**MCIS – Data Engineering Capstone Project**

**Youngstown State University**

**AI Enhanced Configuration Optimization and Security for Network Management and Analysis**

Author: Daniel P Yoder

Faculty Coach: Mr. Todd Jones

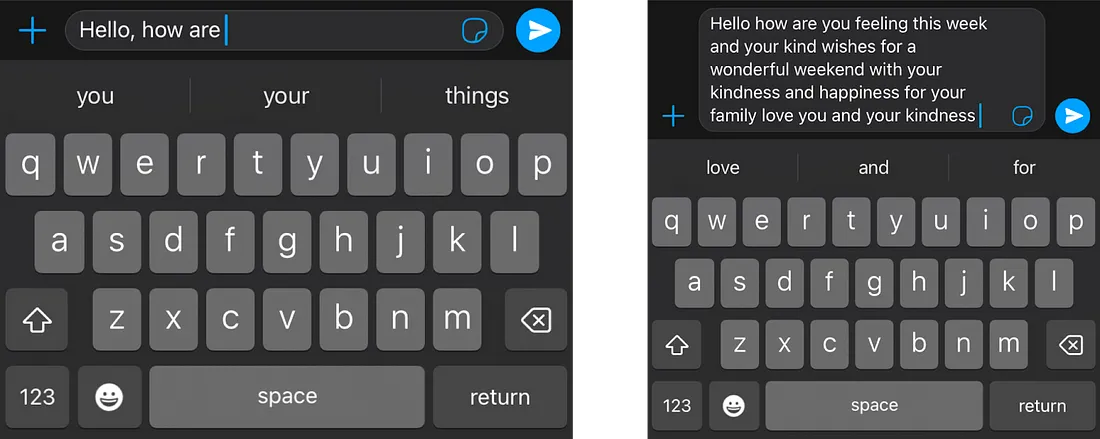
Capstone Coordinator: Dr. Feng (George) Yu

**Introduction:**

With the increasing complexity of network infrastructure and modern cybersecurity threats emerging, organizations are facing increasingly difficult challenges in optimizing and securing device configurations and responding to large amounts of alerts effectively. Cybersecurity threats have become more common and large scale, making it increasingly difficult for information security teams to stay afloat during peak activity hours. Traditional automation is becoming a more manual task with the increased complexity and attack surfaces in modern organizations, and a need has emerged for Artificial Intelligence (AI) to be used to supplement manual configuration baselining and log analysis. Large Language Models (LLMs) have emerged as a useful tool to automate and supplement baseline analysis of network/appliance configuration and to enhance incident response by identifying anomaly trend data automatically without manual intervention. These models have either been pre-trained for this specific purpose or can use data sources from the public internet to recommend optimizations and recognize common alert patterns. Using natural language processing and machine learning, LLMs are capable of identifying misconfigurations, making recommendations, and providing real-time insight into logs, security alerts, and their patterns. This paper explores the application of LLMs in network configuration and security, focusing on their role in analyzing device configurations, improving response mechanisms to security threats, and testing how accurate LLMs can be to take over the heaving lifting with baseline configuration and alert review.

**Background:**

Large Language Model technology (LLMs) is emerging into the public eye after the enormous success of popular LLMs like OpenAI’s ChatGPT, Google Gemini, Microsoft CoPilot and X’s Grok. Largely, these models are able to take even the simplest of prompts and generate complex responses with varying levels of accuracy. LLMs work by taking large sets of human-readable data and leveraging machine-learning techniques to learn how to respond to human prompts. Specifically, they use something called the transformer model. At simplest terms, the model doesn’t create sentences all at once, writing a sentence wholistically to reflect said idea. Instead, it starts with a prompt of context and one word (perhaps, ‘the’) and then it will use deep learning prediction techniques based on the data that the model has been trained on to determine what word should be next (perhaps, ‘car’). You may think that this is like word association and prediction on your phone as pictured below, but as you can see on the right graphic, the full sentence constructed makes little to no ‘contextual’ sense and seems robotic. What separates LLMs from this type of text generation is *context*.



(Amanatulla 2023)

Most LLMs work by using the transformer model. This model is excellent at maintaining context when prompted. While the model will generate text one word at a time, this works well with context from the dataset and prompt being maintained to provide an ‘in-context’ answer to any query or prompt. There is a lot of complexity beneath this layer of abstraction which includes; Tokenization, embedding, positional encoding, transformer blocks, and Softmax (Amanatulla 2023). However, the intent of this paper is not to examine the underlying mechanisms of LLMs, but to evaluate their practical applications and current accuracy in real-world cybersecurity contexts.

Various industries are integrating this technology into their operations, with noticeable impact. Healthcare, education, business and more are all seeing the benefits of LLMs (Thakare 2023). LLMs are being used to automatically respond dynamically to customer inquiries which can free up human customer support agents to handle more complex situations (Thakare 2023). They are being used in the software development sector to proofread code and write drafts for certain application blocks. Most importantly, LLMs can fit into existing systems easily. Now, complex chatbots are being inserted into popular social media websites like Facebook, Instagram, X, and more. While this application is simple, it has the capability to change how people interact online. Under some posts on Facebook, you can see an AI-generated summary of the content and even what people in the comments are saying about said post. All in all, LLMs are appearing more frequently in the daily life of millions of people.

This technology is ever evolving and expanding, and information technology companies are looking for ways to further integrate this technology into their daily workloads. Specifically, cybersecurity teams are beginning to leverage AI technologies in their tools. Recently, the University of Potsdam Germany reflected on the impact of LLMs on security, ‘Thanks to LLMs’ capability in breaking down complex natural language patterns, security experts are now enabled to explore more attack vectors in various contexts associated with textual data’ (Motlagh et al 2024). With more attack vectors and more logging data, IT teams are having difficulties reading and understanding large amounts of logging. Using LLMs, they can analyze datasets and begin to understand patterns of attack. Information security teams are empowered to understand large volumes of logging and automated alerting and remediation to prevent, protect and respond to cyberattacks. These technologies can also be applied to predict attack behavior and act proactively to prevent incidents. Cybersecurity firms and companies are further integrating LLMs into their defense applications in this way to offer other companies a way of managing the chaos that comes with modern AI-enhanced cybersecurity defense. These defensive measures are well-justified, especially given how cybersecurity professionals are already leveraging LLMs to stay ahead of evolving threats. However, it's also important to recognize that threat actors can exploit the same technologies to rapidly develop and launch more sophisticated, targeted attacks. In the same research paper by the University of Potsdam Germany, they mention this by saying, ‘.... with the continuous advancements of LLMs in cyber defense, it is crucial to acknowledge that these language models can also be leveraged by malicious actors. For example, LLMs can be misused by attackers to execute malware in target companies‘ (Motlagh et al 2024). The very tools that empower cybersecurity professionals can also be weaponized by threat actors, creating an ongoing ‘cat-and-mouse' dynamic where each side strives to outpace the other at an ever-accelerating pace. The accuracy of these AI chatbots is increasingly scrutinized as the fight between threat actors and security professionals continues. Being able to *accurately* produce insights and act on behalf of cybersecurity professionals is paramount. Security professionals must be able to trust the data and recommendations these systems produce, as their decisions often impact an organization’s overall safety and resilience. Inaccurate outputs could lead to misconfigurations, overlooked vulnerabilities, or even the failure to detect an active threat. This begs the question of which LLM model is best suited to be used as a base for this use. The basis of this paper is to test the accuracy of LLM models in a basic way, without significant organizational context, to see how they can pick up possible attack patterns and analyze baseline configurations for networking equipment.

**Problem Statement:**

Cybersecurity threats are causing security teams to struggle to keep up with the pace and scale of incoming threats, making it necessary to explore the application and accuracy of Artificial Intelligence (AI) and Large Language Models (LLMs) on baseline configurations and log analysis.

**Methodologies:**

To test the effectiveness of the most popular Large Language Models (LLMs), I will be using various baseline configurations of Cisco networking equipment to test if models can find misconfigurations and suggest improvements. The LLMs to be used will be Google Gemini 2.0 Flash, Microsoft Copilot, ChatGPT 4o, and Grok 3.

I plan to give each model 10 attempts on a set of clean and altered configurations to see if they can pick up the mistakes that I will document beforehand. For example, I will use a Cisco Catalyst 2960 configuration with no alterations and a configuration with bad security practices.

Secondly, I will provide a series of logs that are unrelated and related to a possible security incident to test pattern recognition. For example, I will use Microsoft Entra ID sign-in logs with suspicious activity and normal activity.

Each attempt will be given a score of ‘satisfactory’ or ‘unsatisfactory’ (pass/fail) on its ability to recognize what each log is showing.

**Experimentation and Results:**

The tests were performed on a clean dataset, where nothing is immediately wrong with the configuration, and a mixed dataset, where there was a randomized mix of clean and insecure configurations. These sets will be included in the appended documents.

The prompt for each configuration test is “Please review the attached switch configuration for any security errors: “

The prompt for each log/alert test is “Please review the attached sign-in logs for any threat patterns: “

The results of all testing can be found below:

Configuration Recognition (Clean)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Test 1 | Test 2 | Test 3 | Test 4 | Test 5 | Test 6 | Test 7 | Test 8 | Test 9 | Test 10 |
| ChatGPT | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass |
| Gemini | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass |
| Grok 3 | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass |
| Copilot | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass |

Log Threat Recognition (Clean)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Test 1 | Test 2 | Test 3 | Test 4 | Test 5 | Test 6 | Test 7 | Test 8 | Test 9 | Test 10 |
| ChatGPT | Fail | Fail | Fail | Pass | Pass | Pass | Pass | Pass | Pass | Pass |
| Gemini | Fail | Fail | Fail | Fail | Fail | Fail | Fail | Fail | Fail | Fail |
| Grok 3 | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass |
| Copilot | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass |

Configuration Recognition (Mixed)

Green = Clean

Red = Insecure

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Test 1 | Test 2 | Test 3 | Test 4 | Test 5 | Test 6 | Test 7 | Test 8 | Test 9 | Test 10 |
| ChatGPT | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass |
| Gemini | Fail | Fail | Pass | Fail | Pass | Pass | Pass | Fail | Pass | Pass |
| Grok 3 | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass |
| Copilot | Pass | Fail | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass |

Alert Recognition (Mixed)

Green = Clean

Red = Insecure

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Test 1 | Test 2 | Test 3 | Test 4 | Test 5 | Test 6 | Test 7 | Test 8 | Test 9 | Test 10 |
| ChatGPT | Fail | Fail | Fail | Fail | Fail | Fail | Fail | Fail | Fail | Fail |
| Gemini | Fail | Pass | Pass | Pass | Pass | Fail | Pass | Pass | Fail | Pass |
| Grok 3 | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass |
| Copilot | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass | Pass |

**Findings and Conclusion:**

The findings of the tests were positive and provided significant insight into how each model handled the subject matter. Some models showed extreme promise and detail expected while others had significant limitations.

In the tests evaluating LLMs' ability to identify misconfigurations in Cisco switch configurations, Microsoft Copilot and Grok 3 were consistently accurate across both the clean and mixed datasets. They were able to consistently recognize threats/vulnerabilities and provide suggestions for resolution. ChatGPT 4o was close but not as accurate, which passed most tests but struggled with the mixed dataset for log analysis. Lastly, Google Gemini 2.0 Flash showed mixed results. While it performed well on the clean dataset, its performance declined when handling the mixed data, where it failed to correctly identify misconfigurations in multiple instances.

All of this suggests that while most models are competent in parsing and validating standard, secure configurations, their analysis can become less effective when handling real-world, mixed-quality datasets. ChatGPT and Grok 3 demonstrated a stronger contextual understanding of realistic information, making them better suited for complex configuration baselining and log analysis. Overall, evidence shows that if an AI model is provided with organizational context, LLMs can be very useful for data analysis and baseline configuration optimization.

**Appendix/Sources:**

* Amanatullah. (2023, September 1). Transformer architecture explained. Medium. https://medium.com/@amanatulla1606/transformer-architecture-explained-2c49e2257b4c
* Motlagh, F. N., Hajizadeh, M., Majd, M., Najafi, P., Cheng, F., & Meinel, C. (2024, January 30). Large language models in cybersecurity: State-of-the-art. ArXiv.org. https://arxiv.org/abs/2402.00891
* Thakare, I. & Fagare, R. & Khuspe, S.. (2023). Impact of Chatgpt on Industry. Journal of Advanced Zoology. 44. 10.53555/jaz.v44iS8.3548.
* What is an LLM (large language model)? | cloudflare. (n.d.-a). https://www.cloudflare.com/learning/ai/what-is-large-language-model/
* What is LLM? - large language models explained - AWS. (n.d.-b). https://aws.amazon.com/what-is/large-language-model/
* Xai docs. Overview. (n.d.). https://docs.x.ai/docs/overview#welcome