Contents

[Chapter 4: Analysis of Results and Discussion 2](#_Toc207829777)

[4.1 Introduction of the Analysis and Discussion 2](#_Toc207829778)

[4.2 Presentation of Findings 3](#_Toc207829779)

[4.2.1 Test Case 1: Broad Prompt 3](#_Toc207829780)

[4.2.2 Test case 2: Mid prompt 7](#_Toc207829781)

[4.2.3 Test Case 3: Specific Prompt 12](#_Toc207829782)

[4.3 Cross-Reference Analysis with Other Studies 17](#_Toc207829783)

[4.4 Overall Test Cases Assessment 19](#_Toc207829784)

[References 21](#_Toc207829785)

# Chapter 4: Analysis of Results and Discussion

## 4.1 Introduction of the Analysis and Discussion

This research hypothesised that large language models (LLMs) could be used to detect security misconfigurations in Cisco IOS configurations when assessed against the CIS Benchmarks. By employing a structured evaluation framework built on controlled datasets and carefully engineered prompts, the study aimed to measure both the accuracy and reliability of GPT-4o in identifying benchmark-aligned errors as well as common configuration mistakes. The primary objective was to determine whether prompt engineering could be used to guide the models toward more consistent and accurate compliance assessments.

As part of this investigation, model behaviour was also examined under different temperature settings, Increasing the temperature often led to hallucinations, where the model fabricated information, added commands that were never specified or ignored key instructions altogether, Conversely, lowering the temperature made the model overly rigid, producing outputs that strictly followed the instructions but frequently inserted “UNKNOWN” placeholders where variation or contextual adaptation was required. These observations highlighted that some degree of randomness is necessary for balanced performance, as overly deterministic outputs limited completeness, while excessive randomness compromised reliability.

Additionally, the research aimed to examine how varying levels of prompt specificity (Broad, Mid and Specific) shaped the model’s performance, and to compare the outcomes across multiple protocol domains. This study also aimed to capture qualitative patterns in the model’s responses, such as recurring misclassifications or hallucinations, to better understand their interpretative strengths and limitations. In the following sections, the analysis will present the results of these evaluations, compare model performance across prompt types and protocols, and critically discuss the implications of these findings for the integration of LLMs into professional network auditing and compliance workflows.

## 4.2 Presentation of Findings

### 4.2.1 Test Case 1: Broad Prompt

The Broad prompt was designed as the baseline test, requiring the model to review configurations without explicit reference to the CIS Benchmarks. Its purpose was to measure GPT’s ability to apply general networking knowledge when identifying misconfigurations and Mistype errors. The results indicated limited effectiveness in detecting benchmark aligned issues, with stronger but still inconsistent performance in identifying simple Mistype errors.

Chart 4.1: Broad Prompt Misconfiguration Detection Results

|  |  |  |  |
| --- | --- | --- | --- |
| **AAA** | | | |
| **Number of Errors**  **Present** | **1 Error Detected** | **2 Errors Detected** | **3 Errors Detected** |
| 1 | 5/5 | - | - |
| 2 | 1/5 | 1/5 | - |
| 3 | 4/5 | 0/5 | 0/5 |
| **PP Score:** | | | 40% |
| **EIGRP** | | | |
| 1 | 0/5 |  |  |
| 2 | 1/5 | 0/5 |  |
| 3 | 0/5 | 0/5 | 0/5 |
| **PP Score:** | | | 3% |
| **OSPF** | | | |
| 1 | 1/5 |  |  |
| 2 | 2/5 | 3/5 |  |
| 3 | 1/5 | 2/5 | 1/5 |
| **PP Score:** | | | 56% |
| **RIP** | | | |
| 1 | 0/5 |  |  |
| 2 | 2/5 | 0/5 |  |
| 3 | 1/5 | 0/5 | 0/5 |
| **PP Score:** | | | 10% |
| **Total PP Score:** | | | 28% |

Table 1: Broad Prompt Individual Results

|  |  |  |  |
| --- | --- | --- | --- |
| Protocol | Mistypes Not Detected | Mistypes Detected | PP Score |
| AAA | 2/5 | 3/5 | 60% |
| EIGRP | 1/5 | 4/5 | 80% |
| OSPF | 1/5 | 4/5 | 80% |
| RIP | 1/5 | 4/5 | 80% |
| **Total PP Score:** | | | 75% |

Table2: Broad Prompt Mistype Detection Results

Chart 4.1 and Tables 1 summarises the results across all four protocols. Out of 120 benchmark-related errors, only 33 were detected, resulting in a total PP Score of 28%. Accuracy varied significantly by protocol. OSPF achieved the highest PP Score of 56%, followed by AAA at 40%, while RIP and especially EIGRP performed poorly, with scores of 10% and 3% respectively. By contrast, Mistype errors were detected resulting in a total PP Score of 75%. Accuracy varied between protocol with EIGRP, OSPF and RIP all having PP Scores of 80% while AAA scored a PP Score of 60%. This indicated that GPT was more reliable at identifying basic syntactic mistakes than at enforcing benchmark compliance.

|  |  |
| --- | --- |
| **AAA Protocol Errors** | |
| **Errors** | **Errors Detected** |
| Service password-encryption | 3/3 |
| Enable secret | 1/1 |
| Aaa accounting system | 3/5 |
| Aaa authorization exec | 2/3 |
| Aaa authorization reverse-access | 0/4 |
| Aaa accounting commands 15 | 0/1 |
| Aaa new-model | 0/2 |
| Aaa authorization config-commands | 1/2 |
| Aaa authorization network | 0/2 |
| Line vty | 2/2 |
| **EIGRP Protocol Errors** | |
| **Errors** | **Errors Detected** |
| Router eigrp | 0/2 |
| Key-chain | 0/4 |
| Key | 0/3 |
| Key-string | 0/1 |
| Ip authentication mode eigrp md5 | 0/4 |
| Ip authentication key-chain eigrp | 1/9 |
| Passive-interface | 0/7 |
| **OSPF Protocol Errors** | |
| **Errors** | **Errors Detected** |
| Router ospf | 1/2 |
| Area authentication message-digest | 5/9 |
| Ip ospf message-digest-key | 10/19 |
| **RIP Protocol Errrors** | |
| Router rip | 0/4 |
| Version 2 | 0/2 |
| Ip rip authentication mode md5 | 0/4 |
| Ip rip authentication key-chain | 1/2 |
| Key-chain | 1/3 |
| network | 0/4 |
| Key-string | 1/1 |
| Passive-interface | 0/2 |
| redistribute | 0/4 |
| Maximum-paths | 0/3 |
| Offset-list | 0/1 |

Table 3: Broad Error Detections

To better understand recurring strengths and weaknesses, the configurations were further analysed to identify specific requirements that were either consistently detected or frequently overlooked. Table 3 shows that certain fundamental commands, “service password-encryption” in AAA was always detected with a score of 3/3 detections and “ip ospf message-digest-key md5” in OSPF with a score of 10/19 detections, were reliably flagged when missing, suggesting that the model could recognise high-level authentication mechanisms and simple per-interface security features. EIGRP and RIP had constantly not detected misconfigurations in these protocols for this test. Table 3 also highlights critical requirements that the model struggled with, “aaa authorization reverse-access” in AAA is one of them with a score of 0/4 detections and “area authentication message-digest” in OSPF with a score of 5/9 detections. These are more structural, policy-driven requirements, and the model often failed to detect their absence despite their central importance to CIS compliance. EIGRP and RIP had too low of detection rates to produce meaningful patterns.

Another type of analysis that was done considered the references that GPT cited when justifying its outputs. To explore this, three configurations from each protocol were sampled, and all references mentioned in the model’s reasoning were documents. As shown in Table 4, GPT frequently cited Cisco documentation, CIS Benchmarks and NIST guidelines, occasionally drawing on references such as NSA, RFC standards, DISA STIGs or real-world application examples. While the diversity of references suggests exposure to authoritative material, their use was inconsistent and not always aligned with the errors under review. Cisco, CIS and NIST were by far the most commonly used, indicating that GPT anchored its reasoning on broadly recognised security sources, but without the precision required for consistent compliance auditing.

|  |  |
| --- | --- |
| Reference Source | Count |
| Cisco References | 12/12 |
| CIS References | 11/12 |
| NIST References | 10/12 |
| Real-world applications Referenced | 5/12 |
| NSA References | 2/12 |
| RFC References | 2/12 |
| DISA STIG References | 1/12 |

Table 4: Most common references used by Broad Prompt

|  |
| --- |
|  |

Figure 1: Broad Prompt Output Example

In terms of qualitative patterns, the model frequently produced short and list-like responses as shown in Figure 1 rather than detailed checklists. While it often flagged certain recurring issues, it consistently failed to detect critical requirements in protocols as shown in Table 7. Overall, the Broad prompt demonstrated the weakness of unguided prompting. While capable of identifying obvious typos and occasional misconfigurations, its detection of benchmark-aligned errors was inconsistent and unreliable across protocols. These findings establish a baseline for comparison with the Mid and Specific prompts, where addition guidance was introduced.

### 4.2.2 Test case 2: Mid prompt

The Mid prompt introduced explicit reference to the CIS Benchmarks but did not provide excerpts of the standards themselves. Instead, the model was instructed to review the configurations “according to CIS Benchmarks”, requiring them to rely on any embedded knowledge of the framework gained during training. This test case therefore measured whether mentioning the CIS Benchmarks within the prompt alone improved detection accuracy compared to a purely unguided assessment.

Chart 4.2: Mid Prompt Misconfiguration Detection Results

|  |  |  |  |
| --- | --- | --- | --- |
| **AAA** | | | |
| **Number of Errors**  **Present** | **1 Error Detected** | **2 Errors Detected** | **3 Errors Detected** |
| 1 | 4/5 | - | - |
| 2 | 3/5 | 0/5 | - |
| 3 | 2/5 | 1/5 | 0/5 |
| **PP Score:** | | | 37% |
| **EIGRP** | | | |
| 1 | 1/5 |  |  |
| 2 | 3/5 | 0/5 |  |
| 3 | 1/5 | 2/5 | 0/5 |
| **PP Score:** | | | 30% |
| **OSPF** | | | |
| 1 | 1/5 |  |  |
| 2 | 2/5 | 2/5 |  |
| 3 | 1/5 | 1/5 | 2/5 |
| **PP Score:** | | | 53% |
| **RIP** | | | |
| 1 | 1/5 |  |  |
| 2 | 2/5 | 1/5 |  |
| 3 | 0/5 | 0/5 | 0/5 |
| **PP Score:** | | | 17% |
| **Total PP Score:** | | | 33% |

Table 5: Protocol Mid Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Protocol** | **Total Prompts** | **No Mistypes Detected** | **Mistypes Detected** | **PP Score** |
| AAA | 5 | 1 | 4 | 80% |
| EIGRP | 5 | 0 | 5 | 100% |
| OSPF | 5 | 1 | 4 | 80% |
| RIP | 5 | 1 | 4 | 80% |
| **Total PP Score:** | | | | 85% |

Table 6: Mid Prompt Mistype Results

The results showed moderate gains in benchmark-related error detection as shown in Table 5. Out of 120 misconfigurations implemented, 39 were identified, resulting in a total PP Score of 33%. Accuracy differed by protocol, with OSPF achieving the highest PP Score of 53% followed by AAA at 37% and EIGRP at 30%. RIP remained the weakest domain with only 17% of errors detected. Mistype errors were handled more effectively than in the Broad test, with a PP Score of 85% across all protocols. Protocol-specific mistype scores ranged from 80% in AAA, OSPF and RIP to 100% in EIGRP, demonstrating strong consistency in identifying basic syntactic issues. A per-protocol breakdown of misconfigurations detected is presented in Chart 4.2.

As with the Broad case, further analysis was carried out to determine recurring strengths and weaknesses. Table 14 highlights commands that the Mid prompt reliably detected. These included “service password-encryption” in AAA with 3/3 detections, “key chain” in EIGRP with 3/4 detections, “ip ospf message-difest-key md5” in OSPF with 13/18 detections and “ip rip authentication key-chain” in RIP with 2/2 detections. These results suggest that the introduction of benchmark framing improved recognition of certain authentication and encryption requirements, especially within OSPF.

|  |  |
| --- | --- |
| **AAA Protocol Errors** | |
| **Errors** | **Errors Detected** |
| Service password-encryption | 3/3 |
| Enable secret | 1/1 |
| Aaa accounting system | 0/5 |
| Aaa authorization exec | 3/3 |
| Aaa authorization reverse-access | 3/4 |
| Aaa accounting commands 15 | 0/1 |
| Aaa new-model | 0/2 |
| Aaa authorization config-commands | 1/2 |
| Aaa authorization network | 0/2 |
| Line vty | 2/2 |
| **EIGRP Protocol Errors** | |
| **Errors** | **Errors Detected** |
| Router eigrp | 0/2 |
| Key-chain | 1/4 |
| Key | 2/3 |
| Key-string | 0/1 |
| Ip authentication mode eigrp md5 | 2/4 |
| Ip authentication key-chain eigrp | 3/9 |
| Passive-interface | 1/7 |
| **OSPF Protocol Errors** | |
| **Errors** | **Errors Detected** |
| Router ospf | 0/2 |
| Area authentication message-digest | 5/9 |
| Ip ospf message-digest-key | 11/19 |
| **RIP Protocol Errrors** | |
| Router rip | 0/4 |
| Version 2 | 0/2 |
| Ip rip authentication mode md5 | 1/4 |
| Ip rip authentication key-chain | 2/2 |
| Key-chain | 1/3 |
| network | 0/4 |
| Key-string | 1/1 |
| Passive-interface | 0/2 |
| redistribute | 0/4 |
| Maximum-paths | 0/3 |
| Offset-list | 0/1 |

Table 7: Mid Error Detections

However, as shown in Table 7, the model continued to overlook several critical requirements. For AAA, “aaa accounting system” was inconsistently flagged with 0/5 detections, EIGRP exhibited weak detection of “passive interface”, with a score of 1/7 detections. In OSPF, the omission of “router ospf” was overlooked more often than not with a score of 0/2 and in RIP, missing “network” statements were consistently undetected with a score of 0/4. These gaps demonstrate that while Mid prompting increased structure and alignment with compliance concepts, reliability in detecting mandatory benchmark rules remained inconsistent

|  |
| --- |
|  |

Figure 2: Mid Prompt Output Example

Finally, an examination of the references cited by GPT in its explanations in Table 8 provided further insight into how it justified its outputs. Three configurations per protocol were sampled, and citations were documented. Cisco and CIS documentation were the most frequently invoked with them being used 12 times, following by NIST guidelines with 8 times, real-world application examples with 6 and NSA references with 4. The consistent appearance of Cisco and CIS sources suggest that the Mid prompt successfully anchored GPT’s reasoning to widely recognised compliance frameworks. However, the uneven use of supporting sources illustrates why detection accuracy remained inconsistent.

|  |  |
| --- | --- |
| Reference Source | Count |
| CIS References | 12/12 |
| Cisco References | 12/12 |
| NIST References | 8/12 |
| Real-world Applications Referenced | 6/12 |
| NSA References | 4/12 |

Table 8: Most common references used in Mid Prompt

Overall, the Mid Prompt demonstrated that referencing benchmarks within the instructions improves alignment with compliance-oriented reasoning and enhances detection of certain classes of misconfigurations. Nonetheless, accuracy gains were limited and several benchmark-critical omissions continued to be missed across all protocols.

### 4.2.3 Test Case 3: Specific Prompt

The Specific prompt represented the most constrained test case, limiting the model’s evaluation to a single protocol and attaching the corresponding section of the CIS Benchmark for reference. By providing explicit standards alongside the configuration, this test case measured the model’s ability to apply authoritative compliance rules within a defined scope.

Chart 4.3: Specific Prompt Misconfiguration Detection Results

|  |  |  |  |
| --- | --- | --- | --- |
| **AAA** | | | |
| **Number of Errors**  **Present** | **1 Error Detected** | **2 Errors Detected** | **3 Errors Detected** |
| 1 | 5/5 | - | - |
| 2 | 1/5 | 2/5 | - |
| 3 | 3/5 | 1/5 | 0/5 |
| **PP Score:** | | | 50% |
| **EIGRP** | | | |
| 1 | 2/5 |  |  |
| 2 | 1/5 | 4/5 |  |
| 3 | 1/5 | 2/5 | 2/5 |
| **PP Score:** | | | 73% |
| **OSPF** | | | |
| 1 | 3/5 |  |  |
| 2 | 3/5 | 2/5 |  |
| 3 | 0/5 | 2/5 | 3/5 |
| **PP Score:** | | | 77% |
| **RIP** | | | |
| 1 | 2/5 |  |  |
| 2 | 4/5 | 1/5 |  |
| 3 | 3/5 | 1/5 | 0/5 |
| **PP Score:** | | | 43% |
| **Total PP Score:** | | | 61% |

Table 9: Protocol Specific Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Protocol | Total Prompts | No Mistypes Detected | Mistypes Detected | PP Score |
| AAA | 5 | 1 | 4 | 80% |
| EIGRP | 5 | 0 | 5 | 100% |
| OSPF | 5 | 0 | 5 | 100% |
| RIP | 5 | 2 | 3 | 60% |
| **Total PP Score:** | | | | 85% |

Table 10: Specific Prompt Mistype Results

The results shown in Tables 9 demonstrated considerably higher accuracy than the previous test cases. Out of 120 benchmark-related errors, 73 were detected, giving a total PP Score of 61%. Protocol-level performance varied, with OSPF achieving the highest accuracy of 77%, followed by EIGRP at 73%, AAA at 50% and RIP at 43%. Mistype detection was strong overall as shown in Table 10, with a combined PP Score of 85%, though individual protocol scores ranged from 60% in RIP to 100% in both EIGRP and OSPF. A per-protocol breakdown of misconfigurations detected is illustrated in Chart 4.3.

|  |  |
| --- | --- |
| **AAA Protocol Errors** | |
| **Errors** | **Errors Detected** |
| Service password-encryption | 3/3 |
| Enable secret | 1/1 |
| Aaa accounting system | 4/5 |
| Aaa authorization exec | 2/3 |
| Aaa authorization reverse-access | 0/4 |
| Aaa accounting commands 15 | 1/1 |
| Aaa new-model | 1/2 |
| Aaa authorization config-commands | 1/2 |
| Aaa authorization network | 1/2 |
| Line vty | 1/2 |
| **EIGRP Protocol Errors** | |
| **Errors** | **Errors Detected** |
| Router eigrp | 0/2 |
| Key-chain | 4/4 |
| Key | 2/3 |
| Key-string | 0/1 |
| Ip authentication mode eigrp md5 | 3/4 |
| Ip authentication key-chain eigrp | 9/9 |
| Passive-interface | 4/7 |
| **OSPF Protocol Errors** | |
| **Errors** | **Errors Detected** |
| Router ospf | 1/2 |
| Area authentication message-digest | 7/9 |
| Ip ospf message-digest-key | 15/19 |
| **RIP Protocol Errrors** | |
| Router rip | 2/4 |
| Version 2 | 0/2 |
| Ip rip authentication mode md5 | 4/4 |
| Ip rip authentication key-chain | 2/2 |
| Key-chain | 2/3 |
| network | 0/4 |
| Key-string | 1/1 |
| Passive-interface | 2/2 |
| redistribute | 0/4 |
| Maximum-paths | 0/3 |
| Offset-list | 0/1 |

Table 11: Error’s Found/Missed Total

Further inspection of detection patterns is summarised in Table 11, which highlights the key requirements that were consistently identified. For AAA “service password-encryption” in AAA was detected in all 3/3 cases, “ip authentication key-chain eigrp” was identified in 9/9 runs, “ip ospf message-digest key” in OSPF was flagged in 15/19 cases and “ip rip authentication mode md5” in RIP was always detected with a score of 4/4. These results confirm that providing benchmark excerpts helped the model anchor to certain critical CIS requirements more reliably.

Nevertheless, certain benchmark-mandated rules were still missed, as shown in Table 11. In AAA, “aaa authorization reverse-access” was overlooked in all 0/4 cases. RIP was particularly weak, with “network” consistently being missed in all 4 cases. No consistently missed requirements were observed in EIGRP and OSPF. These results highlight that even under conditions when authoritative CIS excerpts were supplied, GPT was not uniformly consistent in applying all rules.

|  |
| --- |
|  |

Figure 3: Specific Prompt Output Example

Qualitative analysis of the outputs revealed significant improvements compared to the Broad and Mid prompts. Responses where checklist-like, more structured and explicitly referenced CIS Benchmarks, as illustrated in Figure 3. Explanations were generally more detailed, precise and audit-oriented, resembling the style of professional compliance reviews. This indicated that grounding GPTs reasoning in benchmark excerpts reduced ambiguity and encouraged outputs that aligned more closely with compliance auditing practices.

However, analysis of GPT’s cited references shows that it did not exclusively rely on the CIS Benchmarks even when they were explicitly attached. As shown in Table 24, three configurations were sampled, and GPT’s references recorded. CIS had been referenced in all 12 configurations and Cisco in 11 of them which were the most commonly referenced, followed by NIST with 8 references, real-world application sources with 4 and DIS STIG guidelines referenced 2 times. This demonstrates that although the prompt specifically instructed GPT to use the CIS Benchmarks, it continued to integrate information from a range of external sources. While Cisco and CIS remained dominant, the presence of other frameworks such as NIST and DISA STIG suggests that the model blended attaches standards with embedded training data, which may partly explain its inconsistent application of certain rules.

|  |  |
| --- | --- |
| Reference Source | Count |
| CIS References | 12/12 |
| Cisco References | 11/12 |
| NIST References | 8/12 |
| Real-world Applications References | 4/12 |
| DISA STIG References | 2/12 |

Table 24: Most common references used in Specific Prompt

Overall, the Specific prompt provided the strongest results across all three test cases, demonstrating the benefits of supplying authoritative benchmarks within the prompt. While detection accuracy improved significantly, persistent inconsistencies and reliance on external references underscore the limitations of current LLMs in fully adhering to strict compliance requirements.

## 4.3 Cross-Reference Analysis with Other Studies

The findings of this study align closely with those reported by Sare and Debono [21], particularly in the effect that benchmark guidance has on LLM performance. In this research, the Broad prompt produced limited accuracy, with an overall PP Score of 28% for misconfiguration detection, whereas accuracy increased substantially when benchmark excerpts were introduced in the Specific case, reaching 61%. A similar trend was documented by Sare and Debono, who reported that “GPT-4’s response accuracy rate is 75%. Compared to GPT-4’s performance from their Recorded results, without CIS benchmark document provided, which has an accuracy rate of 40.8%, this shows an improvement of 34.2%” [21]. Her work also demonstrated that baseline zero-shot prompting without CIS reference produced accuracy rated of only 40.8% for GPT-4 and 26.3% for GPT-3.5, highlighting the same challenge observed in this study, where unguided prompts struggled to consistently detect benchmark-aligned errors. Furthermore, Sare and Debono observed that when prompts explicitly referenced CIS benchmarks, “There is a 10% improvement in the ‘Cross-reference with CIS Benchmarks’ category” [21], which is consistent with the improvements noted here when using benchmark-guided prompts. Together, these parallels reinforce that while LLMs demonstrate some capacity for detecting vulnerabilities under general prompts, their accuracy is markedly enhanced when grounded in explicit compliance standards.

In addition to Sare’s study, the results reported by Cao et al. [16] further validate the central findings of this dissertation, the degree of guidance in prompt design directly governs the effectiveness of LLMs in fault detection. Cao et al. observed that when ChatGPT was tasked with identifying faulty programs using a minimal prompt template, it successfully detected only 27 out of 34 cases, However, when supplied with a more informative and structured prompt, performance improved dramatically, reaching 34 out of 34 detections. A similar pattern emerged in this dissertation’s evaluation of Cisco IOS configurations. Under the unguided Broad prompt, GPT identified 34 out of 120 misconfigurations. Introduction explicit references to CIS Benchmarks in the Mid prompt improved this slightly to 39 out of 120. The best results were achieved under the Specific prompt, which provided authoritative benchmark excerpts and narrowed the scope to a single protocol, allowing the model to identify 73 out of 120 misconfigurations. It was also reported that a similar pattern regarding the effect of prompt design on LLM performance was shown. It is shown that a basic, minimally guided prompt led ChatGPT to miss a meaningful portion of true faults and sometimes emphasize non-critical issues, whereas providing clearer task intention and context substantially improved performance and shifted the model towards a more functionally relevant direction, these results are consistent with this dissertations results, Unguided broad prompts tended to produce shorter, list-like outputs and occasional false positives, while benchmark-guided, protocol-scoped prompting yielded more structured compliance reasoning and higher accuracy in identifying materially important configuration issues.

The variability and inconsistencies observed in our three test cases align closely with results reported by Sobania et al. [4] on automatic program repair. In our setting, the models identified 34/120 misconfigurations under the Broad test (86 not found), 39/120 under the Mid test (81 not found) and 73/120 under the Specific, CIS-guided test (47 not found). In their benchmark, ChatGPT solved 19/40 buggy programs at baseline, a level comparable to other LLMs tested which are CoCoNut (19/40) and slightly below Codex (21/40), while traditional APR baselines solved 7/40 under a stricter generalization check, indicating that large portions of true faults remained unsolved without additional guidance [4]. Taken together, the specifics from both studies point to the same grading methods, without precise, task-oriented guidance, LLMs yield unstable and incomplete results (ChatGPT 19/40 for Sobania et al. and ours 34/120 under Broad), whereas clearer objectives and structured constraints materially improve out comes (ours 73/120 under Specific), with Sobania et al. further showing that targeted follow-up hints can raise ChatGPT’s solved set to 31/40 when additional problem information is supplied.

## 4.4 Overall Test Cases Assessment

Across all three test cases, a clear progression was observed in the model’s ability to detect benchmark-aligned errors, with accuracy increasing as prompts became more specific and guidance was made more explicit. The Broad prompt identified 34 out of 120 misconfigurations (28%), the Mid prompt rose slightly to 39 out of 120 (33%), while the Specific prompt achieved 73 out of 120 (61%). This steady improvement demonstrated that prompt engineering plays a pivotal role in shaping the reasoning process of large language models, validating the central hypothesis of this study that the precision and structure of prompts directly influence compliance assessment performance.

At the protocol level, results highlighted persistent patterns of strength and weakness. OSPF consistently achieved the highest scores across all prompt types, suggesting that the models possess stronger embedded knowledge and rule application capacity for this protocol compared to others. By contract, EIGRP and RIP were the weakest domains throughout, with frequent failures to detect critical commands. These discrepancies underscore the uneven distribution of protocol knowledge within the models and highlight the importance of scope-specific evaluation.

Qualitative analysis showed that as the prompts progressed, the model’s outputs shifted from short, vague and list-like responses to longer, compliance-oriented explanations and finally to structured checklist-like reasoning anchored in CIS standards. However, even when benchmark excerpts were explicitly attached in the Specific case, the models continued to reference external sources, including Cisco documentation, NIST guidelines and real-world practices. This behaviour indicated that while LLMs can be directed toward authoritative standards, they cannot be fully constrained to them, raising concerns for professional use cases where strict adherence is a must.

Taken together, these findings demonstrate both the potential and the limitations of LLMs in network configuration auditing. Prompt engineering was shown to markedly improve accuracy, yet detection remained inconsistent and heavily protocol-dependent, with many critical requirements still overlooked. While the Specific prompt achieved the highest accuracy, its 61% detection rate remains insufficient for production environments where full compliance is non-negotiable. These outcomes confirm that LLMs cannot yet replace formal auditing tools, but they do provide evidence that with carefully constructed prompts and controlled datasets, they can be leveraged as support tools in compliance checking workflows.

## References

[4] D. Sobania, M. Briesch, C. Hanna, and J. Petke, “An Analysis of the Automatic Bug Fixing Performance of ChatGPT,” in *2023 IEEE/ACM International Workshop on Automated Program Repair (APR)*, May 2023, pp. 23–30. doi: [10.1109/APR59189.2023.00012](https://doi.org/10.1109/APR59189.2023.00012).

[16] J. Cao, M. Li, M. Wen, and S. Cheung, ‘A study on Prompt Design, Advantages and Limitations of ChatGPT for Deep Learning Program Repair’, Apr. 17, 2023, *arXiv*: arXiv:2304.08191. doi: [10.48550/arXiv.2304.08191](https://doi.org/10.48550/arXiv.2304.08191).

[21] A. Sare and D. Debono, “The Dual-Edged Sword: The Impact of Large Language Models in Network Infrastructure Security,” Institute of Information and Communication Technology, Malta College of Arts, Science and Technology, 2025. Accessed: Aug. 30, 2025. [Online]. Available: <https://www.scitepress.org/PersonProfile.aspx?PersonAccountID=SPMqmL3tsi8=&t=1>

‌