Leveraging Large Language Models for OT Network Configuration Analysis

Leveraging Large Language Models for OT Network Configuration Analysis

1st Alberto Salvatore Colletto
AlphaWaves S.r.l.
Torino, Italy
a.colletto@awaves.it

2nd Mario Todaro
Politecnico di Torino
Torino, Italy
mario.todaro@polito.it

3rd Alessio Viticchié
AlphaWaves S.r.l.
Torino, Italy
a.viticchie@awaves.it

4th Alessandro Aliberti
Politecnico di Torino
Torino, Italy
alessandro.aliberti@polito.it

***Abstract*—Operational Technology (OT) networks face growing cybersecurity risks, yet applying best practice guidelines remains difficult—particularly in settings with limited cybersecurity expertise. This paper proposes a modular framework combining a Large Language Model (Llama3 8B Instruct), semantic search (FAISS), and structured prompting to assist in the analysis of OT configurations. The system extracts best practices from authoritative sources, generates standardized JSON templates for data collection, and leverages a chatbot assistant for compliance validation and mitigation guidance. Experimental results show moderate accuracy (60–66.67%), highlighting both the promise and current limitations of LLM-based security tools. The framework offers a foundation for enhancing automation, interpretability, and resilience in OT environments.**

***Index Terms*—Cybersecurity Automation, Operational Technology Security, Large Language Models.**

I. Introduction

As cyber threats increasingly target critical infrastructure, Operational Technology (OT) networks face distinct and complex security challenges. Unlike conventional IT systems, OT environments manage essential industrial processes, including power generation, manufacturing, and transportation, where security breaches can result in severe operational disruptions, safety risks, and economic losses [1]. The integration of cybersecurity best practices into OT networks is particularly challenging due to the presence of legacy systems, strict real-time operational constraints, and the intricacies of modern security frameworks [2].

To address these challenges, we present an innovative framework that automates the extraction and application of OT-specific cybersecurity best practices. Leveraging advanced natural language processing (NLP) techniques—such as sentence-transformer models and Facebook AI Similarity Search (FAISS), we dynamically construct a continuously updated knowledge base sourced from authoritative cybersecurity standards and guidelines. This ensures that OT systems remain aligned with the latest developments in the field. Our framework also incorporates a structured data acquisition methodology using standardized JSON templates, enabling systematic and scalable assessments of OT network configurations. At its core is an interactive, AI-powered security assistant based on Llama3 8B Instruct, which facilitates real-time validation of configurations, detection of potential vulnerabilities, and delivery of mitigation strategies specifically tailored to OT environments.

By strategically integrating automation, structured data representation, and AI-driven advisory capabilities, the proposed framework aims to strengthen the cyber resilience of OT networks. Moreover, it is designed to facilitate and streamline compliance with critical industry standards and regulatory frameworks, including NIST SP 800-82, IEC 62443, and ISO/IEC 27019 [3], thereby promoting both operational security and regulatory alignment.

This publication is partially part of the project funded under the National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 1.3-SERICS PE 00000014 – CUP F53C22000740007.

II. Background

A. Common misconfigurations and best practices

As cybersecurity threats grow in scale and complexity, securing OT environments remains a critical challenge. The National Institute of Standards and Technology (NIST) offers comprehensive guidance for improving the security posture of both IT and OT systems, emphasizing configuration management as a key defense strategy. Notably, NIST Special Publication 800-128, *Guide for Security-Focused Configuration Management of Information Systems*, outlines best practices for maintaining secure configurations and promptly addressing deviations that may introduce vulnerabilities.

Misconfigurations are among the most common and critical security risks in OT networks, often creating exploitable weaknesses for malicious actors [4]. Common examples include:

- *Poor Patch Management*: Failing to apply security updates leaves systems vulnerable to known exploits [5].
- *Default Credentials*: Using factory-set usernames and passwords creates an easy entry point for attackers [6].
- *Weak Access Controls*: Insufficiently restricted user permissions increase the risk of unauthorized access [7].

In OT environments, misconfigurations can lead to severe consequences, compromising system integrity and posing significant risks to operational continuity and human safety. Several high-profile incidents highlight the critical impact of such vulnerabilities. For example, in the Water and Wastewater Systems (WWS) sector, cyber adversaries exploited publicly accessible OT devices by leveraging default credentials and brute-force techniques, ultimately gaining unauthorized control over critical infrastructure components [8]. Similarly, in

2021, a ransomware attack on JBS Foods disrupted meat production across North America and Australia, causing significant supply chain disruptions and price fluctuations [9].

Adhering to cybersecurity frameworks such as NIST SP 800-82, IEC 62443, and ISO/IEC 27019 is essential for strengthening the security posture of OT networks and mitigating risks from misconfigurations and cyber threats. However, translating these comprehensive standards into practical measures remains challenging, particularly in environments where personnel possess strong industrial expertise but limited cybersecurity knowledge. The technical complexity of many guidelines often hinders implementation, especially in the absence of clear and accessible communication. Bridging this gap requires user-centric tools, intuitive training, and automated advisory systems that simplify compliance and support operational integration.

This study contributes to addressing this challenge by introducing a framework that enhances the communication of cybersecurity best practices in OT contexts. It supports non-expert users in understanding potential risks and, crucially, streamlines the collection and interpretation of key security-related information—enabling more effective threat mitigation while preserving operational continuity.

B. Large Language Models in Cybersecurity

Large Language Models (LLMs) have emerged as transformative tools for natural language understanding and generation, offering significant potential in the field of cybersecurity [10]. Models such as OpenAI's GPT series, Meta's Llama3, and BERT-based architectures are capable of processing extensive volumes of technical documentation, extracting actionable insights, and contributing to tasks such as threat identification, incident response, and risk mitigation.

Within the OT domain, LLMs can enhance the capabilities of security analysts by automating the generation of security reports, interpreting complex regulatory and procedural guidelines, and delivering real-time, context-aware recommendations [11]. Nonetheless, the utility of LLMs in such critical applications is contingent upon the relevance and quality of their training data, as well as their ability to accurately interpret and apply domain-specific cybersecurity knowledge.

1) Llama3: Llama3 is a state-of-the-art, open-weight LLM designed for high efficiency and accuracy across diverse natural language processing tasks. With a focus on reasoning, contextual understanding, and reliable response generation, it is well-suited for cybersecurity applications. In OT environments, Llama3 shows strong potential in processing security logs, interpreting threat intelligence, and supporting incident response [12].

Furthermore, Llama3's seamless integration with retrieval-augmented systems—such as FAISS significantly enhances its ability to deliver precise, context-aware answers drawn from large corpora of security documentation. This capability makes it an effective asset for knowledge-driven security applications in complex, real-time environments.

2) Sentence-BERT: Sentence-BERT (SBERT) is an advanced deep learning model specifically designed to generate semantically meaningful sentence embeddings, facilitating a range of natural language processing tasks including semantic similarity analysis, clustering, and information retrieval [13]. Built upon a Siamese network architecture, SBERT processes pairs of sentences in parallel through identical transformer models, enabling the effective capture of semantic relationships between textual inputs.

Through fine-tuning on benchmark datasets such as the Stanford Natural Language Inference (SNLI) and Multi-Genre NLI (MNLI) corpora, SBERT generates dense vector representations that can be efficiently compared using cosine similarity, enabling precise sentence-level understanding. Its computational efficiency is a key advantage: by supporting precomputed embeddings, SBERT enables fast similarity searches and scalable text analysis, making it ideal for real-time information retrieval and other NLP applications.

C. FAISS and Sentence-Transformer for Knowledge Retrieval

The Facebook AI Similarity Search library, when integrated with Sentence-Transformer models, provides a highly efficient solution for retrieving semantically relevant information from large-scale textual datasets [14]. Sentence-Transformers encode unstructured text into dense vector embeddings, which FAISS then indexes to enable rapid and scalable similarity search across high-dimensional data spaces.

This combined approach is particularly advantageous in cybersecurity contexts, where timely access to relevant information—such as security advisories, compliance documentation, and historical incident records—is critical. By leveraging FAISS alongside Sentence-Transformers, organizations can construct dynamic, searchable knowledge bases that support real-time identification of configuration anomalies and facilitate the recommendation of best practices tailored to OT environments.

III. Methodology

To operationalize the proposed framework, we developed a modular architecture structured around three key components: i) automated extraction of cybersecurity best practices, ii) dynamic generation of standardized configuration templates, and iii) an AI-driven assistant for interactive security analysis. These components, each addressing a critical stage in the workflow, are described in detail in the following.

The complete process, illustrated in Fig. 1, integrates essential technologies such as Sentence-Transformer models, FAISS-based semantic indexing, JSON-formatted data structuring, and a conversational interface powered by Llama3 8B Instruct. Cybersecurity documents—including standards, advisories, and internal guidelines—are initially transformed into dense vector embeddings via the all-MiniLM-L6-v2 model. The FAISS library then indexes these embeddings to enable high-speed semantic retrieval of relevant content.

The retrieved best practices are used to generate structured JSON templates that map configuration parameters to specific

security guidelines. To support users in interpreting and completing these templates, an AI assistant engages interactively, offering contextualized recommendations and requesting additional input where necessary.

This methodology ensures that OT network assessments are consistent, auditable, and anchored in authoritative references. Each stage of the pipeline, from document ingestion to real-time user interaction, provides continuous feedback that reinforces compliance monitoring and proactive threat mitigation.

A. Automatic Best Practices Extraction Tool

The automatic extraction of cybersecurity best practices represents a critical component of the proposed framework, enabling continuous updates and customization based on user-selected authoritative sources such as NIST guidelines, IEC standards, or internal organizational policies.

To process these sources, documents are first embedded into a semantic vector space using the `all-MiniLM-L6-v2` model. This lightweight sentence-transformer, derived from BERT, offers efficient performance through reduced dimensionality (i.e., 384 vs. 768) and mean pooling for length-invariant embeddings. Its contrastive learning optimization enhances its ability to detect semantic similarity across varied linguistic expressions. The resulting embeddings are indexed using FAISS for approximate nearest neighbour search in high-dimensional spaces. FAISS allows for fast, scalable retrieval based on semantic similarity, with built-in support for GPU acceleration to handle larger datasets and real-time querying. Best practices are extracted using the Llama3 8B Instruct model, integrated via the LangChain framework. LangChain enables the construction of a Retrieval-Augmented Generation (RAG) pipeline, where the `as_retriever` and `create_stuff_documents_chain` functions manage document access and structured prompting. Few-shot prompting guides the LLM to generate focused, standards-aligned recommendations.

To ensure transparency, the model is instructed to explicitly reference the origin of each extracted best practice, including the title of the source document. This traceability supports interpretability, regulatory alignment, and user trust.

B. Automated JSON Template Generation via LLM

A core component of the proposed architecture is the automated generation of a standardized JSON template, which facilitates a consistent, structured collection of network configuration data across all workflow stages. As the system does not assume prior knowledge of the user network, this step ensures the necessary input is captured to enable accurate evaluation against cybersecurity best practices.

The JSON template is generated using the same LLM (i.e., Llama3 8B Instruct) employed in the best practice extraction phase. Leveraging few-shot prompting, the model produces well-structured outputs guided by predefined examples. To ensure consistent formatting and deterministic behaviour, the generation process is controlled with a low-temperature setting (0.01), minimizing variability. The retrieval mechanism mirrors the previous stage, with relevant best practices accessed from the FAISS-indexed knowledge base to inform template construction.

Each generated JSON template field includes the following metadata:

- *field_name*: The name of the configuration parameter, expressed in snake case.
- *field_value*: The value to be provided by the user, defaulting to `Unavailable`.
- *field_best_practices*: A reference to the corresponding best practice.
- *field_description*: A contextual explanation of the field and the associated input requirement.

Once generated, the JSON template is completed by the user, who inputs the required configuration data. The `field_description` serves as contextual guidance, clarifying the intent and scope of each field to ensure accurate and consistent data entry. To support future automation, the `field_value` is initialized with the placeholder `'Unavailable'`, allowing for the possibility of automated completion in subsequent iterations. At present, data entry is facilitated through a Python script, which programmatically updates the template based on user-provided inputs.

This structured approach ensures interoperability between components and guides users in providing complete and relevant information, ultimately enhancing the accuracy and usability of the security evaluation process.

C. AI-driven Chatbot Assistant

In the initial development phase, particular emphasis was placed on designing a core component capable of analyzing network configurations and detecting potential misconfiguration. This functionality was considered foundational to the proposed architecture, which was subsequently extended with additional pipeline elements to enhance overall system capability.

Multiple implementation strategies were evaluated, including the use of semantic similarity metrics based on cosine similarity. However, this approach proved inadequate for the task, primarily due to its inability to account for negation and subtle semantic nuances. Embedding-based models often assign high similarity scores to semantically contradictory statements, exposing a critical limitation in their applicability to security-sensitive contexts. This issue stems from the inherent design of such models, which prioritize contextual proximity over precise semantic fidelity.

To further investigate this limitation, dimensionality reduction techniques—specifically Principal Component Analysis (PCA) were employed to visualize the spatial distribution of embedding vectors. The analysis revealed that sentences with opposing meanings were frequently mapped to adjacent regions in the vector space. This observation confirmed the inadequacy of simple similarity-based methods for tasks requiring fine-grained language understanding. It also underscored the

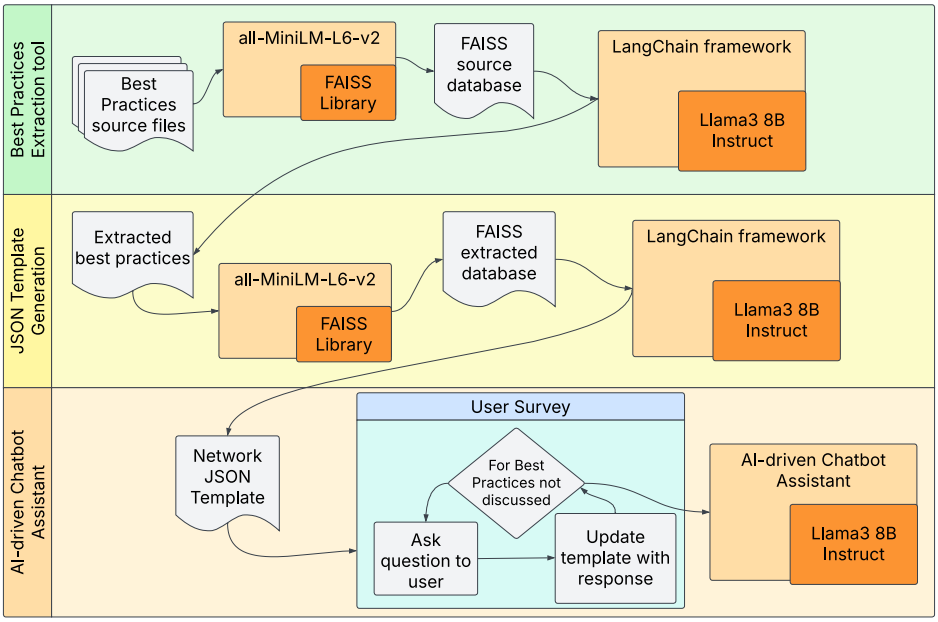

Fig. 1. Architecture of the proposed OT security assistant integrating semantic search, structured data collection, and LLM-based interactive analysis.

necessity for a more advanced, context-aware approach to accurately support security analysis and compliance assessment.

1) Integration of an LLM for Security Analysis: The challenges encountered during early design iterations catalyzed refining the analytical approach, ultimately leading to the integration of a LLM to support security analysis. Considering the complexity of cybersecurity tasks and the need for precise interpretation of domain-specific language, the Llama 3 Instruct model (8B parameters) was selected. This model strikes an effective balance between reasoning capability and computational efficiency, making it well-suited for on-premise deployment within enterprise environments. Its training through Reinforcement Learning from Human Feedback (RLHF) further enhances its ability to produce accurate, context-aware responses, surpassing the performance of conventional text-completion models.

The inclusion of the LLM also enabled the development of a chatbot component within the system architecture. This conversational interface supports both automated security analysis and real-time user interaction, allowing the system to request clarifications, collect additional input, and deliver targeted recommendations for mitigating identified vulnerabilities.

2) Basic Chatbot implementation: The LangChain library played a pivotal role in the implementation of the chatbot component, which supports both security analysis and interactive user engagement. Its modular architecture and extensive functionality enabled the realization of the system's design objectives with flexibility and precision. One of the library's core strengths lies in its support for constructing diverse processing chains, which facilitate contextualized interactions by passing structured prompts to the language model.

A critical feature of LangChain is its management of distinct message types, which help differentiate between conversational dialogue and task-specific automation. The primary message types used during implementation include:

- *Human Messages*: Represent user inputs, including questions or commands, often augmented with additional context to guide the model's response generation.
- *AI Messages*: Correspond to the model's replies, generated in response to user inputs based on the given context.
- *System Messages*: Define the model's operational behaviour before execution, specifying its role, response style, and any constraints to be followed.

System Messages, in particular, were instrumental in applying Prompt Engineering techniques, which are discussed in detail in the following. Their use ensured that the model adhered to predefined behavioural expectations, thereby improving consistency and reliability in the chatbot's responses.

3) Prompt Engineering: Prompt Engineering is a technique for guiding LLM behavior through carefully crafted inputs, without modifying the model's internal parameters. As a non-invasive method, it is especially useful in scenarios where fine-tuning is impractical. Achieving effective results requires well-designed prompts that clearly define tasks, expected outputs, and contextual boundaries. Modern frameworks like LangChain support this by embedding such instructions in System Messages, allowing precise control over the model's reasoning and responses.

Several established Prompt Engineering paradigms are commonly used to enhance model performance across tasks [15]:

- *Zero-Shot Prompting*: Relies solely on a well-structured prompt, without including example input-output pairs. The model draws on its pre-trained knowledge to generate relevant responses.
- *Few-Shot Prompting*: Incorporates a limited number of examples within the prompt to improve contextual un-

derstanding and response accuracy.
- *Chain-of-Thought (CoT) Prompting*: Guides the model through a step-by-step reasoning process, which is especially effective for tasks requiring logical inference or multi-step problem-solving.

In this architecture, a tailored variant of Few-Shot Prompting, referred to as Structured Output Prompting, was employed. Rather than providing explicit example pairs, the model is instructed to conform to a predefined response schema, ensuring consistency, interpretability, and alignment with the system's requirements. The following outlines the prompt strategy adopted for network configuration analysis:

- *No assumptions*: evaluate only the provided JSON data.
- *Missing or unclear data*: return "Cannot Verify" (never mark "Not Satisfied" without sufficient evidence).
- Explicitly request additional details if the data is incomplete.

The prompt design was not ad hoc but carefully engineered to guide the model toward consistent and contextually relevant behaviour. To align the model's outputs with the specific objectives of the cybersecurity task, the prompt explicitly defined the model's domain expertise, establishing its role as a cybersecurity assistant. The task itself was clearly articulated, accompanied by precise operational instructions, and all necessary contextual information was provided as parameters to ensure well-informed and accurate responses.

A structured output format was also specified within the prompt, delineating three distinct categories for evaluating each configuration item: i) *Best Practice Satisfied*, ii) *Best Practice Not Satisfied* and, iii) *Insufficient Information to Determine Best Practice Status*. This classification is critical, as it enables differentiated user feedback. In instances where a best practice is not met, the model includes an associated risk score, empowering users to assess the severity of the identified issue in the context of their network configuration.

To reinforce consistency and reliability, multiple constraints were embedded into the prompt. Particular emphasis was placed on handling incomplete or ambiguous input: if sufficient evidence is not available, the model is instructed to return an "Insufficient Information" status rather than speculating. Additionally, to minimize the risk of hallucinations or unsupported claims, the prompt explicitly restricts the model to operate solely on user-provided data, prohibiting the inclusion of external or inferred content.

4) Chatbot memory management: LLMs inherently lack persistent memory, meaning each request is processed statelessly, without awareness of prior interactions. To enable coherent dialogue and support context-dependent operations—such as incorporating additional information during iterative analysis—a memory management mechanism is essential to preserve conversational context.

To address this, the `manage_chat_history` function dynamically maintains the conversation history, preventing token overload by pruning the oldest messages. When the number of exchanges exceeds the threshold defined by `max_turns` (set to 1), the dialogue is summarized and compressed into a single `SystemMessage` that captures the essence of prior interactions.

To ensure continuity and relevance, the system retains the most recent user input, the corresponding model response, and the chat summary. This strategy provides a balance between maintaining short-term conversational coherence and controlling token usage, enabling the chatbot to deliver contextually accurate responses without accumulating excessive dialogue history.

IV. Experimental Discussion

To evaluate the effectiveness of the proposed methodology, a series of controlled experiments were conducted to assess the language model's ability to interpret and classify user inputs about cybersecurity best practices. The primary objective was to examine the model's capacity to accurately recognize, categorize, and reason over responses mapped to established security standards.

A representative subset of best practices was selected from a broader set of over sixty items tested in a laboratory environment. These practices represent foundational elements of cybersecurity strategy and include:

1) *Defense-in-Depth Strategy:* A multi-layered approach that implements overlapping controls to protect systems against a broad range of threats, thereby minimizing the impact of individual control failures.
2) *Monitoring and Logging:* Continuous observation and logging of system activities to facilitate real-time anomaly detection and post-incident forensic analysis.
3) *Incident Response Planning:* Structured procedures for identifying, managing, and recovering from cybersecurity incidents to minimize operational disruption.
4) *Authentication and Access Control:* Deployment of robust identity verification mechanisms and role-based permissions to prevent unauthorized access.
5) *Regular System Updates and Patching:* Timely application of software updates and security patches to remediate known vulnerabilities and reduce attack surfaces.
6) *Training and Awareness:* Ongoing educational initiatives aimed at equipping employees with the knowledge to recognize and respond to common cyber threats, such as phishing and social engineering.

For each of the selected practices, a range of hypothetical user responses was crafted to simulate realistic inputs with varying levels of clarity, completeness, and technical accuracy. The model was then tasked with classifying these responses according to their alignment with the corresponding best practice, identifying whether the input was compliant, non-compliant, or inconclusive.

While the full evaluation encompassed the complete set of practices, this section focuses on six core items selected for their relevance and diagnostic value in assessing the model's reasoning capabilities. The experimental findings provide insight into the model's strengths and current limitations in understanding and interpreting user compliance with cybersecurity protocols.

A. Evaluation of Model Performance

The model's performance across the selected best practices is summarized in Table I, with accuracy rates ranging from 60.00% to 66.67%. While these figures suggest a solid foundational capability, they also highlight notable limitations in terms of precision and consistency. It is important to emphasize that the evaluation was conducted using the Llama 3.2 Instruct model with 3B parameters—a relatively lightweight architecture. This constrained capacity may have limited the model's ability to accurately interpret nuanced, ambiguous, or partially correct user inputs.

A recurring issue observed during testing was the occurrence of false positives, wherein the model incorrectly classified non-compliant responses as compliant. This behaviour underscores the importance of enhancing the model's ability to differentiate between responses that fully satisfy a best practice and those that only partially address it. Such misclassifications, if uncorrected, could lead to a false sense of security in operational settings.

TABLE I
PERFORMANCES OF THE PROPOSED SOLUTION ACROSS SELECTED CYBERSECURITY BEST PRACTICES.

Best Practice Reference	# True Positives	# False Positives	% Correct Guess
BP-01: Defense-in-Depth	40	20	66,67%
BP-02: Monitoring and Logging	36	24	60,00%
BP-03: Incident Response	38	22	63.33%
BP-04: Access Control	39	21	65.00%
BP-05: System Updates and Patching	37	23	61,67%
BP-06: Training and Awareness	36	24	60,00%

These results reinforce the need for further refinement of the system. Integrating more advanced LLMs or hybrid agent-based reasoning frameworks could significantly enhance interpretability, reduce error margins, and improve the robustness of automated cybersecurity assessments—especially in high-assurance environments where precision is critical.

V. CONCLUSION

This study explored the integration of LLMs within a cybersecurity analysis framework, employing structured prompting, modular interaction via LangChain, and lightweight memory management to support interactive assessments of best practices. While the proposed architecture demonstrates potential in guiding user-driven evaluations, experimental results highlight key areas for improvement.

Model performance showed moderate accuracy, with correct classification rates between 60.00% and 66.67%. A notable issue was the generation of false positives, where non-compliant inputs were misclassified as compliant—raising concerns about reliability in critical operational contexts. Additionally, structured prompts, while improving consistency, introduced a rigidity that limited adaptability to partial or context-sensitive responses.

The reliance on user-provided input also revealed a vulnerability: incomplete or ambiguous data frequently led to inconclusive assessments. Similarly, the memory management approach, based on summarizing prior exchanges, reduced resource use but compromised contextual continuity during extended dialogues.

In conclusion, while LLMs offer a promising foundation for enhancing cybersecurity workflows, their limitations suggest they should be used with human oversight. Future work will explore hybrid models that combine LLM reasoning with rule-based logic or domain-specific knowledge to improve robustness, interpretability, and operational trust.

REFERENCES

[1] K. Stouffer, K. Stouffer, M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, S. Lightman, A. Hahn, S. Saravia, A. Sherule *et al.*, *Guide to operational technology (ot) security*. US Department of Commerce, National Institute of Standards and Technology ..., 2023.

[2] S. Kumar and H. Vardhan, "Cyber security of ot networks: A tutorial and overview," *arXiv preprint arXiv:2502.14017*, 2025.

[3] A. Staves, S. Maesschalck, R. Derbyshire, B. Green, and D. Hutchison, "Learning to walk: Towards assessing the maturity of ot security control standards and guidelines," in *2023 IFIP Networking Conference (IFIP Networking)*. IEEE, 2023, pp. 1–6.

[4] A. Johnson, A. Johnson, K. Dempsey, R. Ross, S. Gupta, and D. Bailey, *Guide for security-focused configuration management of information systems*. US Department of Commerce, National Institute of Standards and Technology, 2011.

[5] CSO Online, "Nail the software setup and avoid attacks with the top 10 cybersecurity misconfiguration list." 2023, available: https://www.csoonline.com/article/3623709/nail-the-software-setup-and-avoid-attacks-with-the-top-10-cybersecurity-misconfiguration-list.html.

[6] CISA - Cybersecurity and Infrastructure Security Agency, "Threat actors continue to exploit ot/ics through unsophisticated means." 2023, available: https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-278a.

[7] ISEC7 Group, "Best practices: 10 common cybersecurity misconfigurations and how to mitigate them (part 1)." 2023, available: https://blog.isec7.com/en/best-practices-10-common-cybersecurity-misconfigurations-and-how-to-mitigate-them-part-1.

[8] CISA - Cybersecurity and Infrastructure Security Agency, "Threat actors exploit internet-accessible ot devices in the wws sector." 2024, available: https://www.cisa.gov/news-events/alerts/2024/09/25/threat-actors-continue-exploit-otics-through-unsophisticated-means.

[9] Waterfall Security, "Ot cybersecurity: The top 10 attacks since 2020." 2024, available: https://waterfall-security.com/ot-insights-center/ot-cybersecurity-insights-center/ot-cybersecurity-the-top-10-attacks-since-2020/.

[10] D. Myers, R. Mohawesh, V. I. Chellaboina, A. L. Sathvik, P. Venkatesh, Y.-H. Ho, H. Henshaw, M. Alhawawreh, D. Berdik, and Y. Jararweh, "Foundation and large language models: fundamentals, challenges, opportunities, and social impacts," *Cluster Computing*, vol. 27, no. 1, pp. 1–26, 2024.

[11] Y. Li, H. Zhao, H. Jiang, Y. Pan, Z. Liu, Z. Wu, P. Shu, J. Tian, T. Yang, S. Xu *et al.*, "Large language models for manufacturing," *arXiv preprint arXiv:2410.21418*, 2024.

[12] S. R. Rahmani, "Integrating large language models into cybersecurity incident response: Enhancing threat detection and analysis," *University of Applied Sciences Technikum Wien*, 2024.

[13] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[14] B. Chandra, P. Preethika, S. Challagundla, and Y. Gogireddy, "End-to-end neural embedding pipeline for large-scale pdf document retrieval using distributed faiss and sentence transformer models," *Journal ID*, vol. 1004, p. 1429, 2024.

[15] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, "A systematic survey of prompt engineering in large language models: Techniques and applications," *arXiv preprint arXiv:2402.07927*, 2024.