0`. The allowed range for this parameter is between `0.0` and `2.0`. If you try to set a value higher than this threshold (for example, `2.1`), the model will return an error. At the same time, local models run through GPT4All do not have clearly defined limits for this parameter. It is possible to specify temperature values to `10`, `100`, or `1000`. The other parameters remain the same as in Listing 5.4 and have not changed significantly. However, in the case of Gemini, these parameters are configured differently, as shown in Listing 5.5.

After receiving a response from the model, the result is saved in the `analyzes` folder in the format `analysis_log_datetime.json`, with the name of the original log file retained in the name. This allows for easy comparison of the analysis with the original log file in the future. The list of newly added libraries is described in Table 5.1.

```
# Initialize the Generative AI client
google.generativeai.configure(api_key=google_api)
# Set the model
model = google.generativeai.GenerativeModel('gemini-2.0-flash')
# Set the config params
config = google.generativeai.GenerationConfig(
    temperature=0.2,
    top_k=10,
    top_p=0.9,
    max_output_tokens=1024,
)

response = model.generate_content(
    contents=prompt,
    generation_config=config
)
```

Listing 5.5: Gemini LLM parameter values.

Library	Purpose
gpt4all	Provides access to GPT4All models for local execution of LLMs.
google.generativeai	Used to interact with Google's Gemini AI via API.
glob	Handles searching for log files based on patterns.
os	Provides functionalities for working with the operating system, such as file and directory management.
re	For parsing the LLM response to make it easier to extract data.
dotenv	Loads environment variables from a `.env` file to securely manage API keys and configuration parameters.

Table 5.1: Required Python libraries.

```
prompt = f"""
You are a system designed to diagnose and explain network log files.
{log}

Please analyze the provided log and respond with the following information (Do not use
    any markdown, bold text. Stick to this format):
1. Explanation: Provide a description of the event or action in the log.
2. Classification: Classify the issue (It should be a meaningful name. Do not just
    label 'Error', 'Issue' etc. If there is no issue, then classify as Normal Operation
    ).
3. Error Reason: Identify and describe the root cause of any error detected (If it is
    classified as 'Normal Operation', then write 'No error detected').
4. Recommendation: Suggest any actions or solutions based on the diagnosis (If it is
    classified as 'Normal Operation', then write 'No action required').

If you are not sure with the diagnosis, please consult with the system administrator.
"""
```

Listing 5.6: Network Log Analysis Prompt.

Expected output

```
[
    {
        "1. Explanation": "log explanation...",
        "2. Classification": "log explanation...",
        "3. Error Reason": "log explanation...",
        "4. Recommendation": "log explanation..."
        "service": "service name"
    },
    {
        "1. Explanation": "log explanation...",
        "2. Classification": "log explanation...",
        "3. Error Reason": "log explanation...",
        "4. Recommendation": "log explanation..."
        "service": "service name"
    },
    <omitted>
]
```

Listing 5.7: Script output file example.

5.3 Summary

This chapter explores the concept of explainability, highlighting its importance and relevance in today's environment. It discusses the principles of using Large Language Models (LLMs) to analyze network issues and methods for obtaining structured, interpretable explanations of logs. A system design is presented that builds on the system described in Chapter 3, and the implementation of key components is demonstrated. In the next chapter, the developed system will be tested in various scenarios to evaluate its functionality and effectiveness.

Chapter 6

Evaluation of scenarios with explainable diagnostics

This chapter presents experiments to evaluate the functionality and accuracy of the network diagnosis system using LLM. The focus is on assessing the ability of LLMs to correctly interpret network log records and generate explainable diagnostic outputs. The chapter outlines the test scenarios, metrics, and methods used to evaluate the quality, stability, and accuracy of the model responses.

6.1 Evaluation metrics

A methodology based on expert evaluation was used to assess the quality of responses generated by the LLM models. The main 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.

For this purpose, a special survey was created using Google Forms, consisting of several sections corresponding to experimental scenarios (a detailed description of the experimental scenarios is given in the following section). Participants were asked to provide answers generated by different LLMs (Table 4.4) to the same log files. Based on these data, they were asked to answer questions about the usefulness, precision, and stability of the responses. Each section of the survey corresponded to one of the experimental scenarios:

- *Scenario one:* Participants were asked to determine whether the response was helpful and whether the model correctly interpreted the log.
- *Scenario two:* Participants were asked to choose the value at which the responses seemed the most accurate and at which value they started to *hallucinate* when the value of the temperature parameter in LLM changed.
- *Scenario three:* Participants were asked how much the responses to the same log differed to verify consistency.

In addition, participants chose which model they thought performed better in each scenario. This approach allows for the quantification of results and the receiving of live qualitative feedback.

6.2 Scenarios

The experimental part involved the development of three separate experimental scenarios, each designed to assess different aspects of LLM model performance in network log analysis. The pre-generated correct (Listing 5.3) and incorrect (Listing 5.2) logs were used in all scenarios, covering two network services: ICMP and SMTP.

Although in the implemented system (Chapter 5), correct logs are not sent for analysis (if there are no errors, the file is skipped), such logs were intentionally included in the experiments. This was done to test how accurately LLM models can interpret normal situations and avoid false positives. Thus, the experimental scenarios cover incorrect and correct records, which is necessary to fully evaluate the behavior of the models.

6.2.1 Scenario one: Interpretation of the network log

This scenario aims to evaluate how accurately LLM models (Table 4.4) interpret logs with different operation execution statuses. Three types of logs were prepared for each of the two services (SMTP and ICMP):

- *Correct log:* reflecting the successful operation of the service;
- *Incorrect log:* containing an incorrect operation (e.g., connection denial);
- *Correct, but with remarks:* formally containing no error but raising doubts (e.g., suspiciously high latency).

Thus, each model will analyze three logs, and the quality of the explanation, classification, reason for the error, and recommendations for each of them were evaluated.

6.2.2 Scenario two: Temperature effect on model behavior

The temperature parameter directly influences the variability and creativity of the LLM models output. In this case, the identical log (which includes an error in the SMTP and ICMP services) was sent to the LLM model with various temperature settings: `0.4`, `0.6`, `0.8`, `1.0` and `10.0`. The purpose of the experiment is to find out:

- How does changing the value of `temperature` affect the accuracy and usefulness of the response?
- At what value do models begin to show signs of *hallucination*, adding fictitious, superfluous, or irrelevant details?
- Does the logical coherence and structure of the response persist as the temperature changes?

It is important to remember that Gemini has a maximum temperature value of `2.0` instead of the last value of `10.0`, as with local models, Gemini will use `2.0`. The previous chapter shows how the temperature parameter is configured in Listing 5.4 for local LLM models and Listing 5.5 for the Gemini model.

6.2.3 Scenario three: Consistency of responses

This scenario aims to test how stable the responses of the LLM models are, and the same log was sent to each model five times in a row to see if the responses would be the same or change. The responses received were compared in terms of the following:

- How similar or different the responses are in terms of wording, meaning, and structure?
- Whether any contradictions, hallucinations, or classification inconsistencies appear between repetitions.

6.3 Results and analysis

A total of 13 people participated in the survey, providing feedback on all experimental scenarios. During the execution of the experimental scenarios, 39 runs of the system were made: 13 queries for each of the models - Gemini, Nous Hermes 2 and LLaMA 2. From the results (Table 6.1), the Gemini model significantly outperforms local solutions in terms of response time, while local models were run on a regular laptop without graphics acceleration, which significantly affects processing time.

Model	Average Time (s)
Gemini	5.019
LLaMa	275.535
Nous Hermes 2	298.101

Table 6.1: Average response time of the model over 13 runs.

The models were provided with three types of logs: a correct log (Listing 6.1), a log with increased delay (Listing 6.3), and a log with an error (Listing 6.2).

```
[
    {
        "service": "SMTP",
        "status": "complete",
        "datetime": "2025-03-29T19:32:42.538143",
        "target_host": "example.smtp.server",
        "port": 587,
        "ip_address": "192.168.0.100",
        "context": {
            "EHLO": "250 mail.example.local....",
            "NOOP": {
                "status": true,
                "smtp_code": "250",
                "smtp_msg": "b'2.0.0 Ok'"
            },
            <omitted>
        },
        "response_time_ms": 58.0
    },
    {
        "service": "Connectivity Test",
        "status": "complete",
        "datetime": "2025-03-29T19:32:42.807867",
        "target_host": "localhost",
        "ip_address": "127.0.0.1",
        "is_alive": true,
        "latency_avg_ms": 0.06,
        "latency_min_ms": 0.053,
        "latency_max_ms": 0.069,
        "jitter_ms": 0.007,
        "packet_loss": 0.0
    }
]
```

Listing 6.1: Correct log.

```
[
    {
        "service": "SMTP",
        "status": "complete",
        "datetime": "2025-03-29T19:37:20.325503",
        "target_host": "example.smtp.server",
        "port": 587,
        "ip_address": "192.168.0.100",
        "context": {
            "EHLO": "250 mail.example.local....",
            "NOOP": {
                "status": true,
                "smtp_code": "250",
                "smtp_msg": "b'2.0.0 Ok'"
            },
            <omitted>
        },
        "response_time_ms": 613256.0
    },
    {
        "service": "Connectivity Test",
        "status": "complete",
        "datetime": "2025-03-13T21:05:53.710873",
        "ip_address": "192.168.0.111",
        "is_alive": true,
        "latency_avg_ms": 150.342,
        "latency_min_ms": 25.134,
        "latency_max_ms": 620.987,
        "jitter_ms": 145.872,
        "packet_loss": 22.5
    }
]
```


Listing 6.2: Log with high delay.


```
[
    {
        "service": "SMTP",
        "status": "error",
        "datetime": "2025-03-29T19:33:35.848968",
        "target_host": "example.smtp.server",
        "port": 587,
        "ip_address": "192.168.0.100",
        "context": {
            "EHLO": "250 mail.example.local...",
            "NOOP": {
                "status": true,
                "smtp_code": "250",
                "smtp_msg": "b'2.0.0 OK'"
            },
            "AUTHENTICATE": {
                "status": false,
                "error_msg": "(535, b'5.7.0 authentication failed')"
            }
        },
        "response_time_ms": 223.0,
        "error_msg": "(535, b'5.7.0 authentication failed')"
    },
    {
        "service": "Connectivity Test",
        "status": "error",
        "datetime": "2025-03-29T19:37:20.558907",
        "target_host": "192.168.0.111",
        "is_alive": false,
        "error_msg": "Host is not reachable"
    }
]
```


Listing 6.3: Log with the error.

Since the number of model responses was quite large, only one or two of the most representative examples are included in the main part of each scenario. The complete responses from all models for all log files can be found in the Appendix A.

6.3.1 Scenario one results

An example of how Gemini responds to a log containing errors (Listing 6.3) is shown in Listing 6.4.

```
[
    {
        "1. Explanation": "An attempt to authenticate with the SMTP server failed.",
        "2. Classification": "SMTP Authentication Failure",
        "3. Error Reason": "The SMTP server rejected the authentication attempt, likely due to incorrect
             username or password.",
        "4. Recommendation": "Verify the SMTP username and password. Ensure that the account is not locked or
              disabled. Check the SMTP server's logs for further details.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "A connectivity test to host 192.168.0.111 failed.",
        "2. Classification": "Host Unreachable",
        "3. Error Reason": "The host 192.168.0.111 is either down, unreachable due to network issues, or a
             firewall is blocking the connection.",
        "4. Recommendation": "Verify that the host 192.168.0.111 is powered on and connected to the network.
             Check network connectivity (e.g., ping) to the host. Investigate any firewall rules that might
             be blocking the connection.",
        "service": "Connectivity Test"
    }
]
```

Listing 6.4: Gemini error log response.

The response is structured, follows the format specified in the prompt, and contains a clear explanation of the causes of errors. The survey respondents rated the usefulness of each response and indicated which model, in their opinion, interpreted the log more accurately.

Correct log analysis result

Participants were given a correct log with the question *Does this explanation correctly reflect the log data?* They were asked to choose one of the following answers:

- Yes, the explanation is accurate.
- Partially correct, but some details are off.
- No, the explanation is incorrect or misleading.

The results of the survey on the correct log are listed in Table 6.2.

Model	**SMTP**			**ICMP**		
	Yes	**Partial**	**No**	**Yes**	**Partial**	**No**
Gemini	92.3 %	7.7 %	0 %	84.6 %	15.4 %	0 %
Nous Hermes 2	100 %	0 %	0 %	100 %	0 %	0 %
LLaMA	69.2 %	30.8 %	0 %	76.9 %	23.1 %	0 %

Table 6.2: Results on normal logs.

All three models performed well with normal logs, and the Nous Hermes 2 model performed perfectly (100%). Gemini was also perceived well, but LLaMA shows less confidence in interpretation.

Error log analysis result

Participants were asked exactly the same question as in the previous scenario. The results of each model on the error log are presented in Table 6.3.

Model	SMTP			ICMP		
	Yes	**Partial**	**No**	**Yes**	**Partial**	**No**
Gemini	69.2 %	30.8 %	0 %	69.2 %	30.8 %	0 %
Nous Hermes 2	92.3 %	7.7 %	0 %	76.9 %	15.4 %	7.7 %
LLaMA	84.6 %	15.4 %	0 %	69.2 %	30.8 %	0 %

Table 6.3: Results on error logs.

Nous Hermes 2 shows the highest accuracy in detecting and explaining clear errors in the SMTP service (92.3%), although in the ICMP case the first disagreement cases occurred. Gemini and LLaMA show comparable results with slight variations in accuracy.

The participants were also asked an extra question: *How helpful is this explanation to diagnose the problem?* The purpose of this question was to assess how helpful the explanation was to identify the cause of the error. Participants were asked to choose from a range of ranges corresponding to the degree of usefulness.

- 0-20% - Not helpful at all
- 21-40% - Slightly helpful
- 41-60% - Somewhat helpful
- 61-80% - Mostly helpful
- 81-100% - Very helpful and clear.

The results of the responses of each model are presented in Table 6.4 for SMTP and in Table 6.5 for ICMP.

Model	**0–20%**	**21–40%**	**41–60%**	**61–80%**	**81–100%**
Gemini	0%	7.7%	15.4%	76.9%	0%
Nous Hermes 2	0%	0%	23.1%	53.8%	23.1%
LLaMA	0%	0%	23.1%	76.9%	0%

Table 6.4: Helpfulness of explanations for SMTP logs.

Model	**0–20%**	**21–40%**	**41–60%**	**61–80%**	**81–100%**
Gemini	0%	15.4%	23.1%	53.8%	7.7%
Nous Hermes 2	0%	0%	23.1%	53.8%	23.1%
LLaMA	0%	23.1%	30.8%	30.8%	15.4%

Table 6.5: Helpfulness of explanations for ICMP logs

The participants generally found the explanations provided by the three models to help explain the SMTP and ICMP logs. Gemini and LLaMA received the highest ratings in the *Mostly helpful* category for SMTP logs. For ICMP logs, Gemini and Nous Hermes 2 were

rated highest in the *Mostly helpful* category, while LLaMA helpfulness was more distributed across the *Somewhat helpful* and *Mostly helpful* categories. Participants considered none of the models unhelpful for either type of log.

Overall, the results show that all models successfully identify and explain errors in the SMTP and ICMP logs. In particular, SMTP logs often contain standard response codes (for example, 220, 221, 235, 250) and well established message such as '*Authentication failed*'. Such messages are easy to interpret because they have a predictable structure and clear diagnostic features that do not require a deep context. Such patterns allow models to effectively recognize and explain the problem, especially if the log has a clear structure and is correctly formulated in the prompt.

As a result, models are more likely to generate clear and logical explanations, although in the form of generic recommendations such as '*check login and password*' or '*make sure the server is available*'. Despite being a formulaic answer, such advice often turns out to be useful and helps solve the problem.

High delay log analysis result

To ensure uniformity of evaluation, the third scenario used the same question as before - similar to scenarios 1 and 2 for the high latency log.

Model	**SMTP**			**ICMP**		
	Yes	**Partial**	**No**	**Yes**	**Partial**	**No**
Gemini	76.9 %	15.4 %	7.7 %	76.9 %	23.1 %	0 %
Nous Hermes 2	15.4 %	61.5 %	23.1 %	23.1 %	53.8 %	23.1 %
LLaMA	15.4 %	53.8 %	30.8 %	7.7 %	53.8 %	38.5 %

Table 6.6: Results on high delay logs.

Table 6.6 indicates that Gemini achieves the highest accuracy (76.9%) in scenarios with high log response latency. In contrast, Nous Hermes 2 and LLaMA perform significantly worse, often producing partial and incorrect responses.

In evaluating the outcome, Gemini demonstrated the best result for correctly identifying the abnormally high `response_time_ms` value as a potential problem. Although the log shows a successful operation, a delay of a few tens of seconds (or even a minute) in the SMTP service and packet losses in the ICMP service are atypical and may indicate a performance degradation.

It is likely that LLaMA and Nous Hermes 2 concentrated exclusively on the '*completed*' status. They probably did not regard an extension in response time as an indication of failure. Gemini demonstrated a deeper understanding of metrics and context. This may be due to both the quality of the training data and the architectural features of the model. For example, Gemini not only analyzes the log structure, but can also interpret metrics in terms of network norms, which is especially valuable in boundary situations.

Participant LLM response preference

At the end of each log-type section, participants were asked to select which model they felt did the best job. The results of these choices are summarized in Table 6.7.

Scenario	Gemini	Nous Hermes 2	LLaMA	All of them	Other
Correct log	7.7%	38.5%	15.4%	38.5%	0%
Incorrect log	7.7%	53.8%	15.4%	7.7%	15.4%
High delay log	76.9%	23.1%	0%	0%	0%

Table 6.7: Preferred model per scenario.

Based on participant preferences, Gemini leads significantly in handling edge cases (76.9%), far surpassing Nous Hermes 2 (23.1%) and LLaMA 2 (0%). For normal cases, the participant chose that all models showed the same preference (38.5%), while Nous Hermes 2 is clearly preferred for erroneous cases (53.8%).

6.3.2 Scenario two results

Below is an example of the response of the Nous Hermes 2 model with parameter values of 0.2 (Listing 6.5) and 10.0 (Listing 6.6).

```
1
2  "1. Explanation": "The log indicates an SMTP connection attempt to example.smtp.server on port 587 from IP
       address 192.168.0.100 at 2025-03-29T19:33:35.848968. The server responded with an error message '(535,
       b'5.7.0 authentication failed')' indicating that the client was unable to authenticate successfully
       during the SMTP session.",
3
4  "2. Classification": "Authentication Failure",
5
6  "3. Error Reason": "The root cause of this issue is likely a failure in providing valid login credentials or
       an incorrect configuration for the AUTHENTICATE command. Without further information, it's difficult
       to determine if there was a network interruption during authentication or if the client used invalid
       credentials.",
7
8  "4. Recommendation": "The user should check their SMTP server settings and ensure that they are using valid
       login credentials. If this issue persists, consult with your system administrator for further
       assistance. No action required if it is a one-time occurrence or the client has successfully
       authenticated in subsequent attempts.",
```

Listing 6.5: NH2 SMTP error log response `temp=0.2`.

```
1
2  "1. Explanation": "In this log, there was an attempt to establish a SMTP connection with example.smtp.server
       on port 587 using ip_address 192.168.0.100. The authentication failed and returned error '(535, b
       '5.7.0 authentication failed')'.",
3
4  "2. Classification": "Error",
5
6  "3. Error Reason": "The reason for this issue is that the provided credentials used during the AUTHENTICATE
       phase were either incorrect or invalid, causing the SMTP server to reject the connection. It can also
       indicate misconfigured settings on client side which led to using an invalid set of credential pairs
       for SMTP authentication.",
7
8  "4. Recommendation": "To resolve this error, verify and confirm the correctness of credentials being used in
       the AUTHENTICATE phase during SMTP connection with example.smtp.server on port 587. Double-check that
       these login credentials are valid by referring to any system or user documentation related to them.
       Also check if client side configuration has proper credential sets for this SMTP server and adjust it
       as required, then retry the SMTP connection with the correct credentials. If you're still facing issue
       consult your System Administrator.",
```

Listing 6.6: NH2 SMTP error log response `temp=10.0`.

Differences can be observed across all fields. Although both responses contain the necessary information in the `Explanation` field, the first provides a more detailed description of the event timestamp and the server response structure. In `Classification`, the second response states *Error*, which is a generic category lacking specificity, unlike the first response that provides a more precise classification of the problem. In `Error Reason`, the first response mentions the possibility of invalid credentials and the likelihood of network

issues during authentication. The second focuses more on incorrect credentials and incorrect client configuration. In `Recommendations`, the first response provides more general instructions. It recommends checking the settings and, if necessary, contacting the administrator. It also notes that no further action may be required if the error occurred only once. The second response, in turn, offers more specific actions: check the credentials and client configuration and reiterate the recommendation to contact the administrator.

Participants were asked to assess at what temperature parameter value the LLM models generated the most accurate and valuable responses and at what value irrelevant, redundant (and potentially hallucinatory) details began to appear. The results showed that the optimal value for most participants lies between 0.4 and 0.6 (Table 6.8), and the threshold for hallucinatory appearance ranges from 0.8 to 2.0 (Table 6.9), depending on the model.

Model	**Temp value**	**Most selected (votes, %)**
Gemini	0.6	7 votes (53.8%)
Nous Hermes 2	0.4	5 votes (38.5%)
LLaMA 2	0.4	6 votes (46.2%)

Table 6.8: Optimal `temperature` value.

Model	**Temp value**	**Most selected (votes, %)**
Gemini	2.0	7 votes (53.8%)
Nous Hermes 2	1.0	5 votes (38.5%)
LLaMA 2	0.8	5 votes (38.5%)

Table 6.9: `temperature` value at which hallucinations begin.

These results show that models with low parameter values (0.4-0.6) generate stable and accurate responses, while increasing the temperature value leads to increased variability and hallucinatory details.

6.3.3 Scenario three results

To determine whether the structure and meaning of the responses are preserved, the survey participants were asked to read five responses from the model to the same log. Then, they needed to assess how significantly these responses differed from each other in content, language, and structure. Response options were offered on the following scale:

1. All answers are identical or nearly the same.
2. Different wording, but the meaning and structure are the same.
3. Answers differ in content but are still relevant.
4. One or more answers seem contradictory or incorrect.
5. The answers vary significantly; the model appears inconsistent.

The benefit of this scale lies in its ability to document variations among responses, from minor stylistic to major semantic differences, offering a nuanced assessment of the stability and consistency of the model, thus revealing its uniformity in reacting to identical input data.

Option №	Gemini	Nous Hermes 2	LLaMA
1	1 votes (7.7%)	1 votes (7.7%)	0 votes (0%)
2	11 votes (84.6%)	7 votes (53.8%)	7 votes (53.8%)
3	1 votes (7.7%)	4 votes (30.8%)	5 votes (38.5%)
4	0 votes (0%)	1 votes (7.7%)	1 votes (7.7%)
5	0 votes (0%)	0 votes (0%)	0 votes (0%)

Table 6.10: Distribution of participant responses for consistency experiment.

As shown in Table 6.10, Gemini showed the most significant stability. Most responses differed only in wording while retaining meaning. Nous Hermes 2 and LLaMA showed more significant content variability, with isolated cases of contradictory answers, unlike Gemini.

6.4 Discussion

In this chapter, a series of experiments were conducted to evaluate the precision, stability, and helpfulness of language model responses in the context of network monitoring. At the end of the survey, participants were asked to assess the very idea of using LLMs to interpret and explain the results of the network monitoring (Table 6.11).

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 6.11: Usefulness of using LLMs for interpreting monitoring results

The responses were highly favorable, with 46.2% of the participants considering the idea *very useful* and another 46.2% describing it as *quite useful*. Only 7.7% remained *neutral*, and no respondents deemed the idea useless or not very useful.

Analyzing the obtained results, all models were generally handled with the task. In most cases, they followed the specified prompt structure, the requirements for which were described in Section 5.1.2. The responses contained a brief explanation of the event, a classification (e.g., '*SMTP Authentication Failure*' or '*Host Unreachable*'), an indication of the cause and appropriate recommendations.

The Gemini model performed the best in all scenarios. Its advantage was especially noticeable in the high delay scenario, where it correctly interpreted a long response time as a potential problem even though the log formally ended successfully. That said, it is conceivable that both Nous Hermes 2 and LLaMA could have handled such a problem with a more precisely worded prompt, for example, if it had explicitly stated the threshold after which high latency should be interpreted as an anomaly. However, even in more apparent situations, such as packet loss in ICMP, the open models did not always recognize this as a problem, suggesting a lack of sensitivity to network degradation.

In experiments with a generation temperature, values up to `0.4` were found to be the *most preferable* under conditions where strict adherence to instructions and minimal variability are required. At higher values, models began to generate more voluminous and, in places, redundant responses while still providing the same basic guidance or explanation,

which may not be desirable. Because when using paid models through APIs such as Gemini, the cost depends on the number of input and output tokens. In terms of stability, all models showed acceptable consistency. The answers remained substantially the same, differing mainly in wording.

In general, when it comes to choosing the most suitable model, there is no universal answer. It depends on many factors, including:

- Data privacy and security requirements.
- Available computing resources and company budget.
- The need for additional training or adaptation of the model for specific tasks.

In a domain where the priority is to have full control over the data or to limit the transfer of information to external services even after anonymization, local models such as LLaMA or Nous Hermes 2 are more suitable. Although they may show less stability in more complex scenarios, they can give quite satisfactory results with properly constructed prompt with thresholds.

When speed is crucial and computational resource usage needs to be minimized, cloud-based models are an option. However, when using the cloud model, risks must also be considered: for example, loss of Internet connection will result in the inability to perform the analysis.

Thus, both models, cloud-based and local, have their advantages, and the choice should be based on the specific application context. For example, in the context of this thesis, where the system analyzes typical network protocols and has well-structured logs, local LLM models have shown decent results on logs with clear error. At the same time, if the system needs to analyze more specific logs of some internal company application and there are no resources to fine-tune the model for this task, but there is the possibility to use an API, anonymize key fields, and provide minimal context, it would be preferable to choose a cloud-based model, such as Gemini.

Chapter 7

Conclusion

In contrast to traditional monitoring systems, which typically log issues without providing further interpretation, this thesis aimed to improve active network monitoring by adding explainable diagnostics using LLMs. As part of the implementation, an active monitoring system (Chapter 3) was developed to check the status of network services such as `SMTP`, `SSH` and `FTP`, as well as the availability test using `ICMP`. The monitoring results are stored in the `JSON` format. Three LLMs (Chapter 5) were integrated to enhance the system's capabilities: *Gemini*, *Nous Hermes 2*, and *LLaMA*. Using `prompt engineering` technique, the models were given clear instructions and an output format to obtain structured and understandable explanations.

The work involved experiments (Chapter 6) in three main areas: evaluating the accuracy of log interpretation, the influence of the temperature parameter on the model behavior, and the stability of responses during multiple generations on the same log file. The results showed that all three models are capable of generating valid and generally correct explanations. *Gemini* demonstrated the highest stability in high-latency scenarios. A participant survey validated the perception that using LLM for explainable diagnostics is favorable and practically valuable.

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. This would allow network administrators to receive both alerts and human-readable diagnostic explanations in real time, directly within their workflow, without having to manually analyze the logs.

Certainly, the work is not without limitations. Firstly, the prompt structure could have been optimized – a simple approach was chosen within the project due to limited time and the need to focus on several models simultaneously. Secondly, as part of the experiment to check stability, the models' responses were compared over five repeated requests. This sample size is limited. Conducting a larger-scale test (for example, with 100 repetitions) would allow more accurate conclusions about the consistency of generations. However, due to time limitations, this was not achieved. 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.

A limitation was also noted with the survey sample: participation was limited to just 13 individuals. Although the responses came from a diverse group, some potential participants declined due to the technical complexity or length of the questionnaire. In future studies, the survey design could be simplified and adapted to a wider audience.

In general, the thesis 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 diagnostic results.

Bibliography

[1] Ryšavý, O.; Kuchynka, M.; Matoušek, P.; Mutua, N. M.; Polčák, L. et al. *Combined passive and active network monitoring* online. 2024. Available at: https://www.fit.vut.cz/research/project/c35612/.en. [cit. 2025-04-13]. Project Period: 1. 1. 2024 – 30. 6. 2026, Project Type: grant, Code: FW10010040, Agency: Technologická agentura ČR.

[2] Wickramasinghe, S. *Active vs. Passive Monitoring: What's The Difference?* online. 2023. Available at: https://www.splunk.com/en_us/blog/learn/active-vs-passive-monitoring.html. [cit. 2024-11-10].

[3] Gigamon. *To TAP or SPAN?* online. 2023. Available at: https://www.gigamon.com/content/dam/resource-library/english/white-paper/wp-tap-vs-span.pdf. [cit. 2024-11-19].

[4] Claise, B. *Cisco Systems NetFlow Services Export Version 9* RFC 3954. IETF RFC, 2004.

[5] Splunk. *Intrusion Detection Systems (IDS): Definition, Types, Purpose* online. 2024. Available at: https://www.splunk.com/en_us/blog/learn/ids-intrusion-detection-systems.html. [cit. 2024-12-09].

[6] Postel, J. *Internet Control Message Protocol* RFC 792. IETF RFC, 1981.

[7] Postel, J. and Reynolds, J. *File Transfer Protocol* RFC 959. IETF RFC, 1985.

[8] Lonvick, C. M. and Ylonen, T. *The Secure Shell (SSH) Transport Layer Protocol* RFC 4253. IETF RFC, 2006.

[9] Klensin, J. *Simple Mail Transfer Protocol* RFC 5321. IETF RFC, 2008.

[10] Wang, Z.; Chu, Z.; Doan, T. V.; Ni, S.; Yang, M. et al. *History, Development, and Principles of Large Language Models-An Introductory Survey.* 2024. Available at: https://arxiv.org/abs/2402.06853.

[11] Engati. *Statistical Language Modeling* online. 2021. Available at: https://www.engati.com/glossary/statistical-language-modeling. [cit. 2025-03-02].

[12] Foote, K. D. *A Brief History of Natural Language Processing* online. 2023. Available at: https://www.dataversity.net/a-brief-history-of-natural-language-processing-nlp/. [cit. 2025-03-02].

[13] Bengio, Y.; Ducharme, R.; Vincent, P. and Janvin, C. A neural probabilistic language model. *J. Mach. Learn. Res.* JMLR.org, 2003, vol. 3, null. ISSN 1532-4435.

[14] Hochreiter, S. and Schmidhuber, J. Long Short-Term Memory. *Neural Computation*, 1997, vol. 9, no. 8, p. 1735–1780.

[15] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L. et al. *Attention Is All You Need*. 2023. Available at: https://arxiv.org/abs/1706.03762.

[16] Devlin, J.; Chang, M.-W.; Lee, K. and Toutanova, K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. Available at: https://arxiv.org/abs/1810.04805.

[17] Yenduri, G.; M, R.; G, C. S.; Y, S.; Srivastava, G. et al. *Generative Pre-trained Transformer: A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions*. 2023. Available at: https://arxiv.org/abs/2305.10435.

[18] Recources, M. L. *Explain the Transformer Architecture (with Examples and Videos)* online. Available at: https://aiml.com/explain-the-transformer-architecture/. [cit. 2025-03-08].

[19] IBM. *What is gradient descent?* online. Available at: https://www.ibm.com/think/topics/gradient-descent. [cit. 2025-03-08].

[20] Kingma, D. P. and Ba, J. *Adam: A Method for Stochastic Optimization*. 2017. Available at: https://arxiv.org/abs/1412.6980.

[21] Team, T. . *What Is Cross-Entropy Loss Function?* online. 2023. Available at: https://365datascience.com/tutorials/machine-learning-tutorials/cross-entropy-loss/. [cit. 2025-03-08].

[22] Raza, M. *LLMs vs. SLMs: The Differences in Large & Small Language Models.* online. 2025. Available at: https://www.splunk.com/en_us/blog/learn/language-models-slm-vs-llm.html. [cit. 2025-03-21].

[23] Li, C. *OpenAI's GPT-3 Language Model: A Technical Overview* online. Available at: https://lambdalabs.com/blog/demystifying-gpt-3. [cit. 2025-03-22].

[24] Google. *Gemma Model Card* online. Available at: https://huggingface.co/google/gemma-2b. [cit. 2025-03-22].

[25] Explorer, L. *LLM Explorer - EXTRACTUM* online. Available at: https://llm.extractum.io. [cit. 2025-03-22].

[26] Cloud, G. *Google models* online. Available at: https://cloud.google.com/vertex-ai/generative-ai/docs/learn/models. [cit. 2025-03-22].

[27] DeepMind, G. *Gemini Developer API* online. Available at: https://ai.google.dev/gemini-api/docs. [cit. 2025-03-22].

[28] AI, M. *Meta Llama 3 8B Instruct* online. Available at: https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct. [cit. 2025-03-22].

[29] NousResearch. *Nous Hermes 2 - Mistral 7B - DPO* online. Available at: https://huggingface.co/NousResearch/Nous-Hermes-2-Mistral-7B-DPO. [cit. 2025-03-22].

[30] Chen, B.; Zhang, Z.; Langrené, N. and Zhu, S. *Unleashing the potential of prompt engineering in Large Language Models: a comprehensive review.* 2024. Available at: https://arxiv.org/abs/2310.14735.

[31] Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J. et al. *Retrieval-Augmented Generation for Large Language Models: A Survey.* 2024. Available at: https://arxiv.org/abs/2312.10997.

[32] Parthasarathy, V. B.; Zafar, A.; Khan, A. and Shahid, A. *The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges and Opportunities.* 2024. Available at: https://arxiv.org/abs/2408.13296.

[33] Fan, W.; Ding, Y.; Ning, L.; Wang, S.; Li, H. et al. *A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models.* 2024. Available at: https://arxiv.org/abs/2405.06211.

[34] Anand, Y.; Nussbaum, Z.; Duderstadt, B.; Schmidt, B. and Mulyar, A. *GPT4All: Training an Assistant-style Chatbot with Large Scale Data Distillation from GPT-3.5-Turbo* online. GitHub, 2023. Available at: https://github.com/nomic-ai/gpt4all.

[35] Anand, Y.; Nussbaum, Z.; Duderstadt, B.; Schmidt, B. and Mulyar, A. *GPT4All Python SDK* online. GitHub, 2023. Available at: https://docs.gpt4all.io/gpt4all_python/home.html.

[36] Holtzman, A.; Buys, J.; Du, L.; Forbes, M. and Choi, Y. *The Curious Case of Neural Text Degeneration.* 2020. Available at: https://arxiv.org/abs/1904.09751.

[37] Cloud, G. *Experiment with parameter values* online. Available at: https://cloud.google.com/vertex-ai/generative-ai/docs/learn/prompts/adjust-parameter-values. [cit. 2025-03-22].

Appendix A

All model responses

A.1 Scenario one results

```
[
    {
        "1. Explanation": "The log shows a successful SMTP transaction with the server example.smtp.server.
             The client connected, authenticated, sent an email, and then disconnected.",
        "2. Classification": "Normal Operation",
        "3. Error Reason": "No error detected",
        "4. Recommendation": "No action required",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The log shows a successful connectivity test to localhost. The test indicates that
              the host is reachable with minimal latency and no packet loss.",
        "2. Classification": "Normal Operation",
        "3. Error Reason": "No error detected",
        "4. Recommendation": "No action required",
        "service": "Connectivity Test"
    }
]
```

Listing A.1: Gemini correct log response.

```
[
    {
        "1. Explanation": "An SMTP transaction with example.smtp.server was initiated, authenticated, sent an
              email, and closed successfully. The transaction took a very long time.",
        "2. Classification": "Excessive Response Time",
        "3. Error Reason": "The response_time_ms is extremely high (613256.0 ms), indicating a severe
             performance issue with the SMTP server or network connectivity.",
        "4. Recommendation": "Investigate the SMTP server's performance, network latency between the client
             and server, and potential bottlenecks. The SMTP server may be overloaded or experiencing other
             issues.",
        "service": "SMTP"
    },

    {
        "1. Explanation": "A connectivity test to 192.168.0.111 was performed and completed successfully,
             however the connection experienced some packet loss and jitter.",
        "2. Classification": "Network Performance Degradation",
        "3. Error Reason": "The packet_loss (22.5%) and jitter_ms (145.872) indicate a potential network
             issue impacting connectivity to 192.168.0.111.",
        "4. Recommendation": "Investigate the network path between the testing device and 192.168.0.111.
             Check for network congestion, faulty network devices, or other factors contributing to packet
             loss and jitter.",
        "service": "Connectivity Test"
    }
]
```

Listing A.2: Gemini high delay log response.

```
[
    {
        "1. Explanation": "An attempt to authenticate with the SMTP server failed.",
        "2. Classification": "SMTP Authentication Failure",
        "3. Error Reason": "The SMTP server rejected the authentication attempt, likely due to incorrect
             username or password.",
        "4. Recommendation": "Verify the SMTP username and password. Ensure that the account is not locked or
              disabled. Check the SMTP server's logs for further details.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "A connectivity test to host 192.168.0.111 failed.",
        "2. Classification": "Host Unreachable",
        "3. Error Reason": "The host 192.168.0.111 is either down, unreachable due to network issues, or a
             firewall is blocking the connection.",
        "4. Recommendation": "Verify that the host 192.168.0.111 is powered on and connected to the network.
             Check network connectivity (e.g., ping) to the host. Investigate any firewall rules that might
             be blocking the connection.",
        "service": "Connectivity Test"
    }
]
```

Listing A.3: Gemini error log response.

```
[
    {
        "1. Explanation": "The log shows an SMTP session that completed successfully between a client at IP
             address 192.168.0.100 and the mail server example.smtp.server on port 587. The session includes
             EHLO command, NOOP response, AUTHENTICATE successful, sending of an email, and finally quitting
             the connection.",
        "2. Classification": "Normal Operation",
        "3. Error Reason": "No error detected",
        "4. Recommendation": "No action required",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The log represents a connectivity test result for localhost using IP address
             127.0.0.1. It shows that the target host is alive and provides information about latency and
             packet loss during the test.",
        "2. Classification": "Normal Operation",
        "3. Error Reason": "No error detected in this log file. The service completed successfully, with no
             issues reported.",
        "4. Recommendation": "No action required as the connectivity test result is within normal parameters
             .",
        "service": "Connectivity Test"
    }
]
```

Listing A.4: Nous Hermes 2 correct log response.

```
[
    {
        "1. Explanation": "The log shows an SMTP session where a client connected to the server at IP address
              '192.168.0.100' on port 587 and completed the communication by executing QUIT command. During
             this session, the client performed EHLO handshake, AUTHENTICATE successfully, sent an email (
             SEND EMAIL), and then closed the connection with a QUIT command.",
        "2. Classification": "Normal Operation",
        "3. Error Reason": "No error detected",
        "4. Recommendation": "No action required",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The log represents a connectivity test result for an IP address '192.168.0.111'.
             It shows that the service is complete and provides information about latency, packet loss, etc
             .",
        "2. Classification": "Normal Operation",
        "3. Error Reason": "No error detected in this log file. The connectivity test has been completed
             successfully with a status of 'complete' and all metrics (latency_avg_ms, latency_min_ms,
             latency_max_ms, jitter_ms, packet_loss) are within expected ranges or limits.",
        "4. Recommation": "No action required as the connectivity test has been completed successfully
             without any issues detected.",
        "service": "Connectivity Test"
    }
]
```

Listing A.5: Nous Hermes 2 high delay log response.

```
[
    {
        "1. Explanation": "The log indicates an SMTP connection attempt to example.smtp.server on port 587
             from IP address 192.168.0.100 at 2025-03-29T19:33:35.848968. The server responded with an error
             message '(535, b'5.7.0 authentication failed')' indicating that the client was unable to
             authenticate successfully during the SMTP session.",
        "2. Classification": "Authentication Failure",
        "3. Error Reason": "The root cause of this issue is likely a failure in providing valid login
             credentials or an incorrect configuration for the AUTHENTICATE command. Without further
             information, it's difficult to determine if there was a network interruption during
             authentication or if the client used invalid credentials.",
        "4. Recommendation": "The user should check their SMTP server settings and ensure that they are using
              valid login credentials. If this issue persists, consult with your system administrator for
             further assistance. No action required if it is a one-time occurrence or the client has
             successfully authenticated in subsequent attempts.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The log indicates that a connectivity test was performed on target host
             '192.168.0.111' and it returned an error status.",
        "2. Classification": "Connectivity Error",
        "3. Error Reason": "The error message states that the host is not reachable, which means there might
             be network or connectivity issues preventing communication with the target host. This could be
             due to a variety of reasons such as firewall rules blocking connections, incorrect IP address,
             or network outage.",
        "4. Recommendation": "Check the network configuration and ensure that all necessary ports are open
             for communication between the source and target hosts. Verify if there are any firewalls or
             routers blocking traffic. If possible, try pinging or connecting to the target host from another
              device on the same network to confirm connectivity issues.",
        "service": "Connectivity Test"
    }
]
```

Listing A.6: Nous Hermes 2 error log response.

```
[
    {
        "1. Explanation": "The log indicates that an SMTP (Simple Mail Transfer Protocol) service has
             completed a transaction successfully. It started by sending an EHLO command to the target host '
             example.smtp.server' and received a response with various features supported by the server. Then
              it performed several commands such as NOOP, AUTHENTICATE, SEND EMAIL, and QUIT.",
        "2. Classification": "Normal Operation",
        "3. Error Reason": "No error detected in this log entry. The SMTP service completed all its
             transactions successfully without any issues or errors reported.",
        "4. Recommendation": "No action required. Since the transaction was successful, there is no need for
             further investigation or corrective actions at this time.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The event recorded in this log is a connectivity test that has been completed
             successfully.",
        "2. Classification": "Normal Operation",
        "3. Error Reason": "No error detected. All metrics such as latency (average 0.06ms, minimum 0.053ms,
             maximum 0.069ms), jitter (7 microseconds) and packet loss (0%) indicate normal network
             performance with no errors or issues found.",
        "4. Recommendation": "No action required. The connectivity test has confirmed that the target host '
             localhost' at IP address '127.0.0.1' is alive and responding normally, indicating a healthy
             connection without any signs of error or malfunction.",
        "service": "Connectivity Test"
    }
]
```

Listing A.7: LLaMa correct log response.

```
[
    {
        "1. Explanation": "The log indicates that an SMTP (Simple Mail Transfer Protocol) service has
             completed a transaction successfully. It started by sending an EHLO command to the target host '
             example.smtp.server' and received a response with various features supported by the server. Then
              it performed several commands such as NOOP, AUTHENTICATE, SEND EMAIL, and QUIT.",
        "2. Classification": "Normal Operation",
        "3. Error Reason": "No error detected in this log entry. The SMTP service completed all its
             transactions successfully without any issues or errors reported.",
        "4. Recommendation": "No action required. Since the transaction was successful, there is no need for
             further investigation or corrective actions at this time.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The event recorded in this log is a connectivity test that has been completed.",
        "2. Classification": "Normal Operation",
        "3. Error Reason": "No error detected. All metrics such as latency (average, minimum, and maximum),
             jitter, and packet loss are within expected ranges for normal network operation.",
        "4. Recommendation": "No action required. The system appears to be functioning normally with no
             indication of any issues that would require attention or intervention.",
        "service": "Connectivity Test"
    }
]
```

Listing A.8: LLaMa high delay log response.

```
[
    {
        "1. Explanation": "The SMTP service experienced an error while attempting to authenticate with a
             remote host (example.smtp.server) over port 587.",
        "2. Classification": "Authentication Failure",
        "3. Error Reason": "The root cause of this issue is that the authentication attempt failed due to
             '(535, b'5.7.0 authentication failed')'. This suggests that there was an invalid username or
             password provided during the authentication process.",
        "4. Recommendation": "To resolve this issue, I recommend verifying and correcting the SMTP
             credentials (username and password) used for authenticating with example.smtp.server. If you are
              unsure about the correct credentials, consult with the system administrator to obtain accurate
             information.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The Connectivity Test service encountered an error while attempting to connect to
             a target host (192.168.0.111) at 2025-03-29T19:37:20.558907.",
        "2. Classification": "Host Unreachable Error",
        "3. Error Reason": "The root cause of the issue is that the 'Host is not reachable' error message
             indicates that the system was unable to establish a connection with the target host
             (192.168.0.111). This could be due to various reasons such as network connectivity issues,
             firewall blocking, or DNS resolution failure.",
        "4. Recommendation": "It's recommended to investigate and resolve any underlying network connectivity
              issues or potential blockages preventing access to the target host. If the issue persists after
              resolving these factors, further troubleshooting may be necessary with the system administrator
             .",
        "service": "Connectivity Test"
    }
]
```

Listing A.9: LLaMa error log response.

A.2 Scenario two results

```
[
    {
        "1. Explanation": "The log shows an SMTP authentication attempt to 'example.smtp.server' on port 587
             failed. The server responded with a 535 error code.",
        "2. Classification": "SMTP Authentication Failure",
        "3. Error Reason": "The SMTP server rejected the authentication attempt. The error message \"(535, b
             '5.7.0 authentication failed')\" indicates incorrect username or password, or the account might
             be locked.",
        "4. Recommendation": "Verify the username and password used for SMTP authentication. If the
             credentials are correct, check if the account is locked or if there are any restrictions on SMTP
              access. Consult with the system administrator to investigate the account status.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "A connectivity test to host '192.168.0.111' failed, indicating the host is not
             reachable.",
        "2. Classification": "Host Unreachable",
        "3. Error Reason": "The target host '192.168.0.111' is either down, unreachable due to network issues
              (e.g., routing problems, firewall blocking ICMP), or does not exist on the network.",
        "4. Recommendation": "Verify that the target host '192.168.0.111' is powered on and connected to the
             network. Check network connectivity, including routing and firewall rules, between the source
             and target hosts. Consult with the system administrator to investigate network connectivity
             issues.",
        "service": "Connectivity Test"
    }
]
```

Listing A.10: Gemini `temp=0.4` log response.

```
[
    {
        "1. Explanation": "An attempt to authenticate with the SMTP server 'example.smtp.server' on port 587
             failed.",
        "2. Classification": "SMTP Authentication Failure",
        "3. Error Reason": "The SMTP server rejected the authentication attempt, likely due to incorrect
             username or password. The error message \"(535, b'5.7.0 authentication failed')\" indicates a
             general authentication problem.",
        "4. Recommendation": "Verify the username and password used for SMTP authentication. If the
             credentials are correct, consult the SMTP server's administrator to ensure the account is active
              and properly configured for authentication from the client's IP address (192.168.0.100).",
        "service": "SMTP"
    },
    {
        "1. Explanation": "A connectivity test to host 192.168.0.111 failed, indicating the host is not
             reachable.",
        "2. Classification": "Host Unreachable",
        "3. Error Reason": "The host 192.168.0.111 is either down, unreachable due to network issues (e.g.,
             routing problem, firewall blocking the connection), or does not exist at the specified IP
             address.",
        "4. Recommendation": "Verify that the host 192.168.0.111 is powered on and connected to the network.
             Check the network configuration, including routing and firewall rules, to ensure there are no
             issues preventing connectivity to the host. Ping the host from a different machine on the same
             network to further diagnose the issue.",
        "service": "Connectivity Test"
    }
]
```

Listing A.11: Gemini `temp=0.6` log response.

```
[
    {
        "1. Explanation": "An attempt to authenticate with the SMTP server failed.",
        "2. Classification": "SMTP Authentication Failure",
        "3. Error Reason": "The SMTP server rejected the authentication attempt, likely due to incorrect
             username or password.",
        "4. Recommendation": "Verify the SMTP username and password are correct. If the issue persists,
             consult the SMTP server's administrator.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "A connectivity test to host 192.168.0.111 failed.",
        "2. Classification": "Host Unreachable",
        "3. Error Reason": "The target host 192.168.0.111 is not reachable from the source. This could be due
              to the host being down, network connectivity issues, or a firewall blocking the connection.",
        "4. Recommendation": "Verify that the target host is powered on and connected to the network. Check
             network connectivity between the source and target, and ensure no firewalls are blocking the
             connection.",
        "service": "Connectivity Test"
    }
]
```

Listing A.12: Gemini `temp=0.8` log response.

```
[
    {
        "1. Explanation": "An attempt to authenticate with the SMTP server failed. The server responded with
             a 535 error code, indicating an authentication problem.",
        "2. Classification": "SMTP Authentication Failure",
        "3. Error Reason": "The SMTP server rejected the authentication attempt, likely due to incorrect
             username or password.",
        "4. Recommendation": "Verify the username and password used for SMTP authentication. If the
             credentials are correct, consult with the system administrator to investigate potential issues
             with the mail server configuration or account permissions.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "A connectivity test to the host 192.168.0.111 failed. The host is reported as not
             reachable.",
        "2. Classification": "Host Unreachable",
        "3. Error Reason": "The target host may be down, unreachable due to network issues (e.g., routing
             problems, firewall blocking), or the IP address may be incorrect.",
        "4. Recommendation": "Verify that the host 192.168.0.111 is powered on and connected to the network.
             Check network connectivity between the testing host and 192.168.0.111, including routing and
             firewall rules. Confirm that the IP address 192.168.0.111 is correct.",
        "service": "Connectivity Test"
    }
]
```

Listing A.13: Gemini `temp=1.0` log response.

```
[
    {
        "1. Explanation": "The log shows an attempt to send an email via SMTP to example.smtp.server on port
             587, which resulted in an authentication failure. The client initially connected, negotiated
             features using EHLO, successfully performed a NOOP command to check server status, but the
             authentication process failed.",
        "2. Classification": "SMTP Authentication Failure",
        "3. Error Reason": "The SMTP server rejected the authentication attempt, indicated by the \"535 5.7.0
              authentication failed\" error message. This typically means the username/password combination
             provided during authentication was incorrect.",
        "4. Recommendation": "Verify the SMTP username and password. Ensure that the account is active and
             not locked. Check for any typos or incorrect configurations in the email client or application
             sending the email. If the issue persists, contact the email service provider to verify the
             account status and authentication settings.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "A connectivity test to host 192.168.0.111 failed. The test determined that the
             host is not reachable.",
        "2. Classification": "Host Unreachable",
        "3. Error Reason": "The target host (192.168.0.111) could not be reached. This could be due to
             several reasons, including the host being down, a network outage, a firewall blocking ICMP
             traffic (ping), or an incorrect IP address.",
        "4. Recommendation": "Verify that the target host is powered on and connected to the network. Check
             network connectivity between the testing device and the target host. Check firewall settings on
             both the testing device and the target host to ensure that ICMP traffic (ping) is allowed.
             Ensure the IP address of the target host is correct.",
        "service": "Connectivity Test"
    }
]
```

Listing A.14: Gemini `temp=2.0` log response.

```
[
    {
        "1. Explanation": "The log indicates an SMTP connection attempt to example.smtp.server on port 587
             from IP address 192.168.0.100 at 2025-03-29T19:33:35.848968. The server responded with an error
             message '(535, b'5.7.0 authentication failed')' indicating that the client was unable to
             authenticate successfully during this connection attempt.",
        "2. Classification": "Authentication Failure",
        "3. Error Reason": "The root cause of the issue is likely due to an incorrect or invalid username and
             /or password being used for SMTP authentication. This could be caused by a user error, such as
             entering the wrong credentials, or it may indicate that the account has been locked or suspended
             .",
        "4. Recommendation": "The system administrator should investigate if there are any issues with the
             provided username and/or password being used for SMTP authentication. It is also important to
             check if the user account is still active and not suspended, locked, or expired. If necessary,
             update the credentials and attempt to reconnect.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The log indicates that a connectivity test was performed on target host
             '192.168.0.111'. However, it could not establish a connection and returned an error status.",
        "2. Classification": "Connectivity Error",
        "3. Error Reason": "The error message suggests that the host is unreachable due to network
             connectivity issues or possible misconfiguration of the target host.",
        "4. Recommendation": "First, ensure that the target host '192.168.0.111' is online and reachable from
              this system. If it is still not working, check the networking configuration on both ends (
             source and destination) for any misconfigurations or potential firewall rules blocking
             communication. Additionally, verify if there are no network devices like routers or switches
             between them that might be causing issues.",
        "service": "Connectivity Test"
    }
]
```

Listing A.15: Nous Hermes 2 `temp=0.4` log response.

```
[
    {
        "1. Explanation": "The log indicates an SMTP communication between a client at IP address '192.168.0.100' and the target server 'example.smtp.server' on port 587. The client attempted to authenticate using the AUTH PLAIN or LOGIN methods, but the authentication process failed.",
        "2. Classification": "Authentication Failure",
        "3. Error Reason": "The error message '(535, b'5.7.0 authentication failed')' indicates that the SMTP server rejected the client's authentication attempt due to invalid credentials or an incorrect authentication method used by the client.",
        "4. Recommendation": "Verify the correctness of the provided authentication credentials and ensure they are valid for the specified AUTHENTICATE method (PLAIN or LOGIN). If the issue persists, consider checking if there is a misconfiguration in the SMTP server settings or contact the system administrator to investigate further. No action required if the client intended to send an email without authentication.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The log indicates that a connectivity test was executed on the target host with IP address '192.168.0.111'. However, it failed to reach the host as it returned an error status.",
        "2. Classification": "Connectivity Error",
        "3. Error Reason": "The issue is caused by the target host being unreachable due to network connectivity problems or potential misconfiguration on either end of the connection. In this case, the 'error_msg' field states 'Host is not reachable', which confirms that it's a connectivity-related problem.",
        "4. Recommendation": "Firstly, ensure that the target host with IP address '192.168.0.111' is powered on and connected to the network. If it is already confirmed, then check if there are any firewall rules or other security measures blocking communication between your system and the target host. Additionally, verify the correctness of the IP address and ensure that the routing tables are properly configured. If none of these solutions resolve the issue, consider consulting with the network administrator for further assistance. No action required if this is a false positive or it's part of normal operation.",
        "service": "Connectivity Test"
    }
]
```

Listing A.16: Nous Hermes 2 `temp=0.6` log response.

```
[
    {
        "1. Explanation": "The log file indicates an SMTP error occurred during authentication with the server example.smtp.server on port 587 from IP address 192.168.0.100 at datetime '2025-03-29T19:33:35.848968'.",
        "2. Classification": "Authentication Failure",
        "3. Error Reason": "The error message '(535, b'5.7.0 authentication failed')' indicates that the server rejected the provided credentials during the login process for SMTP communication. This could be due to incorrect or invalid username/password combination.",
        "4. Recommendation": "Double-check the entered authentication details and ensure they are correct. If necessary, reset the password or consult with system administrator for further assistance in resolving this issue. If it's a new account, make sure that the credentials have been properly set up beforehand. No action required if the provided authentication is correct.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The log indicates that a connectivity test was performed to check if the target host is reachable at '192.168.0.111'. The status of this operation shows an error, and it reports that the target host could not be reached as it is not accessible from the sender's perspective.",
        "2. Classification": "Connectivity Issue",
        "3. Error Reason": "This connectivity issue arises due to network-related problems such as unreachable hosts or routers, firewall blocking connections, or issues with routing protocols and devices. In this specific case, the error message indicates that 'Host is not reachable', suggesting a problem related to either network configuration, security settings on target host, or physical connection loss.",
        "4. Recommendation": "Firstly, check if there are any firewall rules blocking communication between the sender and the target host. Also, ensure that the physical connections (cables) connecting devices in the network path are intact, and no device is experiencing hardware failure. Verify if the target machine is running or powered on. If these basic checks do not resolve the issue, consider reviewing your network configuration settings and routing protocols to identify potential problems.",
        "service": "Connectivity Test"
    }
]
```

Listing A.17: Nous Hermes 2 `temp=0.8` log response.

```
[
    {
        "1. Explanation": "This log shows an SMTP connection attempt made from an IP address '192.168.0.100'
             to an example SMTP server at port 587. The client performed a successful EHLO command and then
             attempted authentication using the AUTH PLAIN or LOGIN method, but it failed resulting in an
             error status of '(535, b'5.7.0 authentication failed')."',
        "2. Classification": "Authentication Error",
        "3. Error Reason": "The error message indicates that the SMTP client was not able to authenticate
             with the server using either AUTH PLAIN or LOGIN method. This can be due to incorrect login
             credentials, a network issue during authentication, or an expired session key among other
             reasons.",
        "4. Recommendation": "Check if the user is entering the correct username and password for SMTP
             authentication. If these are correct and there still appears to be an issue, consult with your
             system administrator as it might require more complex troubleshooting steps. In case it's
             classified as 'Normal Operation', then no action required.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "This log indicates a connectivity test was performed and encountered an error
             while attempting to connect to the target host at IP address '192.168.0.111'. The status of this
              test is marked as 'error'.",
        "2. Classification": "Connectivity Issue",
        "3. Error Reason": "The issue here is that the host at IP address '192.168.0.111' was not reachable
             during the connectivity test, which caused an error in the connectivity testing process. This
             could be due to network connectivity problems or the target host being unavailable/shut down.",
        "4. Recommendation": "First, verify that the IP address is correct and that the device at this
             address is operational. Check if there are any network issues such as firewalls blocking traffic
              or routing problems. If possible, perform a traceroute to identify where in the connection
             process things might be going wrong. Also, ensure all required ports for your service are open
             on both ends of the communication. If you're not sure how to resolve these types of issues,
             consult with your system administrator.",
        "service": "Connectivity Test"
    }
]
```

Listing A.18: Nous Hermes 2 `temp=1.0` log response.

```
[
    {
        "1. Explanation": "In this log, there was an attempt to establish a SMTP connection with example.smtp
             .server on port 587 using ip_address 192.168.0.100. The authentication failed and returned error
              '(535, b'5.7.0 authentication failed')'.",
        "2. Classification": "Error",
        "3. Error Reason": "The reason for this issue is that the provided credentials used during the
             AUTHENTICATE phase were either incorrect or invalid, causing the SMTP server to reject the
             connection. It can also indicate misconfigured settings on client side which led to using an
             invalid set of credential pairs for SMTP authentication.",
        "4. Recommendation": "To resolve this error, verify and confirm the correctness of credentials being
             used in the AUTHENTICATE phase during SMTP connection with example.smtp.server on port 587.
             Double-check that these login credentials are valid by referring to any system or user
             documentation related to them. Also check if client side configuration has proper credential
             sets for this SMTP server and adjust it as required, then retry the SMTP connection with the
             correct credentials. If you're still facing issue consult your System Administrator.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "This log represents a network connectivity test performed to check if the host at
             IP address '192.168.0.111' is reachable or accessible.",
        "2. Classification": "Error",
        "3. Error Reason": "The issue encountered here seems to be related to connectivity with the specified
              target_host ('192.168.0.111') as indicated by the error message 'Host is not reachable'. This
             may suggest that there could be issues with network configuration or the target host might be
             offline at this point in time, hence unable to establish communication.",
        "4. Recommendation": "First and foremost, check the status of your networking infrastructure such as
             router/switches for connectivity problems if you are unsure whether it is a general issue on the
              network. Next, ensure that there isn't an explicit network or security configuration stopping
             communication to this particular host (firewall rule blocking, etc). Finally, verify if the
             target_host ('192.168.0.111') is powered up and accessible in any other tests/methods as a
             possible starting step for troubleshooting.",
        "service": "Connectivity Test"
    }
]
```

Listing A.19: Nous Hermes 2 `temp=10.0` log response.

```
[
    {
        "1. Explanation": "The SMTP service experienced an authentication failure while attempting to connect to a remote server (example.smtp.server) using port 587 from IP address 192.168.0.100.",
        "2. Classification": "Authentication Failure",
        "3. Error Reason": "The root cause of this error is likely due to incorrect or invalid login credentials, such as username and password, provided during the authentication process with the target SMTP server (example.smtp.server). This could be a configuration issue on the client-side or an account lockout on the server-side.",
        "4. Recommendation": "To resolve this issue, please verify that the login credentials are correct and ensure they match any configured settings for the remote SMTP server. If issues persist, consult with the system administrator to review authentication logs and confirm whether there have been any changes made recently that could be causing this error.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The 'Connectivity Test' service attempted to connect to a target host at IP address 192.168.0.111 but failed.",
        "2. Classification": "Unreachable Host",
        "3. Error Reason": "The error message indicates that the root cause of this issue is that the target host (192.168.0.111) is not reachable, possibly due to network connectivity problems or the host being down.",
        "4. Recommendation": "It's recommended to check if there are any issues with the local network configuration or if the target host is indeed down. If it's a temporary issue, waiting for some time and re-running the test might resolve the problem. However, if the issue persists, consulting with the system administrator may be necessary to investigate further.",
        "service": "Connectivity Test"
    }
]
```

Listing A.20: LLaMa `temp=0.4` log response.

```
[
    {
        "1. Explanation": "The SMTP service attempted to authenticate with a remote host (example.smtp.server) but failed due to authentication issues.",
        "2. Classification": "Authentication Failure - Invalid Credentials",
        "3. Error Reason": "The root cause of this error is that the authentication attempt by the SMTP server was rejected because the credentials provided were invalid or incorrect. This could be due to various reasons such as incorrect username/password, account lockout, or temporary network issues preventing successful authentication.",
        "4. Recommendation": "To resolve this issue, I recommend verifying and re-entering the login credentials for the remote host (example.smtp.server) to ensure they are correct. If the problem persists after retrying with valid credentials, it may be necessary to consult with the system administrator to troubleshoot further or consider alternative configuration options.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The log indicates that a connectivity test has failed due to an error.",
        "2. Classification": "Host Unreachable Error",
        "3. Error Reason": "The root cause of this issue is that the target host '192.168.0.111' is not reachable, indicating that there may be a problem with network connectivity or routing.",
        "4. Recommendation": "It is recommended to investigate and resolve any issues affecting network connectivity between the testing system and the target host '192.168.0.111'. This could involve checking for outages, misconfigured firewalls, or other network-related problems. If unsure about how to proceed, it would be best to consult with a system administrator for further guidance.",
        "service": "Connectivity Test"
    }
]
```

Listing A.21: LLaMa `temp=0.6` log response.

```
[
    {
        "1. Explanation": "The SMTP service encountered an error while attempting to authenticate to example.smtp.server on port 587.",
        "2. Classification": "Authentication Failure",
        "3. Error Reason": "The root cause of this issue is that the authentication attempt failed with a status code (535) and message '5.7.0 authentication failed'. This suggests that there was an issue with the username or password used for authentication, possibly due to incorrect credentials or account lockout.",
        "4. Recommendation": "To resolve this issue, I recommend checking if the SMTP service is using correct login credentials and verifying that the account has not been locked out by any security measures on example.smtp.server. If the issue persists after confirming the credentials are valid, consulting with system administrators may be necessary to troubleshoot further or investigate potential security issues on the remote server.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The Connectivity Test service has failed to reach its target host (192.168.0.111) and reported that the host is not reachable.",
        "2. Classification": "Connectivity Issue - Host Not Reachable",
        "3. Error Reason": "The root cause of this error appears to be that the target host 192.168.0.111 is currently unreachable, possibly due to network connectivity issues or the host being down.",
        "4. Recommendation": "It's recommended that you investigate why the target host (192.168.0.111) cannot be reached and take corrective action if necessary. This may involve checking the network configuration, verifying the host's availability, or troubleshooting any potential outages. If unsure about what to do next, please consult with the system administrator for further guidance.",
        "service": "Connectivity Test"
    }
]
```

Listing A.22: LLaMa `temp=0.8` log response.

```
[
    {
        "1. Explanation": "The SMTP service attempted to authenticate to an email server (example.smtp.server) but failed due to authentication issues.",
        "2. Classification": "Authentication Failure",
        "3. Error Reason": "The root cause of the error is that the authentication attempt was unsuccessful because '(535, b'5.7.0 authentication failed')' message indicates that there was a problem with the username and/or password provided during the SMTP authentication process.",
        "4. Recommendation": "It's recommended to check if the login credentials (username and password) are correct for the email server example.smtp.server. If not, update them accordingly. If still experiencing issues, consult with system administrator or contact the email service provider for further assistance.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The Connectivity Test service attempted to connect to host '192.168.0.111' but was unable to do so.",
        "2. Classification": "Network Connectivity Failure",
        "3. Error Reason": "The root cause of this issue is that the target host ('192.168.0.111') is not reachable, indicating a problem with network connectivity or the target host itself. Further investigation would be needed to determine which component is responsible for the failure (e.g., network equipment, router, firewall).",
        "4. Recommendation": "The recommendation is to investigate and troubleshoot why the target host '192.168.0.111' is not reachable. This may involve checking network connectivity issues on the client or server side, verifying routing tables and firewalls configuration, etc.",
        "service": "Connectivity Test"
    }
]
```

Listing A.23: LLaMa `temp=1.0` log response.

```
[
    {
        "1. Explanation": "The log event indicates an authentication failed attempt for a SMTP (Simple Mail
             Transfer Protocol) service connection to the example.smtp.server on port 587 from IP address
             192.168.0.100.",
        "2. Classification": "Authentication Failure - Invalid Credentials",
        "3. Error Reason": "The root cause of this error is likely due to invalid credentials used in the
             authentication process, which resulted in a response code of (535) and message '5.7.0
             authentication failed'.",
        "4. Recommendation": "It is recommended that the administrator reviews and verifies the username/
             password or other credential combinations being attempted for SMTP authentication with example.
             smtp.server, ensuring they are correct and properly configured. Additionally, if there have been
              any changes to accounts or permissions recently, this may be worth investigating further.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "A connectivity test has failed due to the target host being unreachable.",
        "2. Classification": "Network Host Unreachable",
        "3. Error Reason": "The error is likely caused by a problem in reaching or connecting to the target
             host, which is '192.168.0.111'. This could be due to issues such as firewall restrictions,
             incorrect IP address, DNS resolution failure, network outages, or other connectivity problems.",
        "4. Recommendation": "Further investigation and troubleshooting are recommended to resolve the issue
             of host unreachability. Some possible steps include verifying the target host's availability by
             using other tools or checking for any potential misconfigurations in firewalls, routers, or DNS
             servers that may be blocking access to the host.",
        "service": "Connectivity Test"
    }
]
```

Listing A.24: LLaMa `temp=10.0` log response.

A.3 Scenario three results

```
[
    {
        "1. Explanation": "An attempt to authenticate with the SMTP server failed.",
        "2. Classification": "SMTP Authentication Failure",
        "3. Error Reason": "The SMTP server rejected the authentication attempt, likely due to incorrect
             username or password.",
        "4. Recommendation": "Verify the SMTP username and password. If the credentials are correct, consult
             with the system administrator to check the SMTP server's authentication configuration.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "A connectivity test to host 192.168.0.111 failed.",
        "2. Classification": "Host Unreachable",
        "3. Error Reason": "The host 192.168.0.111 is either down, unreachable due to network issues, or a
             firewall is blocking the connection.",
        "4. Recommendation": "Verify that the host 192.168.0.111 is powered on and connected to the network.
             Check network connectivity and firewall rules to ensure that the host is reachable.",
        "service": "Connectivity Test"
    }
]
```

Listing A.25: Gemini response on the same log (1/5).

```
[
    {
        "1. Explanation": "The log shows an attempt to send an email via SMTP to example.smtp.server on port
             587, but the authentication failed.",
        "2. Classification": "SMTP Authentication Failure",
        "3. Error Reason": "The SMTP server rejected the authentication attempt, likely due to incorrect
             username or password. The error message \"(535, b'5.7.0 authentication failed')\" indicates a
             general authentication failure.",
        "4. Recommendation": "Verify the SMTP username and password are correct. If the credentials are
             correct, ensure the account is not locked or disabled. Consult with the system administrator to
             check the mail server configuration.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The log indicates a failed connectivity test to the host 192.168.0.111. The host
             is reported as unreachable.",
        "2. Classification": "Host Unreachable",
        "3. Error Reason": "The target host (192.168.0.111) is either down, not connected to the network, or
             there is a network connectivity issue preventing communication with the host. This could be due
             to a firewall, routing problem, or the host being offline.",
        "4. Recommendation": "Verify that the target host (192.168.0.111) is powered on and connected to the
             network. Check network connectivity to the host using ping or other network diagnostic tools.
             Invest",
        "service": "Connectivity Test"
    }
]
```

Listing A.26: Gemini response on the same log (2/5).

```
[
    {
        "1. Explanation": "An attempt to authenticate with the SMTP server failed. The server responded with
             a 535 error code, indicating an authentication problem.",
        "2. Classification": "SMTP Authentication Failure",
        "3. Error Reason": "The SMTP server rejected the authentication attempt. The error message \"(535, b
             '5.7.0 authentication failed')\" suggests incorrect credentials (username/password) or an issue
             with the authentication mechanism used.",
        "4. Recommendation": "Verify the SMTP username and password. Ensure the correct authentication method
              (PLAIN, LOGIN) is being used and is supported by the server. Check if the account is locked or
             disabled. Consult with the system administrator if the issue persists.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "A connectivity test to the host 192.168.0.111 failed, indicating the host is not
             reachable.",
        "2. Classification": "Host Unreachable",
        "3. Error Reason": "The host 192.168.0.111 may be down, unreachable due to network issues (e.g.,
             routing problems, firewall blocking ICMP), or the IP address may be incorrect.",
        "4. Recommendation": "Verify that the host 192.168.0.111 is powered on and connected to the network.
             Check network connectivity to the host using ping or traceroute. Examine firewall rules to
             ensure they are not blocking ICMP traffic to/from the host. Verify the IP address is correct.",
        "service": "Connectivity Test"
    }
]
```

Listing A.27: Gemini response on the same log (3/5).

```
[
    {
        "1. Explanation": "An attempt to authenticate with the SMTP server failed.",
        "2. Classification": "SMTP Authentication Failure",
        "3. Error Reason": "The SMTP server rejected the authentication attempt, likely due to incorrect
             username or password.",
        "4. Recommendation": "Verify the SMTP username and password. Ensure that the authentication method (
             PLAIN, LOGIN) is supported and correctly configured.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "A connectivity test to the host 192.168.0.111 failed.",
        "2. Classification": "Host Unreachable",
        "3. Error Reason": "The host 192.168.0.111 is either down, not connected to the network, or
             unreachable due to network issues (e.g., routing problems, firewall).",
        "4. Recommendation": "Verify that the host 192.168.0.111 is powered on and connected to the network.
             Check network connectivity and routing to ensure that the host is reachable. Investigate any
             potential firewall issues blocking communication.",
        "service": "Connectivity Test"
    }
]
```

Listing A.28: Gemini response on the same log (4/5).

```
[
    {
        "1. Explanation": "The log shows an attempt to authenticate with an SMTP server failed. The server
             responded with a 535 error code, indicating an authentication problem.",
        "2. Classification": "SMTP Authentication Failure",
        "3. Error Reason": "The SMTP server rejected the authentication attempt. The error message \"(535, b
             '5.7.0 authentication failed')\" suggests incorrect username/password or an issue with the
             authentication mechanism used.",
        "4. Recommendation": "Verify the SMTP username and password. Ensure the correct authentication method
              (PLAIN, LOGIN) is being used and is supported by the server. Check if the account is locked or
             disabled.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "A connectivity test to host 192.168.0.111 failed, indicating the host is not
             reachable.",
        "2. Classification": "Host Unreachable",
        "3. Error Reason": "The host 192.168.0.111 may be down, unreachable due to network configuration
             issues (e.g., incorrect gateway, subnet mask), or a firewall is blocking the connection.",
        "4. Recommendation": "Verify that the host 192.168.0.111 is powered on and connected to the network.
             Check the network configuration (IP address, gateway, subnet mask) on both the source and
             destination hosts. Investigate any firewalls that may be blocking communication between the
             hosts.",
        "service": "Connectivity Test"
    }
]
```

Listing A.29: Gemini response on the same log (5/5).

```
[
    {
        "1. Explanation": "The log shows an SMTP connection attempt to example.smtp.server on port 587 from IP address 192.168.0.100. The server responded with a failure due to authentication issues.",
        "2. Classification": "Authentication Error",
        "3. Error Reason": "The error message indicates that the SMTP client failed to authenticate successfully with the server, resulting in an 'authentication failed' response. This could be caused by incorrect login credentials or misconfiguration of the SMTP client.",
        "4. Recommendation": "Verify and ensure correct authentication details are used for this connection. Check if there is any issue with the configuration settings on both ends (SMTP server and client).",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The log indicates that a connectivity test was performed on the target host (192.168.0.111) at the specified datetime. However, the status of this operation is 'error', indicating that there was an issue with reaching or connecting to the target host.",
        "2. Classification": "Connectivity Issue",
        "3. Error Reason": "The error message provided in the log indicates that the host is not reachable ('Host is not reachable'). This could be due to various reasons such as network connectivity issues, firewall rules blocking connections, or the target host being offline/unreachable.",
        "4. Recommendation": "First, check if there are any network connectivity issues between the system performing the test and the target host (192.168.0.111). Verify that firewall rules on both systems allow connections to each other. If possible, try pinging or connecting to the target host from another device to confirm it's not an issue with the specific machine running the connectivity test. Additionally, check if the target host is online and reachable by verifying its status using other tools like a system monitor or network management tool.",
        "service": "Connectivity Test"
    }
]
```

Listing A.30: Nous Hermes 2 response on the same log (1/5).

```
[
    {
        "1. Explanation": "The log shows an SMTP connection attempt to example.smtp.server on port 587 from ip_address '192.168.0.100'. The server responded with a status of error due to authentication failure.",
        "2. Classification": "Authentication Failure",
        "3. Error Reason": "The error message indicates that the authentication process failed, resulting in an error code (535) and error message 'b'5.7.0 authentication failed''",
        "4. Recommendation": "Check if the login credentials used for authentication are correct or if there is any issue with the authentication mechanism being utilized by the system. If necessary, reset the password or consult with the relevant authorities to ensure proper functioning of the SMTP service. No action required if the authentication details are correct and the error persists; in such cases, it may be due to network connectivity issues or server-side problems which would need further investigation.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The log indicates that a connectivity test was performed on the target host '192.168.0.111' at the specified datetime. However, the host could not be reached as it returned an error status.",
        "2. Classification": "Connectivity Issue",
        "3. Error Reason": "The issue is due to the fact that the target host '192.168.0.111' was not reachable at the specified time, which resulted in an error during the connectivity test. This could be caused by various factors such as network issues, firewall rules blocking connections, or the host being offline.",
        "4. Recommendation": "Check if there are any network connectivity issues between the system performing the connectivity test and the target host '192.168.0.111'. Verify that the necessary ports for communication are open on both ends, and ensure that the target host is online or powered on. If possible, try to perform the connectivity test from a different network or system to isolate if it's an issue with the specific system performing the test.",
        "service": "Connectivity Test"
    }
]
```

Listing A.31: Nous Hermes 2 response on the same log (2/5).

```
[
    {
        "1. Explanation": "The log indicates an SMTP connection attempt to example.smtp.server on port 587
             from IP address 192.168.0.100 at datetime '2025-03-29T19:33:35.848968'. The server responded
             with an error message '(535, b'5.7.0 authentication failed')' after the client sent an EHLO
             command and attempted to authenticate using AUTH PLAIN or LOGIN methods.",
        "2. Classification": "Authentication Failure",
        "3. Error Reason": "The issue is due to incorrect or invalid credentials being used during the SMTP
             AUTHENTICATE process. This could be caused by a misconfiguration of the client's authentication
             settings, an expired password, or incorrect user credentials.",
        "4. Recommendation": "Verify that the correct username and password are being used for authentication
             . Check if there have been any recent changes to the account's password or access permissions.
             If necessary, reset the password and update the client configuration with the new credentials.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The log indicates that a connectivity test was performed on the target host
             '192.168.0.111' at the specified datetime. However, it could not establish a connection and
             returned an error status.",
        "2. Classification": "Connectivity Error",
        "3. Error Reason": "The error message suggests that the host is not reachable due to network
             connectivity issues or possibly the target host might be down.",
        "4. Recommendation": "First, check if the network connectivity between the system performing the test
              and the target host is working properly. If it's a local issue, try pinging or connecting from
             another device on the same network to confirm whether the problem lies with the target host or
             the network itself. If the network connection appears fine, then investigate further about the
             status of the target host (e.g., if it's powered off, not reachable due to firewall rules etc.)
             .",
        "service": "Connectivity Test"
    }
]
```

Listing A.32: Nous Hermes 2 response on the same log (3/5).

```
[
    {
        "1. Explanation": "The log indicates an SMTP connection attempt to example.smtp.server on port 587
             from IP address 192.168.0.100 at 2025-03-29T19:33:35.848968. The server responded with an error
             message '(535, b'5.7.0 authentication failed')' indicating that the client was unable to
             authenticate successfully during the SMTP session.",
        "2. Classification": "Authentication Failure",
        "3. Error Reason": "The root cause of this issue is likely an incorrect or invalid username and/or
             password provided by the client attempting to connect to example.smtp.server. This could also be
              due to a misconfiguration in the SMTP server settings on the client side, such as using the
             wrong authentication method or providing incorrect credentials for the specified authentication
             method.",
        "4. Recommendation": "The system administrator should verify that the correct username and password
             are being used by the client attempting to connect to example.smtp.server. If the issue persists
              after verifying the login credentials, it may be necessary to review the SMTP server settings
             on the client side for any misconfigurations or errors.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The log indicates that a connectivity test was performed on target host
             '192.168.0.111' at the specified datetime. The status of this test is reported as an error, and
             it seems that the host could not be reached or is unreachable.",
        "2. Classification": "Connectivity Issue",
        "3. Error Reason": "The target host '192.168.0.111' appears to be unreachable due to a network
             connectivity problem, which might be caused by issues such as misconfigured routers or firewalls
              blocking connections, physical cable disconnection, or the target device being powered off.",
        "4. Recommendation": "First, check if the target host is online and accessible from other devices on
             the network. If it's not reachable from any device, then there might be a problem with the
             network infrastructure (e.g., misconfigured routers or firewalls). In this case, consult your
             system administrator to investigate further and resolve the issue. If the target host is
             accessible from other devices on the network, then it's possible that there is an issue specific
              to the device running the connectivity test. Check its network configuration (e.g., IP address,
              DNS settings) or try performing the connectivity test from another device to confirm if the
             problem persists.",
        "service": "Connectivity Test"
    }
]
```

Listing A.33: Nous Hermes 2 response on the same log (4/5).

```
[
    {
        "1. Explanation": "The log shows an SMTP connection attempt to example.smtp.server on port 587 from IP address 192.168.0.100. The server responded with a '535 authentication failed' error message, indicating that the client was unable to authenticate successfully during the session.",
        "2. Classification": "Authentication Failure",
        "3. Error Reason": "The error occurred because the SMTP client failed to provide valid credentials for authentication during the connection attempt. This could be due to incorrect username or password, expired credentials, or a misconfiguration of the client's authentication settings.",
        "4. Recommendation": "Check and verify the client's authentication credentials (username and password) and ensure they are correct and valid. If necessary, update any outdated or expired credentials. Additionally, review the client's SMTP configuration to ensure that it is properly configured for authentication with example.smtp.server.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The log indicates that a connectivity test was performed on target host '192.168.0.111' at the specified datetime and it returned an error status.",
        "2. Classification": "Connectivity Issue",
        "3. Error Reason": "The error message 'Host is not reachable' suggests that there might be a network connectivity problem or the target host could be down, which prevented the connectivity test from reaching it successfully.",
        "4. Recommendation": "Check if the target host '192.168.0.111' is online and accessible within your network. Verify that there are no firewalls or other security measures blocking communication to this IP address. If possible, try performing the connectivity test from another device on the same network to rule out any local issues.",
        "service": "Connectivity Test"
    }
]
```

Listing A.34: Nous Hermes 2 response on the same log (5/5).

```
[
    {
        "1. Explanation": "The SMTP service attempted to authenticate with a remote server (example.smtp.server) on port 587 but failed due to authentication issues.",
        "2. Classification": "Authentication Failure",
        "3. Error Reason": "The root cause of this error is that the authentication attempt was rejected by the remote server, likely because the credentials provided were invalid or incorrect. This could be due to various reasons such as incorrect username/password combination, account lockout, or temporary network connectivity issue.",
        "4. Recommendation": "To resolve this issue, I recommend verifying and double-checking the SMTP login credentials (username and password) used by the service. If issues persist, it may be necessary to consult with the system administrator for further assistance in troubleshooting or resetting the authentication settings.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The Connectivity Test service has encountered an error while attempting to connect to the target host '192.168.0.111'. The test resulted in a failure, indicating that the host is not reachable.",
        "2. Classification": "Host Unreachable Error",
        "3. Error Reason": "The root cause of this issue appears to be that the target host '192.168.0.111' is currently unreachable or down. This could be due to various reasons such as network connectivity issues, firewall rules blocking access, or the host being powered off.",
        "4. Recommendation": "It is recommended to investigate and resolve any underlying network connectivity issues or check if the target host is indeed reachable from other devices on the same network. If the issue persists after troubleshooting, it may be necessary to consult with the system administrator for further assistance.",
        "service": "Connectivity Test"
    }
]
```

Listing A.35: LLaMa response on the same log (1/5).

```
[
    {
        "1. Explanation": "The SMTP service experienced an error while attempting to authenticate with a
             remote host (example.smtp.server) over port 587.",
        "2. Classification": "Authentication Failure",
        "3. Error Reason": "The root cause of this issue is likely due to incorrect or invalid authentication
              credentials being provided by the client, resulting in the server responding with a '535' SMTP
             code and an error message indicating that the authentication failed (5.7.0).",
        "4. Recommendation": "To resolve this issue, verify that the correct username and password are being
             used for authentication. If using PLAIN text or LOGIN authentication methods, ensure that they
             are properly configured and not compromised by any unauthorized access. Consider enabling
             additional security measures such as TLS encryption to secure the connection. Please note that
             if you're unsure about the diagnosis or would like further assistance, it's always best to
             consult with a system administrator for more detailed analysis and guidance.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The Connectivity Test service has encountered an error while attempting to connect
              to the target host '192.168.0.111'. The test indicates that the host is not reachable.",
        "2. Classification": "Host Unreachable Error",
        "3. Error Reason": "The root cause of this issue appears to be that the target host '192.168.0.111'
             is currently unreachable, possibly due to network connectivity issues or the host being down.",
        "4. Recommendation": "It would be recommended to investigate and resolve any underlying network
             connectivity issues or check if the target host is indeed down for maintenance. If the issue
             persists after resolving these potential causes, further troubleshooting may be necessary with
             the system administrator's assistance.",
        "service": "Connectivity Test"
    }
]
```

Listing A.36: LLaMa response on the same log (2/5).

```
[
    {
        "1. Explanation": "The SMTP service experienced an error while attempting to authenticate with a
             remote server (example.smtp.server) over port 587 from IP address 192.168.0.100.",
        "2. Classification": "Authentication Failure",
        "3. Error Reason": "The root cause of the issue is that authentication failed due to incorrect
             credentials or invalid username/password combination, resulting in an error message '(535, b
             '5.7.0 authentication failed')'.",
        "4. Recommendation": "To resolve this issue, please verify and correct the SMTP server's login
             credentials (username and password) used by the service. If you are unsure about the correct
             credentials, consult with the system administrator or contact the remote server administrators
             for assistance. Note: Since there is an error detected in the log, I did not classify it as '
             Normal Operation'.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The Connectivity Test service has encountered an error while attempting to connect
              to a target host at IP address '192.168.0.111'. The test indicates that the host is not
             reachable.",
        "2. Classification": "Host Unreachable Error",
        "3. Error Reason": "The root cause of this issue appears to be that the target host ('192.168.0.111')
              is currently unreachable, possibly due to network connectivity issues or the host being down.",
        "4. Recommendation": "It would be recommended to investigate and resolve any underlying network
             connectivity issues before retrying the Connectivity Test service. If the issue persists after
             resolving potential network problems, it may be necessary to consult with the system
             administrator for further assistance.",
        "service": "Connectivity Test"
    }
]
```

Listing A.37: LLaMa response on the same log (3/5).

```
[
    {
        "1. Explanation": "The SMTP service experienced an error while attempting to authenticate with a
             remote host (example.smtp.server) over port 587.",
        "2. Classification": "Authentication Failure",
        "3. Error Reason": "The root cause of this issue is that the authentication attempt failed due to
             incorrect credentials or invalid username/password combination, resulting in a response code
             '535' and message '(5.7.0 authentication failed)' from the remote host.",
        "4. Recommendation": "To resolve this issue, please verify the SMTP server's login credentials (
             username and password) are correct and try re-authenticating with the corrected information. If
             you're unsure about the credentials or have any questions, consult with your system
             administrator for further assistance.",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The Connectivity Test service attempted to connect to a target host
             (192.168.0.111) but failed.",
        "2. Classification": "Host Unreachable Error",
        "3. Error Reason": "The root cause of this error is that the target host '192.168.0.111' is not
             reachable, indicating that there may be an issue with network connectivity or the host itself
             being down.",
        "4. Recommendation": "It's recommended to check if the target host is actually up and running, and
             also verify the network configuration to ensure that it can reach the host. If the problem
             persists, consult with a system administrator for further assistance.",
        "service": "Connectivity Test"
    }
]
```

Listing A.38: LLaMa response on the same log (4/5).

```
[
    {
        "1. Explanation": "The SMTP service experienced an error while attempting to authenticate with a
             remote host (example.smtp.server) over port 587.",
        "2. Classification": "Authentication Failure",
        "3. Error Reason": "The root cause of this issue is that the authentication attempt failed due to
             '(535, b'5.7.0 authentication failed')'. This suggests that there was an invalid username or
             password provided during the authentication process.",
        "4. Recommendation": "To resolve this issue, I recommend verifying and correcting the username and/or
              password used for SMTP authentication with example.smtp.server. If you are unsure about the
             correct credentials, please consult with the system administrator to obtain accurate information
             .",
        "service": "SMTP"
    },
    {
        "1. Explanation": "The Connectivity Test service has encountered an error while attempting to connect
              to the target host '192.168.0.111'. The test was unable to establish a connection and
             determined that the host is not reachable.",
        "2. Classification": "Host Unreachable Error",
        "3. Error Reason": "The root cause of this issue is likely due to the fact that the target host
             '192.168.0.111' is either down or unreachable, possibly due to network connectivity issues,
             firewall rules blocking access, or other infrastructure problems.",
        "4. Recommendation": "It would be recommended to investigate and troubleshoot why the target host is
             not reachable. This could involve checking network configurations, verifying that the host is up
              and running, and ensuring that there are no firewalls or other obstacles preventing connection
             attempts. If the issue persists after troubleshooting, it may be necessary to consult with the
             system administrator for further guidance.",
        "service": "Connectivity Test"
    }
]
```

Listing A.39: LLaMa response on the same log (5/5).