

LLM-based Cyber Security Testing for Consumer Internet of Things

Yu-Chiang Wang*, Chiung-Fang Hsu*, Yu-Pin Shen*, Yu-Ting Lu*, Jiann-Liang Chen*, Han-Hung Hsu**

*Department of Electrical Engineering, National Taiwan University of Science and Technology Taipei, Taiwan

**Auray Technology Corp. Taoyuan, Taiwan

M11207509@gapps.ntust.edu.tw, M11207524@gapps.ntust.edu.tw, M11207508@gapps.ntust.edu.tw,
M11207516@gapps.ntust.edu.tw, lchen@mail.ntust.edu.tw, hayes.hsu@auray.com.tw

Abstract— With the increasing prevalence of handheld devices in daily life, routers have become essential for connecting devices to the Internet while also becoming prime targets for hacker attacks. Currently, the cybersecurity testing of these devices primarily relies on manual processes. Testing engineers are required to thoroughly understand various regulations, such as EN 303 645 and TS 103 701, and to develop specific testing procedures for different devices. This paper proposes a semi-automated AI system for testing Internet of Things (IoT) devices. The system uses standardized procedures and AI-driven regulatory interpretation to rapidly generate test reports, significantly reducing engineers' time searching for relevant regulations and improving testing efficiency and accuracy. This study evaluates multiple open-source embedding models and the Llama 3.1 70B large language model to develop an Advanced Retrieval-Augmented Generation (Advance-RAG) framework. This framework automatically retrieves test items from cybersecurity regulations and generates preliminary assessments based on self-declarations provided by device manufacturers. It quickly determines whether a device's functionalities comply with specified testing standards. Testing engineers can then upload supporting evidence through the report generation system, providing a comprehensive test report.

Keywords— Large Language Model, Embedding Model, Advance RAG, EN 303 645, TS 103 701

I. INTRODUCTION

With technological advancements, Internet of Things (IoT) devices have become deeply integrated into our lives, from large-scale infrastructure controllers to smart home appliances. According to the latest 2024 SonicWall Cyber Threat Report [1], the number of attacks on IoT devices increased by 107% in the first half of 2024. The TP-Link command injection vulnerability (CVE-2023-1389) was a significant factor driving these attacks, which impacted 21.25% of small and medium-sized businesses (SMBs). This vulnerability facilitated the spread of the Mirai malware, which hijacks IoT devices to create botnets for executing distributed denial-of-service (DDoS) attacks. For instance, the Volt Typhoon botnet attack in the United States exploited compromised SOHO routers to target critical infrastructure. The urgency of addressing these vulnerabilities is clear, and our proposed AI system can play a crucial role in this effort.

Attackers also exploited device vulnerabilities from manufacturers like Cisco and NetGear, often outdated and no longer receive software updates. IoT devices are crucial critical infrastructure components but are frequently targeted due to their relatively weak security. Successful attacks on these devices offer substantial economic rewards for cybercriminals. In addition to the security risks posed by unsupported software updates on IoT devices, the cybersecurity community has advocated for the "Secure by Design" principle in recent years.

In early 2024, the Cybersecurity and Infrastructure Security Agency (CISA) issued a security alert [2] highlighting attackers' frequent use of default passwords and urging manufacturers to turn off default settings to prevent abuse. CISA also called for unique instance passwords, periodic password changes, or requiring physical access during initial setup to enhance security. Manufacturers are responsible for ensuring that their products are securely configured upon release.

These alerts and investigation reports have led many countries to introduce IoT cybersecurity legislation to regulate product safety standards. Given the large number of IoT devices and the varying cybersecurity regulations across different countries, this study proposes a semi-automated AI system for detecting IoT devices. This system aims to improve the efficiency of testing engineers and further ensure the security of network connections.

II. RELATED WORK

This chapter introduces the advancements in Large Language Models (LLMs), vector stores and embeddings, fine-tuning techniques, and Retrieval-Augmented Generation (RAG) related to our LLM-based Information Security Detection System.

A. Large Language Models

Large Language Models (LLMs) are widely used for natural language understanding and reasoning tasks. These models can comprehend complex, domain-specific language and provide detailed analysis. GPT-3 [3] demonstrated unprecedented performance in few-shot learning, making it a versatile tool for

a wide range of NLP tasks with minimal fine-tuning. Llama3 [4] builds on its predecessors by offering enhanced performance and efficiency, particularly in resource-constrained environments, making it suitable for real-time applications. Llama3.1 improves on Llama3 with refined architecture, enabling better contextual understanding and faster processing for large-scale language tasks.

B. Vector stores and embeddings

Reference [5] introduces a technique for improving the quality of dense vector representations in retrieval systems by leveraging targeted supervision. This approach enhances retrieval accuracy, making it especially useful for tasks. Embedding models convert text into dense vector representations, enabling efficient similarity searches and converting the provided documents into vectors. These embeddings are stored in a vector database optimized for fast retrieval. By comparing these vector representations, the system can quickly assess whether the self-claims align with the relevant security standards.

C. Advanced RAG

The Retrieval-Augmented Generation (RAG) technique enhances LLM capabilities by integrating information retrieval systems with generative models. Reference [6] explores RAG as an approach to tackle challenges in Large Language Models, including issues like hallucination and outdated information. In this approach, the model retrieves relevant documents or information from a vector store, which is then used to generate responses or solutions. Recently, Self-RAG [7] was proposed to enable the model to develop a deeper understanding of complex reasoning tasks, improving its ability to handle more sophisticated queries and decisions.

D. Cybersecurity Regulations

The ETSI EN 303 645 standard (Cyber Security for Consumer Internet of Things: Baseline Requirements) [8] focuses on the cybersecurity of consumer IoT products and incorporates GDPR provisions for data protection. The primary aim of this standard is to provide a unified framework and baseline for enhancing the cybersecurity of consumer IoT devices, ensuring that security and privacy protections are effectively managed and implemented. This helps reduce the risk of compromised or misused devices and has become a global trend.

Furthermore, to ensure consistency in the testing and validation of this standard, ETSI released TS 103 701 (Cyber Security for Consumer Internet of Things: Conformance Assessment of Baseline Requirements) [9] in 2021. This document provides guidelines for testing laboratories on conducting testing and assessment against the EN 303 645 standard. The testing scenarios outlined in TS 103 701 include conceptual and functional validation. The former involves verifying the reasonableness of design content based on additional implementation information provided. In contrast, the latter means validating that the product's actual operation

aligns with the design content specified in the implementation information.

III. METHODS

This study introduces a cybersecurity assessment system for IoT devices based on LLM. The system refines the model using an Advanced Retrieval-Augmented Generation (Advance-RAG) framework and Instruction Tuning. This method ensures precise retrieval of cybersecurity regulations and creates thorough testing scenarios. Moreover, the system assesses whether the manufacturer-provided device declarations meet the specified testing requirements, providing initial assessment results and justifications to testing engineers. This approach improves the accuracy of regulatory interpretation and reduces the risk of human error.

A. Large Language Model

This study evaluates open-source and proprietary language models, including Llama 3.1 70B, Llama 3.1 8B, Gemma2 9B, and OpenAI 4o. The experiments were conducted on a hardware setup consisting of four Tesla V100 GPUs, each with 32 GB of memory. Due to hardware limitations, the Llama 3.1 70B model was quantized to 4 bits using the BitsAndBytesConfig utility from the Transformers library.

To assess the models' performance, we evaluated their accuracy and processing speed while optimizing memory usage to ensure efficient execution. The memory required for intermediate variables, particularly in the attention mechanism, was calculated using the following simplified formula(1):

- Total Memory Usage:

$$\text{Model Weights Memory Usage} + \text{Intermediate Variables Memory Usage} \quad (1)$$

B. Advanced RAG

1. Chunking

The cost between vector databases and LLMs (Large Language Model) generation are inversely related. As the chunk size of text decreases and the number of chunks increases, the cost of vector storage and query time rises. Conversely, smaller chunks can reduce token consumption and less noise when input into LLMs for generation. Chunking is a critical component of RAG (Retrieval-Augmented Generation), and an effective chunking strategy can significantly impact retrieval quality. The choice of chunking strategy can affect storage costs and query latency in vector databases and may even help prevent LLM hallucinations. Based on the regulations, we determined that setting the chunk size to 1000 and the chunk overlap to 200 yielded the best results. This configuration was tested against chunk sizes of 800, 900, and 1000, with the overlap set to 200. Due to the varying lengths of sections in the regulations, the 1000-size chunk with a 200 overlap demonstrated the highest accuracy, effectively capturing the content of different regulations.

2. MultiQueryRetriever

Distance-based vector database retrieval embeds queries into a high-dimensional space and identifies similar documents based on their "distance." However, if the user's input query is ambiguous or the embeddings fail to capture the semantic meaning of the data accurately, the retrieval results may vary significantly.

MultiQueryRetriever leverages LLMs to generate multiple queries from different interpretive perspectives based on the user's input query, automatically adjusting the prompts. For each input query, it retrieves the top K relevant documents and performs a unique union across all retrieved documents to obtain a broader set of potentially relevant documents. MultiQueryRetriever can generate multiple perspectives for the input query, interpreting the user's input differently. This approach helps to overcome certain limitations of distance-based retrieval methods and provides a richer set of results.

This study introduces a re-rank function to improve the retrieval results further. Therefore, a more significant parameter of K is set in the MultiQueryRetriever, specifically $K=10$, as shown in Figure 1. This means that for each user input query, two additional queries from different perspectives are generated by LLMs, resulting in three queries in total. Each of these queries retrieves the Top $K=10$ relevant documents, meaning up to 30 documents may be retrieved.

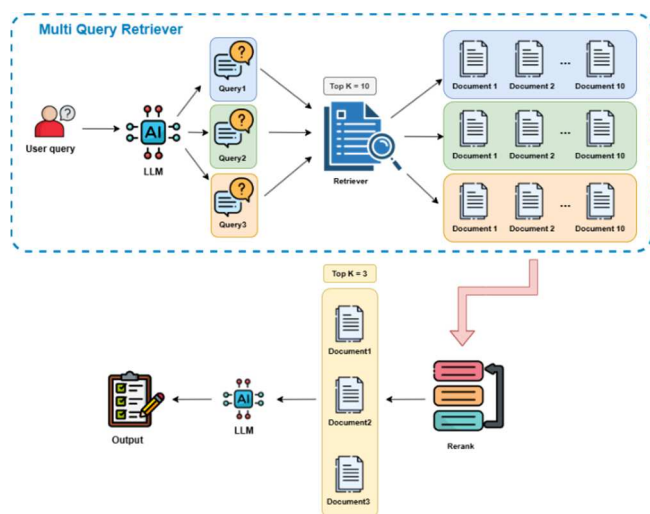


Figure 1. Advanced RAG Workflow

3. Rerank

Traditional retrieval systems rely on vector space models or inner product-based similarity calculations. While these methods can provide reasonable retrieval results in most cases, they may fail to effectively distinguish the relevance of documents when dealing with semantically complex or subtly relevant queries, such as the regulations used in this study. Therefore, this research employs a secondary filtering and ranking process on the initial set of retrieved

documents. The objective is to reorder the documents based on their relevance to the query, thereby enhancing the quality and accuracy of the retrieval results.

C. Instruction Tuning

Research has demonstrated that when large language models (LLMs) are fine-tuned using descriptive natural language across diverse multitask datasets, they enhance performance on new tasks. Instruction tuning helps the model better interpret user intentions and adhere to directives. The initial phase involves fine-tuning the pre-trained model to adapt to specific tasks. The data structure used for this training follows the format {instruction, input (context returned by retriever), output}. The training dataset encompasses four key attributes the system aims to internalize and master: provision details, test scenarios, self-declarations, reasons, and results.

The system develops a nuanced understanding of these elements by learning to discern the relationships between different types of provisions, corresponding test scenarios, and associated self-declarations. This comprehensive understanding is crucial for enabling the system to produce accurate assessment results and generate explainable reasons for these judgments. We anticipate that this targeted learning approach will allow the system to identify subtle relationships within the data, enhancing its evaluative accuracy. The template for this methodology is outlined in Table 1.

TABLE 1. PROMPT TEMPLATE

Prompt Template
Below is an instruction that describes a task paired with an input that a retriever retrieves. Write a response that appropriately completes the request. ### Instruction: Please evaluate the self-declaration provided by the tester based on the following requirements: 1. Verify the three key points of the evaluation basis above, determine whether the self-declaration meets the Assignment of verdict, and provide a reason ('reason'), initial result ('result'), compliance level 0%~100% ('level'), and modification suggestions for the self-declaration ('suggestion'). 2. If the self-declaration is unrelated to the provision, test scenario, or assignment of verdict, 'result' should be "N/A," 'level' should be "N/A," and a modification suggestion for the self-declaration should be provided in 'suggestion'. If the result is "PASS" or "FAIL," 'suggestion' should be "N/A." 3. Make the evaluation lenient, and if there are no obvious violations of key points, the 'result' is "PASS." ... ### Input(context) {self_declaration}, {provision}, {provision_detail}, {test_scenario}, {criteria}

Response:

```
{
  "reason": "The self-declaration describes the contact method and vulnerability analysis process for D-Link website vulnerability reporting but does not mention a timeline for preliminary confirmation of receipt and status updates for resolved report issues.",
  "level": "30%",
  "result": "FAIL",
  "suggestion": "N/A"
}
```

```
</eot_id>
```

To further improve the model's interpretative accuracy, we employed prompt engineering techniques to refine how instructions are presented to the model. The prompt evaluated whether the device manufacturer's self-declaration aligns with specified provisions, test scenarios, and assigned verdicts. The model generated an initial judgment and an accompanying explanation by structuring the prompt to assess three core elements. Additionally, we utilized the chain of thought approach and optimized existing prompts using ChatGPT, applying enhancements through techniques such as few-shot learning. These optimizations effectively addressed issues like incoherent sentences or system hang-ups, which were occasionally observed with Llama 3.1.

D. Web Application Environment

The web system uses the Django framework as the foundation for development. This framework is primarily based on the MVT architecture, which consists of three parts: Model, View, and Template. It allows for the separation of frontend interface design and backend data processing, improving development efficiency and maintainability of the code. Users communicate with the View through HTTP requests via a browser, and based on the user's request, the system returns either data or a webpage. If it's a webpage, the corresponding Template will be rendered, and if it's data, the system communicates with the database using the Model to retrieve the required information in Figure 2.

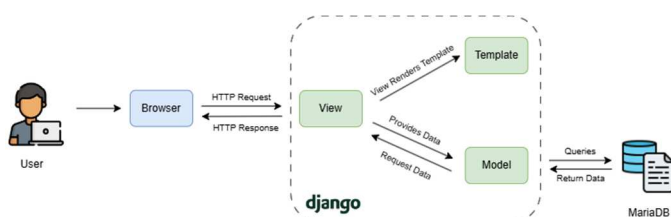


Figure 2. Web Application Dataflow

IV. EVALUATION

This evaluation section outlines the hardware specifications used in the study and calculates the memory usage for various models, including Llama 3.1 70B (4-bit), Llama 3.1 70B (8-bit), and OpenAI GPT-4o. The primary focus is on assessing the accuracy and quantifying the output of these models when integrated into the Advanced RAG framework, which employs techniques such as Chunking MultiQueryRetriever, reranking,

and hybrid search to prove the experiments on the use of the LLM have been successfully fine-tuned and could support the system. Figure 3 describes the whole system flow of how preliminary verdicts are generated.

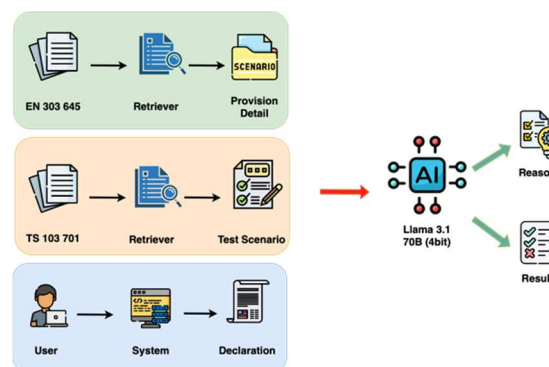


Figure 3. System Flow of verdicts

A. Hardware and Memory Usage

This section details the experimental environment and parameter configurations used in the study. It begins with a description of the hardware specifications. Two distinct hardware configurations were employed: the first setup, dedicated to developing and training the LLM models and implementing the main program, consisted of four Tesla V100-Sxm2 GPUs, each with 32 GB of memory. The second configuration was designed to simulate a typical user or consumer's web application environment, featuring an Intel Core i9 12th Gen processor, 32 GB of RAM, and 1 TB of storage.

To accommodate hardware constraints, we quantized the Llama 3.1 70B model to 4-bit and 8-bit representations for comparative analysis. The 4-bit quantized model required a minimum of 35 GB of memory to load, while the needed 8-bit version was 70 GB. When executing a test report involving 20 provisions, the 8-bit model took an advance of 20.76% to complete compared to its 4-bit counterpart. As a benchmark, we also evaluated the performance of the OpenAI GPT-4o, a widely recognized LLM model. Due to its deployment on cloud-based computing resources, GPT-4o demonstrated a significantly shorter execution time of 8.6 seconds for the same task. The detailed memory usage and runtime for each LLM model are summarized in Figure 4.

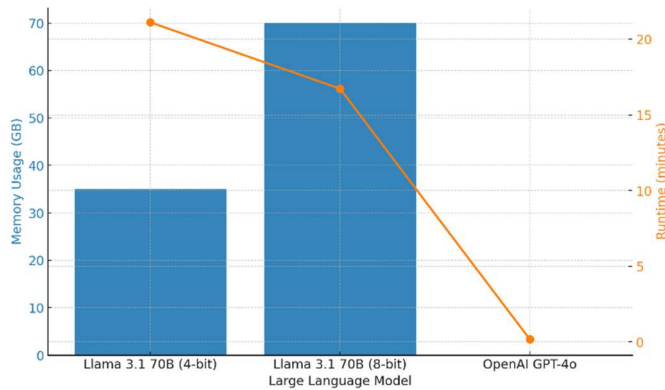


Figure 4. Memory Usage and Runtime of LLMs

B. Accuracy of Retrievers

This comprehensive evaluation meticulously assesses the retrieval accuracy of the open-source embedding model `lier007/xiaobu-embedding-v2` compared to OpenAI's `text-embedding-ada-002` within the Advanced RAG framework. The analysis revealed subtle differences in how each model retrieved content according to specific provisions.

For instance, in the context of the provision regarding the publication of the software update support period (Provision 5.3-13), `lier007/xiaobu-embedding-v2` accurately included the stipulation that the support period should be understandable for users with limited technical knowledge. In contrast, `text-embedding-ada-002` only referred to the need for the support period to be publicly available, overlooking the aspect of user comprehension.

Another disparity was observed in the provision related to verifying the authenticity and integrity of software updates (Provision 5.4-4). While `lier007/xiaobu-embedding-v2` accurately retrieved the requirement for verifying both software updates and critical security parameters, `text-embedding-ada-002` focused on the mechanisms and side effects of the update process, which was not in line with the core requirement.

These examples illustrate the 100% accuracy achieved by `lier007/xiaobu-embedding-v2` compared to the 91.53% accuracy of `text-embedding-ada-002`, underscoring its potential to significantly improve alignment with the specific requirements outlined in the TS103701 standards.

TABLE 2. ACCURACY OF TEST SCENARIO GENERATION

Large Language Model	Embedding Model	Accuracy
Llama 3.1 70B (4-bit)	<code>lier007/xiaobu-embedding-v2</code>	100%
Llama 3.1 70B (8-bit)	<code>lier007/xiaobu-embedding-v2</code>	100%
OpenAI GPT-4o	<code>text-embedding-ada-002</code>	91.53 %

C. Accuracy of Preliminary Verdicts

This section evaluates the differences in the preliminary verdicts generated by three model configurations, as shown in Table 3. Each model's outputs for the "Reason" and "Results" parts across various test items reveal some key differences, summarized and analyzed in Table 4.

TABLE 3. COMPARISON OF PRELIMINARY VERDICTS ACROSS MODELS

Test Item	Llama 3.1-70B 4bit	Llama 3.1-70B 8bit	GPT-4o	Analysis of Differences
5.1-1 Password Uniqueness	PASS	PASS	FAIL	Llama models are compliant with password settings; GPT-4o requires explicit alignment between self-declaration and control measures.
5.1-3 Cryptographic Authentication	PASS	PASS	PASS	GPT-4o adds consistency checks with IXIT documentation, indicating stricter compliance evaluation.
5.3-13 Support Period Transparency	PASS	PASS	FAIL	GPT-4o demands explicitly mentioning the software update support period, whereas Llama models base the verdict on general availability information.
5.3-16 Model Identification	PASS	PASS	FAIL	GPT-4o requires more precise identification of device models; Llama models consider existing labelling sufficient.
5.4-3 Hardcoded Parameters	PASS	FAIL	FAIL	Llama 4bit is more lenient; Llama 8bit and GPT-4o require specific evidence to confirm no hardcoded parameters.
5.4-4 Update Integrity Checks	PASS	PASS	FAIL	GPT-4o emphasizes a direct correlation between self-declaration and test standards; Llama models focus on compliance with technical details.
5.5-1 Cryptographic Best Practices	PASS	PASS	PASS	GPT-4o specifies compliance with FIPS 140-3 standards, showing a higher level of scrutiny.

5.6-1 Disabling Unused Interfaces	PASS	PASS	FAIL	GPT-4o requires detailed confirmation of the disabling status, whereas Llama models grant a more lenient "PASS."
5.8-2 Protection of Sensitive Data	PASS	PASS	FAIL	GPT-4o demands explicit connection between self-declaration and control measure; Llama models rely on the presence of encryption methods.
6.1 Transparen cy of Personal Data	PASS	PASS	FAIL	GPT-4o prioritizes transparency and information completeness; Llama models rely on a more general interpretation of compliance.

The analysis in Table 3 reveals that GPT-4o tends to be more stringent, focusing on the direct relevance and specificity of self-declarations to control measures for each test item. In contrast, the Llama3.1-70B-4bit and 8bit models offer a more relaxed interpretation of compliance standards, occasionally granting a "PASS" based on a general impression of the self-declaration. These differences may stem from the models' varying approaches to handling ambiguous information and interpreting compliance standards. GPT-4o exhibits higher expectations for compliance, requiring more explicit evidence and justification.

V. CONCLUSIONS

In this study, we introduced a semi-automated AI system designed to enhance the cybersecurity testing of IoT devices. The system significantly improves regulatory interpretation and test report generation efficiency by leveraging the Advanced Retrieval-Augmented Generation framework and fine-tuning large language models like Llama 3.1 70B. Our evaluation confirms that integrating LLM-based retrieval and assessment methods effectively reduces the time engineers spend searching for relevant regulations and increases the accuracy of cybersecurity testing. Furthermore, by incorporating LoRA-based optimization and advanced retrieval techniques, the system provides comprehensive preliminary assessments that align with industry standards such as EN 303 645 and TS 103 701. These findings highlight the potential for AI-driven solutions to streamline the IoT cybersecurity testing process, helping manufacturers and testing laboratories to meet the growing demand for secure IoT products in an increasingly connected world.

REFERENCES

- [1] SonicWall Inc., "2024 SonicWall Mid-Year Cyber Threat Report," Milpitas, CA, 2024.
- [2] Cybersecurity and Infrastructure Security Agency, "Secure by Design Alert: How Manufacturers Can Protect Customers by Eliminating Default Passwords," Dec. 2023.
- [3] Brown, Tom B, etal. "Language models are few-shot learners." arXiv preprint arXiv:2005.14165, 2020.
- [4] Dubey, Abhimanyu, et al. "The llama 3 herd of models." arXiv preprint arXiv:2407.21783, 2024.
- [5] Neelakantan, Arvind, et al. "Text and code embeddings by contrastive pre-training." arXiv preprint arXiv:2201.10005, 2022.
- [6] Gao, Yunfan, et al. "Retrieval-augmented generation for large language models: A survey." arXiv preprint arXiv:2312.10997, 2023.
- [7] Asai, Akari, et al. "Self-rag: Learning to retrieve, generate, and critique through self-reflection." arXiv preprint arXiv:2310.11511, 2023.
- [8] ETSI. (2020). EN 303 645 V2.1.1 Cybersecurity for Consumer Internet of Things. [Online]. Available: https://www.etsi.org/deliver/etsi_en/303600_303699/303645/02.01.01_60/en_303645v020101p.pdf, Jun. 2020
- [9] ETSI. (2021). TS 103 701 V1.1.1 Cyber Security for Consumer Internet of Things: Conformance Assessment of Baseline Requirements [Online]. Available: https://www.etsi.org/deliver/etsi_ts/103700_103799/103701/01.01.01_60/ts_103701v010101p.pdf, Aug. 2021/1992.



Yu-Chiang Wang was born in Taiwan in 2000. He received the B.S. degree in 2023. He is currently pursuing an M.S. degree in electrical engineering at the National Taiwan University of Science and Technology, Taipei. His main research interests include artificial intelligence and penetration testing.



Chiung-Fang Hsu was born in Taiwan in 2001. She received her B.S. degree in 2023 and is currently pursuing an M.S. degree in Electrical Engineering at the National Taiwan University of Science and Technology. Her main research interests include applications of artificial intelligence.



Yu-Pin Shen was born in Taiwan in 1999. He received a B.S. degree in Electronic Engineering in 2022. He is currently pursuing an M.S. degree in Electrical Engineering at the National Taiwan University of Science and Technology (NTUST), Taipei. His main research interests include Artificial Intelligence, the Internet of Things (IoT), AI Image Detection, and Cybersecurity.



Yu-Ting Lu was born in Taiwan in 1999. He received the B.S. degree in 2022. She is currently pursuing an M.S. degree in electrical engineering with the National Taiwan University of Science and Technology, Taipei. Her main research interests include artificial intelligence and computer Vision.



Jiann-Liang Chen (Senior Member, IEEE) was born in Taiwan on December 15, 1963. He received a Ph.D. in Electrical Engineering from the National Taiwan University, Taipei, Taiwan, in 1989. Since August 1997, he has been with the Department of Computer Science and Information Engineering of National Dong Hwa University, where he is a professor and Vice Dean of Science and Engineering College. Prof. Chen joins the Department of Electrical Engineering, National Taiwan University of Science and Technology, as a Distinguished Professor and Dean. His research interests are cellular mobility management, cybersecurity, personal communication systems, and the Internet of Things (IoT).



Han-Hung Hsu is a master's student at the Defense University Institute of Technology (CCIT) and an engineer at Auray Technology. He is mainly responsible for research on topics related to AI and information security.