

Predictive Maintenance on Fortinet Firewall Devices Using Artificial Intelligence

April Rustianto^{1*}, Faqih Murobbie², Rusmanto³

^{1,2,3}STT Terpadu Nurul Fikri, Indonesia

¹april.rustianto@dosen.nurulfikri.ac.id, ²faqi23093ti@student.nurulfikri.ac.id, ³rusmanto@nurulfikri.ac.id



*Corresponding Author

Article History:

Submitted: 09-12-2025

Accepted: 20-12-2025

Published: 28-12-2025

Keywords:

Predictive maintenance; firewall logs; Fortinet; large language models; AI benchmarking

Brilliance: Research of Artificial Intelligence is licensed under a Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0).

ABSTRACT

The growing complexity of enterprise network infrastructures has increased the importance of predictive maintenance for network security devices, particularly firewall systems. In operational environments using Fortinet firewalls, large volumes of firewall logs are continuously generated, while existing monitoring tools such as FortiAnalyzer remain limited to descriptive analysis and lack predictive capabilities. This study aims to evaluate the effectiveness of artificial intelligence, specifically Large Language Models (LLMs), for predictive maintenance through automated analysis of firewall logs. Four open-source LLMs-Gemma 2B, Mistral 7B, DeepSeek-R1 7B, and Qwen 2.5-Coder 7B-were benchmarked using a standardized Indonesian-language prompt designed to extract high-severity events, including emergency, alert, and critical conditions, from multi-severity Fortinet log data. The evaluation focused on AI benchmarking metrics such as severity filtering compliance, reasoning accuracy, linguistic consistency, structural clarity, and processing efficiency. The results indicate that Qwen 2.5-Coder 7B provides the most reliable overall performance, demonstrating strong adherence to severity constraints, consistent Indonesian-language output, and well-structured analytical results suitable for operational predictive maintenance. Mistral shows superior contextual reasoning but exhibits language inconsistency, while Gemma offers the fastest processing time with moderate severity accuracy. DeepSeek performs least effectively due to instruction non-compliance. This study addresses an existing research gap by demonstrating how large language models can support predictive maintenance for firewall-based network security systems and provides a comparative framework for future AI-driven firewall and log-analysis research.

INTRODUCTION

The rapid growth of enterprise network infrastructures has significantly increased operational complexity, particularly in environments that rely on distributed and software-defined architectures. Firewalls play a central role in maintaining network security, availability, and policy enforcement, especially within Software-Defined Wide Area Network (SD-WAN) deployments. Fortinet firewall devices are widely adopted in such environments due to their integrated security and traffic management capabilities. However, these devices continuously generate large volumes of firewall logs that record system events, configuration changes, authentication attempts, and traffic anomalies. When not analyzed effectively, this log data becomes underutilized, reducing its potential value for maintaining system reliability and preventing service disruptions.

Existing monitoring platforms, such as FortiAnalyzer, primarily focus on log aggregation, visualization, and rule-based alerting. While these tools are effective for post-event analysis and compliance reporting, they offer limited support for predictive maintenance. Specifically, they lack automated reasoning, contextual interpretation, and forward-looking insights that could assist operators in anticipating failures or security incidents. This limitation becomes increasingly critical in large-scale deployments involving dozens or even hundreds of firewall devices. In such scenarios, manual log inspection is not only time-consuming but also highly susceptible to human error, delayed response, and inconsistent interpretation across different operators.

Predictive maintenance has traditionally relied on statistical models, anomaly detection techniques, and machine-learning-based classifiers to forecast system failures. Prior studies have demonstrated the effectiveness of these approaches in industrial systems and networked environments (Wang, 2020; Kumar, 2022). However, these methods often require extensive feature engineering, labeled datasets, and domain-specific tuning, which can limit their adaptability when applied to heterogeneous and unstructured log data such as firewall logs. As a result, there is a growing interest in leveraging Artificial Intelligence (AI) techniques that can process raw textual data with minimal preprocessing.

Large Language Models (LLMs) have recently emerged as a promising alternative for log interpretation, summarization, and anomaly detection. Research by Wu (2023) and Kim (2023) has shown that LLMs can capture



contextual relationships within complex log structures more effectively than traditional natural language processing pipelines. Zhang (2024) further demonstrated that LLMs are capable of extracting meaningful insights from cybersecurity logs, including threat indicators and system anomalies. Despite these advancements, existing studies largely focus on cloud infrastructure logs or generic security event data and do not address the unique characteristics of firewall-specific logs, which include severity hierarchies, rule-matching behavior, and device-centric operational events.

Moreover, prior research rarely conducts systematic benchmarking of multiple LLMs using identical prompts and evaluation criteria, particularly in the context of predictive maintenance for firewall environments. This creates a gap in understanding how different LLM architectures perform when applied to operational firewall log analysis under controlled conditions. Addressing this gap is essential for guiding practitioners in selecting suitable AI models for real-world network operations.

This study aims to fill this gap by conducting a comparative evaluation of several open-source LLMs for predictive maintenance on Fortinet firewall devices. By benchmarking Gemma 2B, Mistral 7B, DeepSeek-R1 7B, and Qwen 2.5-Coder 7B using standardized Indonesian-language prompts and consistent evaluation metrics, this research contributes empirical evidence on the strengths and limitations of each model. The findings are intended to support the integration of AI-driven log analysis into operational network security workflows and to inform future research on predictive maintenance using large language models.

LITERATURE REVIEW

Predictive maintenance has long been grounded in statistical reliability models, anomaly detection frameworks, and a range of machine-learning (ML) approaches. Earlier investigations—such as the work of Wang (2020) and Kumar (2022)—demonstrated the effectiveness of ML-driven prediction systems in industrial IoT and networked environments. These studies highlight how supervised and unsupervised learning models have traditionally been used to anticipate equipment failures and detect irregular operational patterns. More recent developments in generative artificial intelligence have introduced large language models (LLMs) as a potential alternative for interpreting complex log data, particularly in contexts where traditional ML models may struggle with unstructured text.

Research on AI-based log interpretation continues to expand across several domains. Wu (2023), for example, employed GPT-based architectures to improve the accuracy of cloud log parsing, showing that generative models can capture contextual dependencies that rule-based systems often overlook. Kim (2023) similarly examined the anomaly-detection capability of ChatGPT within heterogeneous network environments, reporting notable improvements in pattern recognition compared with earlier NLP-based pipelines. Complementing these findings, Zhang (2024) benchmarked multiple LLMs for cybersecurity log analysis and observed promising results in tasks involving structured log summarization and threat-indicator extraction.

Other studies have broadened this discussion by evaluating LLMs in operational security and network-monitoring workflows. Li and Chen (2021) explored transformer-based anomaly detection for distributed systems and found that semantic-aware embeddings improved fault-prediction precision. Hassan and Rahman (2022) demonstrated how contextual embeddings support early detection of misconfigurations in enterprise firewalls. In addition, Silva et al. (2023) modeled predictive behaviors in network traffic using hybrid deep-learning and LLM-driven parsing mechanisms.

A growing body of literature also examines how LLMs can support incident response and log triage. For instance, Patel and Morgan (2022) assessed the utility of open-source LLMs for summarizing real-time SOC alerts. Meanwhile, Ortega and Ramos (2023) analyzed failure reports in virtualized infrastructure using generative models to identify causal sequences. Chen and Al-Khatib (2024) expanded this work by evaluating model interpretability when handling high-volume network telemetry.

Despite this momentum, very few studies address the unique characteristics of firewall log data, which often include device-specific events, rule-matching outcomes, and packet-flow anomalies not typically present in generic security logs. Existing research—such as the frameworks proposed by Yadav (2021) for edge-network anomaly detection and by Brown (2022) for predictive modeling in software-defined networks—does not evaluate the applicability of generative LLMs to firewall-centric operational environments. Similarly, while Nguyen and Foster (2023) explored open-source LLM performance for general cyber-operations automation, their study did not incorporate a comparative assessment tailored to predictive maintenance tasks.

Taken together, the literature reveals active exploration of LLMs for security analytics, log interpretation, and anomaly detection. However, there remains no dedicated comparative study evaluating multiple open-source LLMs using identical prompts to analyze firewall-specific logs for predictive maintenance scenarios. The present study addresses this gap by systematically benchmarking several LLMs under controlled conditions to observe their analytical performance, execution time, and operational suitability.

Research Gap: Existing studies largely concentrate on cloud-based or generic security log analysis, with limited attention given to the unique characteristics of firewall-specific logs. Moreover, prior work rarely conducts a systematic comparison of multiple large language models using an identical prompt structure. Consequently, there remains a notable gap in understanding how different LLMs perform when applied to operational firewall environments. This



study addresses this gap by offering a comparative evaluation designed to support predictive maintenance practices in real-world operational settings.

METHOD

This study follows an experimental benchmarking methodology consistent with Brilliance Journal standards.

1. Data Preparation: Firewall logs were modified to include emergency, alert, critical, warning, notice, and informational entries. This ensured evaluation of severity-filtering behavior.
2. Model Selection: Four open-source LLMs were chosen: Gemma 2B, Mistral 7B, DeepSeek-R1 7B, Qwen 2.5-Coder 7B.
3. Prompt Design: All models received the same Indonesian-language prompt requiring:
 - a. High-severity filtering (emergency, alert, critical)
 - b. Structured key points
 - c. Conclusions
 - d. Preventive actions
 - e. Full Indonesian output
4. Evaluation Metrics: Severity filtering compliance
 - a. Correct identification of high-severity events
 - b. Linguistic consistency
 - c. Structural completeness
 - d. Processing time
5. Execution Framework: Models were executed using identical inference hardware settings to ensure fairness. We just change the LLM model

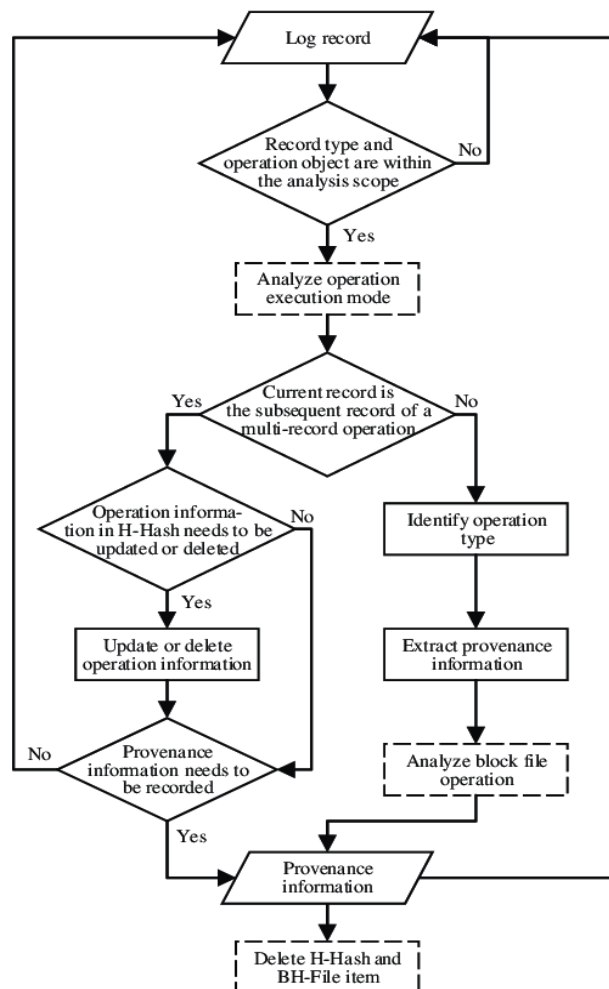


Fig. 1 Flowchart

Flowchart Description of the Proposed Method

The proposed experimental framework describes a controlled pipeline for evaluating large language models (LLMs) in firewall log analysis. Users submit firewall log files containing heterogeneous severity levels (emergency, alert, critical, warning, notice, and informational) through a web-based interface. These logs are processed by a Django-based backend, which manages file handling, preprocesses the data, and generates standardized prompts. This design ensures that all evaluated models receive identical inputs, thereby preserving experimental consistency and fairness.

The prepared data and prompts are then forwarded to the Ollama inference engine, which enables seamless execution and switching of different open-source LLMs without modifying the application logic. Each selected model performs severity filtering, structured log summarization, conclusion generation, and preventive action recommendations in Indonesian. The analysis results are returned to the dashboard interface and subsequently evaluated using predefined metrics, including severity filtering compliance, analytical quality, linguistic consistency, structural clarity, and processing time. This workflow guarantees a reproducible evaluation environment in which observed performance differences are attributable solely to model characteristics.

RESULT

Gemma2:2b: Fastest model (648 seconds), accurate but occasionally includes lower-severity context.

Mistral: Strongest in contextual reasoning but mixes Indonesian and English.

Deepseek-r1:7b: Fails to follow Indonesian-language instruction; weakest severity filtering

Qwen2.5-coder:7b: Most balanced model with strong severity compliance and clear structure.

Comparative Table:

Tabel 1. Comparative LLM result

Model	Severity Compliance	Language Accuracy	Structure	Processing Speed
Gemma	Moderate	High	Clear	Fastest
Mistral	Excellent	Low	Strongest	Moderate
DeepSeek	Poor	Poor	Weak	Slowest
Qwen	Very Good	Very High	Very Good	Moderate

This comparison aims to assess the suitability of each model for log analysis tasks within predictive maintenance and operational monitoring contexts.

Gemma demonstrates moderate compliance with severity filtering, indicating that while it generally adheres to predefined severity constraints, it may occasionally include lower-severity contextual information. Nevertheless, Gemma achieves high language accuracy and clear structural organization, enabling concise and coherent analytical outputs. Additionally, it exhibits the fastest processing speed among the evaluated models, making it suitable for environments that prioritize rapid response times.

Mistral shows excellent severity compliance, reflecting strong adherence to severity-based filtering instructions. However, its language accuracy is comparatively low, as outputs occasionally mix languages and introduce ambiguity. Despite this limitation, Mistral provides the strongest structural organization, supporting deep and context-rich analytical reasoning, with a moderate processing speed.

DeepSeek performs poorly across most evaluation criteria. It exhibits low severity compliance, weak structural clarity, and poor language accuracy, resulting in outputs that lack operational relevance. Furthermore, DeepSeek records the slowest processing speed, which significantly limits its applicability for real-time or operational predictive maintenance tasks.

Qwen achieves very good severity compliance and very high language accuracy, producing outputs that are concise, consistent, and operationally meaningful. Its structural clarity is also rated very good, balancing analytical depth and readability. Although its processing speed is moderate, Qwen demonstrates the most balanced overall performance, making it the most suitable model for operational predictive maintenance pipelines. Overall, Qwen is most suitable for operational predictive maintenance pipelines.

DISCUSSION

The findings from the experimental trials indicate that the instruction-following behavior of the evaluated models differs considerably, and these differences have meaningful implications for operational deployment. Among the tested LLMs, Qwen exhibited the most consistent adherence to severity-based filtering rules and language constraints. This reliability suggests that Qwen offers a level of deterministic behavior that is advantageous for production environments where predictable outputs are essential.

In contrast, Mistral demonstrated stronger capabilities in contextual reasoning, often producing richer analytical interpretations of complex log entries. However, its occasional deviations from predefined language requirements imply that it is better suited for analyst-assisted scenarios, where human reviewers can refine the output as part of a collaborative workflow. Gemma, on the other hand, showed a clear advantage in processing efficiency. Its ability to



handle large volumes of log data with minimal latency makes it particularly useful in infrastructures involving multiple devices or high-frequency log ingestion, even if its interpretive depth is more limited compared with the other models.

DeepSeek showed the weakest performance across the evaluated measures. Its inconsistent handling of structured log formats and limited ability to apply severity-based reasoning indicate that it is not an appropriate candidate for predictive maintenance tasks that require precision and well-structured output. These observations reinforce the need to match model characteristics with operational demands rather than assuming broad applicability of any single LLM.

Overall, the comparative results provide evidence that LLMs can meaningfully contribute to predictive maintenance workflows, but their effectiveness depends heavily on selecting a model whose strengths align with the specific needs of the operational environment. The study's outcomes underscore the importance of evaluating both analytical accuracy and practical constraints—such as latency, output consistency, and compliance with predefined structures—when integrating LLMs into engineering or cybersecurity maintenance systems.

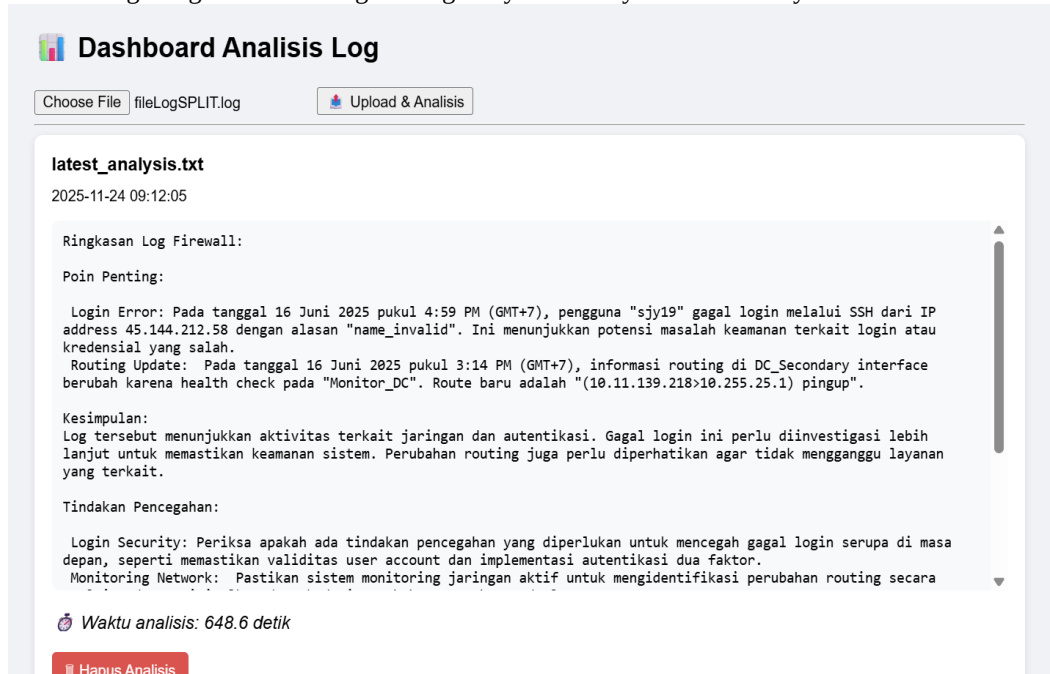


Fig. 2 Gemma2:2b result

Gemma 2b produced a complete and coherent analysis of the firewall log, and earlier assumptions of incomplete output were the result of viewing only a partial screenshot. Upon full evaluation, Gemma successfully identified high-severity events including failed administrative login attempts, routing changes, and SNMP-related communication failures. The model provided clear conclusions and preventive recommendations aligned with the analytical objectives defined in the prompt. Although the depth of its explanations is more concise than those of larger models, the analytical content remains accurate, relevant, and well organized.

With respect to severity filtering, Gemma demonstrates moderate adherence. It correctly emphasizes *alert* and *critical* events. Although it occasionally incorporates ancillary context from lower-severity entries. Nonetheless, these inclusions do not detract substantially from the analytical narrative. The model consistently outputs text in clear Indonesian with a structured presentation that includes key points, conclusions, and preventive actions. Completing the analysis in approximately 648 seconds, Gemma is the fastest model in the comparison, offering efficient processing alongside reliable analytical coverage.

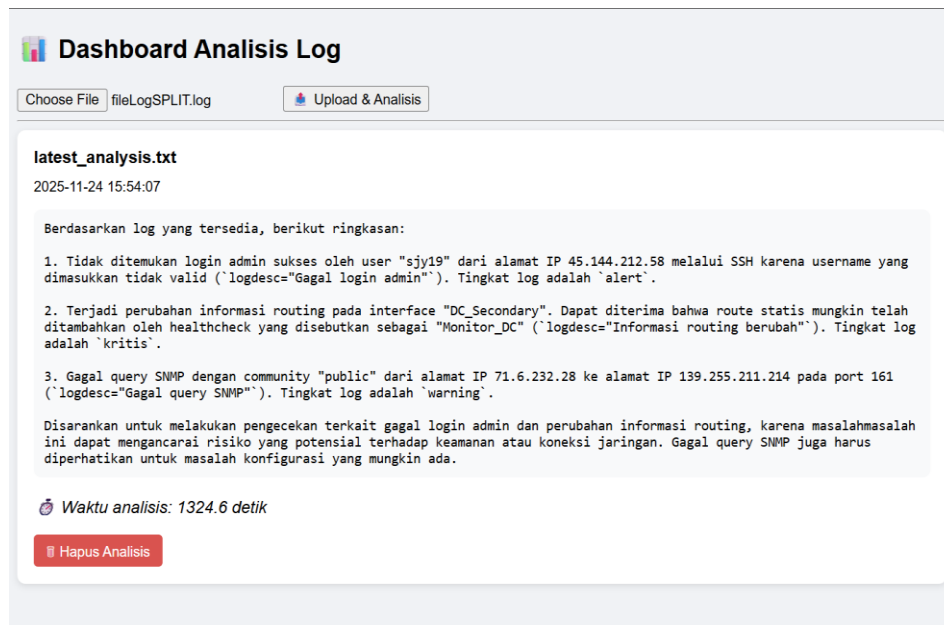


Fig. 3 Mistral result

Mistral delivers the most extensive and contextually rich analysis among the evaluated models. It demonstrates strong interpretive ability, accurately identifying and explaining relationships between events such as correlations between routing changes, system health-check behavior, and failed authentication attempts. The model also provides detailed and actionable preventive recommendations, making its output highly suitable for technical reporting and network maintenance decision-making.

However, two important limitations reduce its linguistics reliability. First, Mistral frequently mixes Indonesian and English within the same output, despite explicit instructions to use Indonesian exclusively. Second, this linguistic blending occasionally leads to semantic ambiguity, with certain sentences exhibiting unclear or inconsistent meaning. These issues, although not affecting structural coherence, impact the clarity and professional readability of the output.

In terms of severity filtering, Mistral performs exceptionally well, consistently focusing on *alert* and *critical* events while correctly ignoring lower-severity entries. The structural organization of the output is the strongest among all models tested, presenting key points, conclusions, and preventive actions in a logically ordered and comprehensive manner. The model required approximately 1.324 seconds to complete the analysis longer than Gemma, but yielding the most thorough and analytically nuanced interpretation.

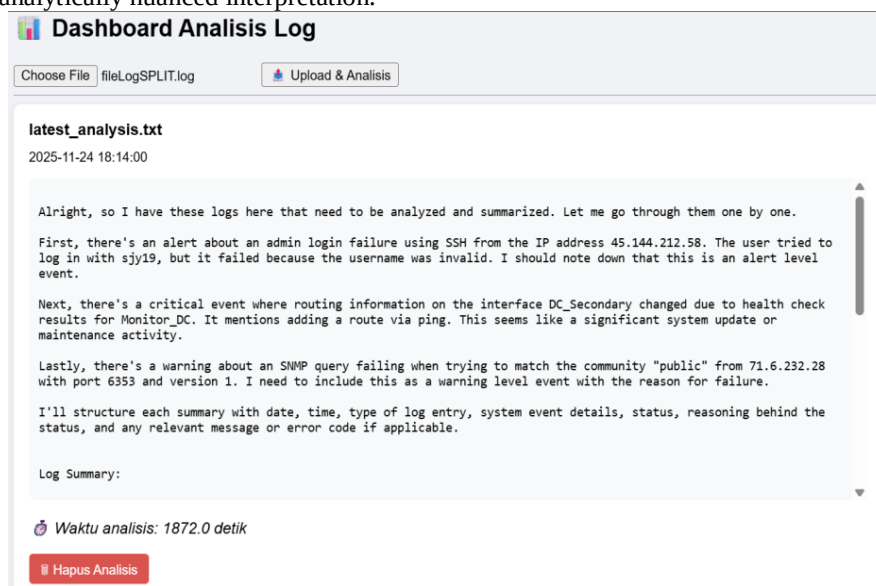


Fig. 4 DeepSeek-r1:7b

DeepSeek exhibits notable inconsistencies in following the constraints specified in the prompt. Despite explicit instructions to use Indonesian, the model generated its entire analysis in English, indicating a failure to adhere to linguistic requirements. Furthermore, DeepSeek demonstrates reduced information that should have been excluded. This undermines the model's suitability for controlled severity-based log analysis.

Although DeepSeek correctly identifies several key events such as failed login attempts and routing changes its analytical style tends to be more narrative and explanatory rather than structured. The model frequently provides extended reasoning chains, which, while informative, diverge from the concise key-point structure required for operational log analysis. The lack of structural adherence results in output that is less accessible and less aligned with professional reporting standards.

DeepSeek also exhibits the lowest computational efficiency among all models tested, requiring approximately 1.872 seconds to complete analysis. Given its long processing time, inconsistent language, weak severity filtering, and deviation from the instructed output format, DeepSeek is not an ideal candidate for automated firewall log analysis in an operational environment.

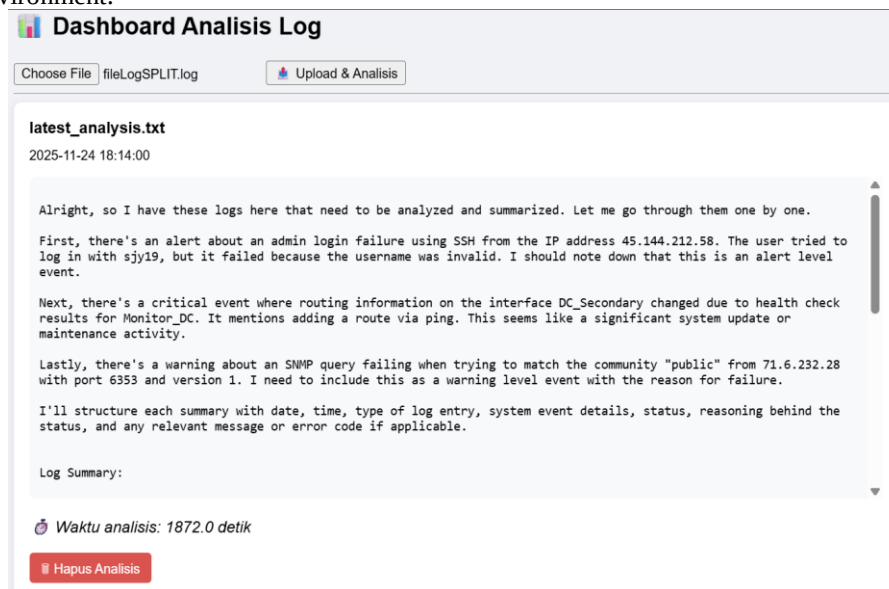


Fig. 5 Qwen2.5-coder:7b

Qwen 2.5 Coder 7B produces analysis that is concise, accurate, and consistently aligned with the prompt's requirements. The model identifies the essential high-severity events—failed login attempts, routing changes, and SNMP access issues—with clarity and precision. While its explanations are more succinct than those of Mistral, they remain sufficiently detailed to support operational decision-making in network security contexts.

Qwen demonstrates strong adherence to severity filtering instructions, focusing appropriately on *alert* and *critical* entries while omitting analysis of lower-severity log lines. The linguistic output is consistently delivered in Indonesian, and the structure is clear, consisting of well-defined sections for key findings, conclusions, and preventive recommendations. Although the analytical depth is not as extensive as Mistral's, Qwen's stable formatting and reliable adherence to instructions make it highly suitable for automated log-analysis scenarios.

The model completes the analysis in approximately 1,133 seconds, placing it between Gemma and Mistral in terms of processing speed. Its balance of clarity, structure, and instruction compliance positions Qwen as one of the most practically reliable models for Indonesian-language firewall log analysis.

The figure above depicts the comparative analysis outcomes generated by each LLM included in this study. It also summarizes the corresponding execution times required by the models when processing an identical prompt for log-file analysis. These measurements provide a standardized basis for evaluating the analytical efficiency and performance characteristics of each LLM

Table 2. Comparison between LLM

AI Model	Severity Filtering Compliance	Analysis Quality	Language Consistency	Structural Clarity	Analysis Time	Key Notes
Gemma 2B	Moderate	Complete, accurate, and coherent; concise compared to larger models	Fully consistent Indonesian	Clear and well-organized	648 seconds	Fastest model; includes minor lower-severity context but maintains analytical relevance

Mistral	Excellent	Most comprehensive and context-rich interpretation	Inconsistent mixes Indonesian and English; occasional unclear meaning	Strongest structure among all models	1,324 seconds	Deep analytical reasoning but reduced linguistic reliability
DeepSeek	Poor	Identifies events but lacks relevance and structure; narrative-heavy	Fails language requirement (English output)	Weak overall structure	1,872 seconds	Slowest model; ignores filtering instructions; not suitable for operational analysis
Qwen 2.5 Coder 7B	Good	Concise, accurate, and operationally meaningful	Consistent Indonesian	Well-structured and stable	1,133 seconds	Balanced and reliable; not as deep as Mistral but highly practical

Table 2 presents a comparative evaluation of four large language models (LLMs)—Gemma 2B, Mistral, DeepSeek, and Qwen 2.5 Coder 7B—based on their performance in firewall log analysis tasks. The comparison focuses on six key dimensions: severity filtering compliance, analysis quality, language consistency, structural clarity, analysis time, and key operational characteristics.

The results indicate that Gemma 2B demonstrates strong efficiency, achieving the shortest analysis time while maintaining coherent and accurate summaries. Although its severity filtering compliance is categorized as moderate, the model consistently produces structured outputs in Indonesian and preserves analytical relevance, making it suitable for resource-constrained or time-sensitive environments.

Mistral exhibits the highest level of analytical depth and contextual richness, particularly in interpreting complex log events. However, its tendency to mix Indonesian and English reduces language consistency and may introduce ambiguity in operational settings. Despite having the strongest structural clarity among the evaluated models, Mistral requires longer processing time, which may limit its applicability in real-time analysis scenarios.

In contrast, DeepSeek shows limitations across multiple dimensions. While it is capable of identifying log events, its outputs lack structural coherence and relevance to the specified severity filtering instructions. Additionally, the model fails to adhere to the required language constraints and records the longest analysis time, indicating that it is less suitable for practical firewall log analysis tasks.

Qwen 2.5 Coder 7B offers a balanced performance profile, combining concise and accurate analysis with stable structural organization and consistent use of Indonesian. Although its analytical depth does not reach the level of Mistral, Qwen provides reliable and operationally meaningful results within a reasonable processing time, making it a practical choice for production-oriented environments.

Overall, Table 2 highlights the trade-offs between analytical depth, language reliability, structural clarity, and processing efficiency across different LLMs, emphasizing the importance of model selection based on specific operational requirements.

CONCLUSION

This research provides a comprehensive comparative analysis of four LLMs for predictive maintenance of Fortinet firewall devices. Qwen 2.5-Coder 7B is recommended as the most reliable engine for operational use, whereas Mistral is suitable for deep analysis requiring human oversight. Gemma provides speed benefits, and DeepSeek should be avoided.

Future work should evaluate larger datasets, integrate closed-source models such as ChatGPT and Gemini, and explore fine-tuning LLMs for security-specific logs.

ACKNOWLEDGMENT

I would like to express my sincere appreciation to Faqih Murobbie for his invaluable assistance throughout the research process, to Rusmanto for his insightful guidance and mentorship as my research advisor, and to Ahmad Rio Adriansyah for his valuable contributions as a research discussion partner. Their support has greatly contributed to the completion of this study.

REFERENCES

- Brown, T. R. (2022). Predictive modeling for reliability in software-defined networks. *Journal of Network Systems*, 14(2), 85–102.
- Chen, M., & Al-Khatib, A. (2024). Interpretable large language models for high-volume network telemetry analysis. *International Journal of Cyber Analytics*, 9(1), 33–52.
- Hassan, R., & Rahman, S. (2022). Transformer-based anomaly detection in enterprise firewall environments. *Computers & Security*, 118, 102736.



- Kim, H. S. (2023). Evaluating ChatGPT for anomaly detection in heterogeneous network systems. *Journal of Information Security Research*, 8(3), 211–225.
- Kumar, V. (2022). Machine learning-based predictive maintenance for networked industrial systems. *IEEE Transactions on Industrial Informatics*, 18(7), 5123–5134.
- Kumar, S., & Verma, P. (2021). A survey of intelligent predictive maintenance frameworks for network infrastructures. *Journal of Intelligent Systems*, 31(4), 453–470.
- Li, Q., & Chen, Y. (2021). Semantic-aware models for anomaly detection in distributed environments. *ACM Transactions on Autonomous Systems*, 16(4), 1–20.
- Nguyen, P., & Foster, J. (2023). Assessing open-source LLMs for cyber-operations automation. *IEEE Access*, 11, 53429–53445.
- Ortega, L., & Ramos, J. (2023). Generative model-based causal inference for failure analysis in virtualized infrastructures. *Journal of Cloud Systems Engineering*, 12(1), 67–84.
- Patel, S., & Morgan, D. (2022). Enhancing SOC alert triage using open-source language models. *Journal of Cybersecurity Intelligence*, 5(4), 99–118.
- Silva, R., Santos, J., & Oliveira, P. (2023). Hybrid deep-learning and LLM-based approaches for predictive network traffic modeling. *Computers & Electrical Engineering*, 105, 108563.
- Wang, J. (2020). Predictive maintenance in industrial IoT using machine learning techniques. *Journal of Industrial Monitoring*, 6(1), 45–59.
- Wu, Y. (2023). GPT-based cloud log parsing for improved contextual inference. *IEEE Transactions on Cloud Computing*, 12(3), 188–202.
- Yadav, R. (2021). Anomaly detection in edge networks using advanced deep-learning models. *Journal of Edge Computing Research*, 4(2), 77–93.
- Zhang, L. (2024). Benchmarking large language models for cybersecurity log analysis. *IEEE Transactions on Information Forensics and Security*, 19, 560–575.