Can AI Outsmart Firewall Errors? a Study on LLMs for Anomaly Generation and Detection

Chang-Sheng Lee[1] and Ling-Jyh Chen[1,2]
[1]*Institute of Information Science, Academia Sinica*
[2]*Research Center for Information Technology Innovation, Academia Sinica*
{johnsonlee911205, cclljj}@iis.sinica.edu.tw

***Abstract*—Modern network security relies on firewalls to block advanced cyber threats. Managing large firewalls with thousands of rules is difficult, and rule anomalies can weaken protection. Traditional algorithms detect these anomalies with speed and accuracy, but Large Language Models (LLMs) offer reasoning abilities that may reveal hidden patterns beyond fixed rules. This paper evaluates state-of-the-art LLMs for two tasks: generating firewall datasets with anomalies and detecting those anomalies. Although these models are built for general-purpose reasoning, they struggle with the structured logic required in firewall management. Future work should focus on adapting their reasoning capabilities to fit real-world firewall tasks.**

***Index Terms*—Firewall, Anomaly Generation, Anomaly Detection, LLMs**

I. Introduction

Firewalls are a core component of network security. As the number of firewall rules grows, managing them becomes increasingly difficult. Al-Shaer and Hamed [1] identified four common types of rule conflicts, referred to as anomalies:

1) **Shadowing:** A preceding rule matches all packets that a later rule would match.
2) **Correlation:** Two rules overlap, each matching some of the same packets.
3) **Generalization:** A broader rule appears after a more specific one that contains all the necessary details.
4) **Redundancy:** A rule duplicates another's effect.

Several algorithms have been developed to detect anomalies, such as Firewall Policy Advisor [1] and FIREMAN [2]. These tools are fast and accurate, but rely on pre-defined rule.

Recent work by Louro et al. [3] suggests that Large Language Models (LLMs) may help generate firewall rules. Unlike fixed-rule systems, LLMs offer general reasoning capabilities, which can uncover patterns that traditional tools miss. This study tests whether state-of-the-art LLMs can perform two key tasks: generating firewall rule sets with anomalies and detecting anomalies within them.

We designed a two-stage experiment. In Stage 1, four reasoning-focused LLMs generate firewall rule datasets that include anomalies. This reveals their ability to simulate realistic rule sets with internal conflicts. In Stage 2, the same models attempt to detect anomalies within the generated datasets. This enables us to assess how effectively these models interpret structured firewall logic.

This research was supported by the National Science and Technology Council of Taiwan under Grant: NSTC 112-2221-E-001-005-MY3.

II. Methodology

We evaluated four leading reasoning models available as of June 2025. OpenAI o3 (o3) [5], Google Gemini 2.5 Pro (Gemini) [6], xAI Grok3 (Grok) [7], and DeepSeek R1 Distill Llama 70B (DeepSeek) [8]. The first three are commercial models. DeepSeek is open source. These models are designed to handle complex tasks by extending reasoning time and breaking down problems into smaller parts.

We also observe that LLM prompts often lead to better results than manually written ones. This may be because LLMs can summarize the task requirements more clearly. Therefore, we used OpenAI GPT-4.5 [4], renowned for its strong creative capabilities, to generate prompts for the experiment.

A. Stage 1 – Firewall Rule Anomaly Generation

In Stage 1, each model generates 50 datasets. Each dataset contains 100 firewall rules. To support a later evaluation, we require each dataset to include two instances of each of the four types of anomalies. We use few-shot prompting to guide the models. Each rule consists of the following fields: counter, source interface, destination interface, source address, destination address, service (protocol and port), and action. The counter helps ensure that the model produces exactly 100 rules, as LLMs often miscount.

To check whether the generated datasets meet the anomaly requirement, we build a brute-force program. This tool scans each dataset to identify all actual anomalies and the corresponding rule pairs.

B. Stage 2 – Firewall Rule Anomaly Detection

In Stage 2, each model analyzes 200 datasets, which are all the datasets produced in Stage 1. Each model attempts to detect anomalies in these rule sets.

We compare each model's results with the ground truth found by the brute-force program from Stage 1. This lets us measure how well each model identifies each type of anomaly across data generated by different models.

TABLE I: Confusion matrix definitions for Stage 1.

Label	Description
TP	An expected anomaly that is correctly generated.
FP	An extra anomaly beyond the expected amount.
FN	An expected anomaly that was not generated.

TABLE II: Confusion matrix definitions for Stage 2.

Label	Description
TP	A real anomaly that is correctly detected.
FP	A detected anomaly that does not exist.
FN	A real anomaly that is missed.

III. Evaluation

Inference for commercial models is conducted through their official APIs. For DeepSeek, inference is performed through an API running on Groq's Language Processing Unit (LPU) [9], due to hardware constraints. All parameters are set to default values, except `max_tokens=20000` for DeepSeek.

We use precision and recall in Stage 1, and F1-score in Stage 2. All metrics are based on confusion matrices, but their interpretations differ between stages.

Table I defines the confusion matrix terms for Stage 1. Based on these, we calculate the following.

- **Precision:** The proportion of anomalies generated that match the expected number.
- **Recall:** The proportion of expected anomalies that are successfully generated.

A precision close to 1 means fewer extra anomalies are generated. A recall close to 1 means that the model generated all required anomalies.

Table II defines the confusion matrix terms for Stage 2. Based on these, we calculate the following.

- **Precision:** The proportion of anomalies detected that exist.
- **Recall:** The proportion of actual anomalies that are correctly detected.

We use the F1-score in Stage 2 to balance precision and recall. A higher F1-score means better detection performance.

Figure 1 shows the Stage 1 results. o3 performs best across all types of anomalies, with high precision and recall. DeepSeek performs the worst, exhibiting a lower recall for correlation anomalies, suggesting that it struggles to identify correlation anomalies, which also indicates that it may struggle to generate them.

Figure 2 shows the Stage 2 F1-scores. Gemini achieves the best overall detection performance. o3, though strong in Stage 1, ranks lower here. DeepSeek again performs the worst. Models do not detect anomalies more accurately in datasets they generated themselves.

IV. Conclusion

This study evaluated four leading Large Language Models on two tasks: generating and detecting anomalies in firewall rules. OpenAI o3 showed the strongest performance in generation. Google Gemini 2.5 Pro led the detection. No model performed well in both tasks.

These results suggest that current LLMs lack the precision and consistency needed for complex firewall management. They require domain-specific training and should be used in conjunction with traditional tools. Although not yet reliable for detecting firewall errors, LLMs may play a helpful role in future cybersecurity systems.

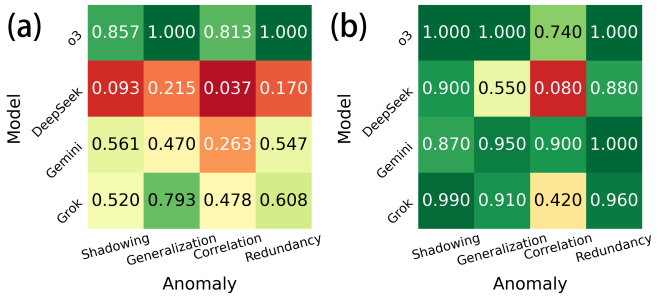

Fig. 1: (a) Stage 1 precision (b) Stage 1 recall

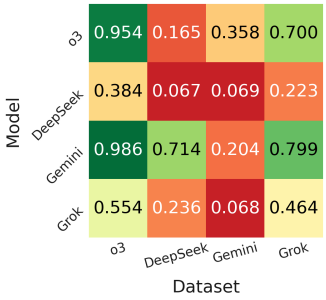

Fig. 2: Stage 2 F1-score

References

[1] E. S. Al-Shaer and H. H. Hamed, "Firewall Policy Advisor for anomaly discovery and rule editing," IFIP/IEEE Eighth International Symposium on Integrated Network Management, 2003., Colorado Springs, CO, USA, 2003, pp. 17–30.

[2] Lihua Yuan, Hao Chen, Jianning Mai, Chen-Nee Chuah, Zhendong Su and P. Mohapatra, "FIREMAN: a toolkit for firewall modeling and analysis," 2006 IEEE Symposium on Security and Privacy (S&P'06), Berkeley/Oakland, CA, USA, 2006, pp. 15 pp.-213.

[3] B. Louro, R. Abreu, J. C. Costa, J. B. F. Sequeiros, and P. R. M. Inácio, "Analysis of the Capability and Training of Chat Bots in the Generation of Rules for Firewall or Intrusion Detection Systems," in Proc. 19th Int. Conf. Availability, Reliability and Security (ARES '24), New York, NY, USA: ACM, 2024, Art. no. 123, pp. 1–7.

[4] OpenAI, "GPT-4.5 Preview." https://platform.openai.com/docs/models/gpt-4.5-preview.

[5] OpenAI, "o3." https://platform.openai.com/docs/models/o3.

[6] Google, "Gemini 2.5 Pro." https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-pro.

[7] xAI, "Grok 3." https://docs.x.ai/docs/models.

[8] Groq, "DeepSeek R1 Distill Llama 70B." https://console.groq.com/docs/model/deepseek-r1-distill-llama-70b.

[9] Groq, "What is a Language Processing Unit?." https://groq.com/the-groq-lpu-explained/