ChatIoT: Large Language Model-based Security Assistant for Internet of Things with RAG

Ye Dong[1,(✉)], Yan Lin Aung[2], Sudipta Chattopadhyay[3], and Jianying Zhou[4]

[1] National University of Singapore, Singapore, Singapore
dongye@nus.edu.sg
[2] University of Derby, Derby, UK
y.aung@derby.ac.uk
[3] University of Missouri–Kansas City, Missouri, USA
sudiptac@ieee.org
[4] Singapore University of Technology and Design, Singapore, Singapore
jianying_zhou@sutd.edu.sg

Abstract. The Internet of Things (IoT) has gained widespread popularity, revolutionizing industries and daily life. However, it has also emerged as a prime target for attacks. Numerous efforts have been made to improve IoT security, and substantial IoT security and threat information, such as datasets and reports, have been developed. However, existing research often falls short in leveraging these insights to assist or guide users in harnessing IoT security practices in a clear and actionable way. In this paper, we propose ChatIoT, a large language model (LLM)-based IoT security assistant designed to disseminate IoT security and threat intelligence. By leveraging the versatile property of retrieval-augmented generation (RAG), ChatIoT successfully integrates the advanced language understanding and reasoning capabilities of LLM with fast-evolving IoT security information. Moreover, we develop an end-to-end data processing toolkit to handle heterogeneous datasets. This toolkit converts datasets of various formats into retrievable documents and optimizes chunking strategies for efficient retrieval. Additionally, we define a set of common-use case specifications to guide the LLM in generating answers aligned with users' specific needs and expertise levels. Finally, we implement a prototype of ChatIoT and conduct extensive experiments with different LLMs, such as LLaMA3, LLaMA3.1, GPT-4o, and DeepSeek-R1. Experimental evaluations demonstrate that ChatIoT can generate more reliable, relevant, and technical in-depth answers for most use cases. When evaluating the answers with LLaMA3:70B, ChatIoT improves the above metrics by over 10% on average, particularly in relevance and technicality, compared to using LLMs alone.

Keywords: Internet of Things · Security · Large Language Model · Retrieval-Augmented Generation

1 Introduction

The Internet of Things (IoT) constitutes a vast network of physical and virtual entities, characterized by their sensing and/or actuation capabilities, programma-

bility features, and unique identifiers. This interconnected infrastructure has rapidly expanded, and IoT connections surpassed non-IoT connections in 2020. It is estimated that over 40 billion IoT devices will be integrated into homes and workplaces via sensors, processors, and software by 2030. With the rapid developments of IoT, increased connectivity and complexity of IoT ecosystems also introduce various vulnerabilities, making them attractive targets for attacks. Consequently, IoT security has become a critical issue.

Over the decades, IoT security has garnered significant attention from researchers, covering both defensive [6,33,55] and offensive [17,25] strategies. As the large language model (LLM) has made significant strides in recent years, it has been explored in the context of IoT security as well, such as threat/vulnerability identification [22,39,54,58], perceive IoT sensor data [41], device management and labeling [40,45], and IoT trust semantics enhancements [24]. This surge in interest has led to many domain-specific datasets that offer extensive insights covering various aspects of IoT security, such as vulnerabilities and exploits [15,30], tactics, techniques, and procedures (TTPs) [49], and industry-standard guidelines [5,56]. However, most existing works focus on discovering new vulnerabilities/threats or designing novel defensive/offensive techniques, but pay less emphasis on leveraging the insights contained in datasets to assist or guide users in enhancing their security practices. Although [15,30,49] offer information search services, they typically only provide raw data (*e.g.*, vulnerability), which is often challenging for non-technical users to understand. Even experienced security analysts may find it difficult to extract actionable insights from massive unprocessed information. Recently, Google announced Sec-Gemini v1, a new experimental AI model focused on advancing cybersecurity AI frontiers, by combining Google's Gemini with near real-time security data and tooling (*e.g.*, fine-tuning). However, it is not focusing on IoT security. As highlighted in Gartner's IoT Security Primer: Challenges & Emerging Practices (2018, updated in 2020): *Information Security's largest IoT challenge is poor visibility and understanding of IoT devices and how the organization uses them.* Consequently, *it is urgent to provide a solution that delivers timely, actionable, and easily accessible IoT security and threat intelligence to a wide range of users, including technical and non-technical ones.*

Inspired by LLM's advanced capability, there are two promising approaches i) *fine-tuning* the LLM specifically on IoT security information and ii) *RAG*, which *R*etrieves IoT security information to *A*ugment the LLM's *G*eneration. However, naïve applying them is still facing one or several of the following challenges:

Challenge-❶: IoT Security Evolves Rapidly. The field of IoT security is continuously developing, with new vulnerabilities, exploits, and security protocols emerging regularly. For example, in August 2024, the VARIoT vulnerability dataset [30] received 79 updates, and cybersecurity websites almost report new developments daily[5]. Fine-tuning an LLM on static IoT data would quickly become outdated, failing to provide the latest threats or innovations in IoT security. Moreover, while fine-tuning may yield comparable or slightly better results, it requires substantial resources, i.e., GPUs or paid APIs, incurring significant com-

[5] https://www.bleepingcomputer.com/, https://www.darkreading.com/

putational expenditure and financial overhead. For instance, OpenAI fine-tuning API charges $25 for training GPT-4o on 1M tokens, compared to $2.50 for solely processing inputs[6]. This creates an accessibility barrier for resource-constrained researchers and startups aiming to contribute to IoT security advancements. With frequent IoT updates, direct fine-tuning becomes a bottleneck and expensive.

Challenge-❷: Heterogeneous Dataset Formats. IoT security datasets come in a variety of formats, such as structured data (*e.g.*, vulnerability databases [30]), unstructured text reports, and product compliance lists (such as Cybersecurity Labelling Schemes in Singapore, Finland, and Germany). Fine-tuning/augmenting an LLM on such diverse data types without a robust data processing mechanism would significantly limit the utilization of information contained in these datasets.

Challenge-❸: Diverse User Requirements. Users in IoT ecosystems range from consumers to security analysts, each with distinct expertise levels and specific security concerns. Directly using LLMs (even fine-tuned/augmented ones) without considering varied requirements will limit their ability to provide meaningful and contextually appropriate responses for different user groups.

In light of the knowledge gap and technical challenges as above, we ask:

Can we use large language models to effectively disseminate IoT security assistance to various key users of the IoT ecosystems in an understandable and actionable manner to provide better IoT security guarantees?

We present ChatIoT, an LLM-based IoT security assistant augmented with external information retrieval, to answer this question affirmatively. On the paradigm of RAG, we design ChatIoT by combining the advanced language understanding and reasoning capabilities of LLM with fast-evolving IoT security-specific information to generate reliable, relevant, and technical answers. To handle heterogeneous datasets, we have developed an end-to-end data processing toolkit that integrates a range of existing technologies, enabling the conversion of diverse data formats into retrievable documents and optimizing chunking strategies to improve retrieval performance. Besides, we define several use-case specifications and provide users' *background* to enable role-query aware retriever selection and LLM generation. In summary, ChatIoT offers the following contributions:

- **LLM-based Adaptive Retrieval & Generation.** To accommodate the needs of diverse users, we propose user role-aware self-querying retrieval and LLM generation mechanisms. Consequently, our ChatIoT is able to i) dynamically activate different retrievers based on the user's background and query context, and ii) align generated outputs with the user's expertise level.
- **Data processing toolkit.** We design toolkit **DataKit** to handle datasets in diverse formats. Our toolkit first parses the raw data and converts the parsed contents into text. Notably, we utilize LLM to i) select the appropriate content for page_content and metadata of documents, and ii) process the multimodal contents, such as figures, tables, and codes, into text. We integrate Ragas [20] to optimize chunking strategy, including splitter methods and chunk sizes. We integrate these technologies from existing works as an end-to-end data processing toolkit, which might be of independent interest.

[6] https://openai.com/api/pricing/

– **Implementation & Evaluation.** We implement a prototype of ChatIoT and define five use cases. For each case, we specify the user's background in terms of *knowledge*, *goals*, and *requirements* to guide the LLM in generating answers that are not only reliable, relevant, and technical but also user-friendly. Extensive evaluations show the improvements achieved by ChatIoT. Specifically, we compare ChatIoT's answers with those generated directly from LLaMA3, LLaMA3.1, and GPT-4o in terms of reliability, relevance, technical depth, and user-friendliness. When evaluated with LLaMA3:70B, ChatIoT improves all metrics by over 10% on average, particularly in relevance and technicality. Human evaluations also confirm ChatIoT provides better answers.

Organization. We introduce the background in § 2. An overview of design goals and system architecture is given in § 3. Our concrete design is illustrated in § 4. We implement the prototype and perform experimental evaluations in § 5. Finally, we summarize the related works in § 6 and conclude this work in § 7.

2 Background

2.1 IoT Threat Intelligence

IoT threat intelligence refers to the collected and analyzed data related to threats, vulnerabilities, and attack patterns. The threat intelligence helps to monitor and analyze threats specific to IoT ecosystems, including vulnerabilities [30], communication protocols [19], malware [9], tactics, techniques, and procedures (TTPs) [49], and [12,27,53]. Recent research has highlighted the importance of IoT threat intelligence in identifying zero-day vulnerabilities [47] and mitigating threats using collaborative defense mechanisms [7]. The dynamic and fast-evolving nature of IoT systems, including both consumer and industrial settings, highlights the need for up-to-date and automated threat intelligence solutions.

2.2 Large Language Models

Large Language Models (LLMs) are pre-trained on billions of available datasets. Due to extensive training data, LLMs can be directly employed for many downstream tasks [10,32,44,59]. However, LLMs have certain limitations, particularly when handling highly specialized or constantly evolving domains. Since the training data does not always reflect the latest information, it may generate incomplete or outdated responses for specific queries. RAG [21,34] is a powerful approach that combines the generative capabilities of LLMs with the latest external information retrieval. It augments LLMs by retrieving relevant, up-to-date information from external databases or documents when generating. Therefore, RAG enhances the performance of LLMs on tasks that require both advanced language understanding, reasoning, and up-to-date information.

3 Overview

We introduce our design goals (§ 3.1) and then present system architecture (§ 3.2).

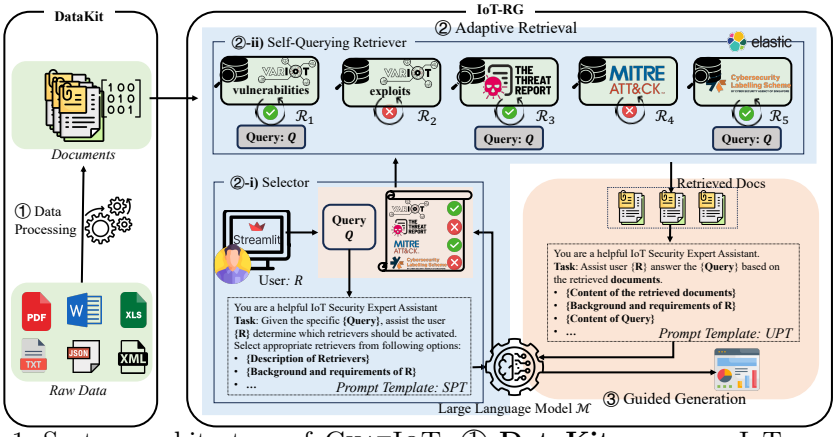

Fig. 1: System architecture of ChatIoT: ① **DataKit** processes IoT security datasets from various formats to documents for building retrievers. ② **IoT-RG** provides an interface for submitting the query and retrieves relevant documents adaptively: i) Select activated retrievers, and ii) Activated self-querying $\mathcal{R}_i$s retrieve similar documents and filter out unsatisfied ones based on metadata. ③ LLM will synthesize all information to generate answer.

3.1 Design Goals

ChatIoT is equipped with the following features:

- **G-①: Coping with the rapid evolution of IoT threat intelligence.** ChatIoT is designed to stay up-to-date with the rapidly evolving IoT threat intelligence. By combining the advanced capabilities of LLMs with external information retrieval, it is enhanced by the latest IoT threat information, such as new vulnerabilities, to deliver timely and technical insights.
- **G-②: Filtering relevant information.** To prevent information overload, ChatIoT uses advanced retrieval and filtering techniques to prioritize important, highly relevant information while discarding irrelevant data. This ensures that ChatIoT generates actionable intelligence while avoiding overlooking critical information and filtering out unnecessary data.
- **G-③: Tailored for different user types.** ChatIoT guides its generated answers based on the users' roles and their corresponding backgrounds. This allows user to get insights or solutions that are relevant, understandable, and actionable based on specific requirements and expertise level.

3.2 System Architecture

As illustrated in Figure 1, we designed ChatIoT with two modules: the data processing toolkit, named **DataKit**, and our LLM-based adaptive retrieval and generation system, abbreviated as **IoT-RG**, to achieve our goals:

- **Data Processing.** First, we integrate a variety of technologies to build our end-to-end data processing toolkit **DataKit**. It can convert the collected

Table 1: Selector configurations $\{\mathcal{S}_i\}_{i=1}^{n}$ generated by LLaMA3:8B for user roles and example queries. ✔ indicates `True` and ✘ is for `False`.

Role	Example Query	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{S}_4$	$\mathcal{S}_5$
Consumer	Is it secure to use Signify Smart Lighting in home?	✔	✘	✔	✘	✔
Security Analyst	Conduct a security assessment for TP-Link AX6000 Wi-Fi Router.	✔	✔	✔	✘	✘
Technical Officer	Check TTPs and security labeling of company's TP-Link WiFi Routers.	✔	✔	✘	✔	✔
Developer	Develop a security roadmap for the next generation of Wi-Fi routers.	✔	✘	✔	✘	✔
Trainer	Explain the importance of cybersecurity labeling for smart locks.	✔	✘	✘	✘	✔

IoT threat datasets of various formats into *documents*, which are suitable for retrieval and LLMs processing. For each dataset, it extracts specific content for page_content and metadata for documents, and a corresponding retriever is subsequently constructed. **DataKit** ensures ChatIoT can keep up with the rapid evolution of IoT threat intelligence (G-①).

– **Adaptive Retrieval.** Second, we achieve adaptive retrieval through *Selector* and *Self-Querying* mechanisms. With query Q, the Selector will invoke the equipped LLMs to generate a configuration that determines which retrievers to activate. Once the configuration and query are passed to the retrievers, **IoT-RG** executes the activated self-querying retrievers to get highly relevant documents while filtering out irrelevant ones. This approach prevents overload, ensuring that only relevant and important documents will be utilized (G-②).
– **Guided Generation.** Finally, **IoT-RG** synthesizes the user's background, Q, and retrieved documents to instruct the LLMs to generate answers. In this way, the answers are generated not only based on the advanced language understanding capabilities of LLMs and the retrieved IoT security-related information, but also aligned with the user's background (G-③).

4 Design of ChatIoT

We first construct our **IoT-RG** in § 4.1, followed by the data processing toolkit in § 4.2. Finally, we present the use case specifications in § 4.3.

4.1 Construction of IoT-RG

IoT-RG consists of *Adaptive Retrieval* and *Guided Generation* as Figure 1.

Adaptive Retrieval Recall that we have multiple retrievers, each dedicated to retrieving documents generated from a specific dataset. We achieve an adaptive retrieval mechanism that works in two aspects: i) selecting which retrievers should be activated and ii) trying to retrieve only relevant documents while discarding irrelevant ones, even for the activated retrievers.

Design of Selector. When user role submits a query Q, a straightforward and *static* approach is to Q to all retrievers and gather retrieved documents from them. However, this method has the following drawbacks: i) *Irrelevant Retrievers.*

```
vulns_metadata_field_info = [
  AttributeInfo(
    name = "products",
    description = "Products affected by vulnerability.",
    type = "string or list[string]",
    ),
  AttributeInfo(
    name = "id",
    description = "Entry ID in VARIoT vulnerability.",
    type = "string",
  )
]
vulns_examples = [
  (
    "Is it secure to use TP-Link router?",
    {
      "query": "vulnerability about TP-Link router",
      "filter": "contain(\"products\", \"TP-Link router\")",
      "limit": 5
    },
  ),
  (
    "Provide the vulnerability with ID VAR-201810-0092.",
    {
      "query": "vulnerability with ID VAR-201810-0092",
      "filter": "eq(\"id\", \"VAR-201810-0092\")",
      "limit": 1
    },
  ),
]
```

Fig. 3: Metadata fields information and examples for self-querying retriever corresponding to VARIoT vulnerabilities dataset.

Documents of some retrievers may not be relevant to the query. Retrieving them not only fails to improve the quality of the generated answer but may even negatively affect it. ii) *Resource Costs.* Retrieving unnecessary retrievers leads to increased resource costs, such as computational overhead, during both the retrieval and generation processes.

To address these issues, we propose an LLM-based role-aware adaptive *Selector*. As shown in Figure 1, we put the user Role with the background, the query Q, and the descriptions of all retrievers as *Selector Prompt (SPT)* and input SPT to an LLM $\mathcal{M}$, which generates $\{\mathcal{S}_i\}_{i=1}^{n}$, where $\mathcal{S}_i$ = True implies that the retriever $\mathcal{R}_i$ should be activated, and False indicates not. In Table 1, we present the configurations generated by LLaMA3:8B for some example queries.

Self-Querying Retrievers. The activated retrievers may return information that does not meet requirements, *e.g.*, mismatch *id* and *products*, so we leverage self-querying technique to filter documents by metadata.

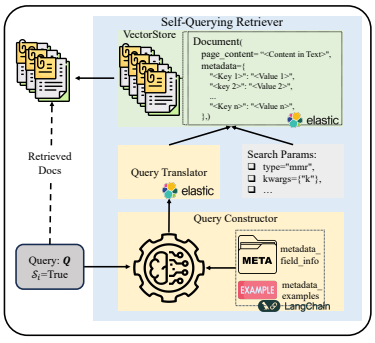

Fig. 2: Self-querying retriever. $\mathcal{S}_i$ = True: retrieving documents semantically similar to Q and filtered by metadata.

```
# Submitted user role and query
User Role = 'Security Analyst';
Query = 'What are security issues with DLink DCS-942 camera?';

# Generated query and filter for VARIoT vulnerabilities
INFO:langchain.retrievers.self_query.base: Generated Query:
  query = 'security issues with DLink DCS-942 camera'
  filter = Comparison(
    comparator=<Comparator.CONTAIN: 'contain'>, attribute='products', value='DLink DCS-942
    ↪  camera'
)
# Generated query and filter for CLS
INFO:langchain.retrievers.self_query.base: Generated Query:
  query = 'security issues with DLink DCS-942 camera'
  filter = Operation(operator=<Operator.AND: 'and'>,
    arguments=[
    Comparison(comparator=<Comparator.CONTAIN: 'contain'>, attribute='manufacturer',
    ↪  value='DLink'),
    Comparison(comparator=<Comparator.CONTAIN: 'contain'>, attribute='Model',
    ↪  value='DCS-942')
  ]
)
```

Fig. 4: Generated queries and filters for VARIoT vulnerabilities and CLS for "What are the security issues with DLink DCS-942 camera?" by Security Analyst.

As Figure 2, self-querying retriever is composed of a Query Constructor, Query Translator, Search Params, and VectorStore. When query Q is passed to Query Constructor, LLM will generate *internal query language elements* based on predefined *metadata_field information* and *metadata_examples*. Query Translator converts these elements into a *structured query* with appropriate filters. Finally, the structured query and search parameters are applied to Vector Store to retrieve documents. To specialize the self-querying retriever for IoT security, we conducts:

① **Metadata & Examples.** We provide the metadata field and examples from relevant datasets. For instance, in the VARIoT vulnerabilities, fields *id* and *products* are utilized as metadata. The corresponding *metadata_field_info* and *examples* are illustrated in Figure 3 (c.f. Appendix C for others). This ensures the LLM can gain the necessary IoT security-specific knowledge to generate effective internal query language elements from the user's query.

② **Create Structured Queries.** Based on the above customized internal query language elements, Query Translator can create specific structured queries for each retriever. Figure 4 shows how to enable the retrieval of VARIoT vulnerabilities and CLS lists that are both semantically similar to the query and appropriately filtered by their respective metadata.

Guided Generation After retrieval, one direct step is feeding the retrieved documents and query to LLM to generate the answer. However, this simple approach is likely to result in user-unfriendly outputs. For example, consumers often lack the expertise needed to fully comprehend highly technical content.

To address this issue, we incorporate user-specific backgrounds, including knowledge, goals, and requirements, into each user type's user-friendly prompt template *UPT*. This adjustment guides LLM in generating role-aware answers

Algorithm 1 IoT-RG

Require: User role, query Q, large language model $\mathcal{M}$, and retrievers $\{\mathcal{R}_i\}_{i=1}^n$.
Ensure: Generated answer A.

▷ **Procedure of Selector:**
Set prompt $\mathsf{SPT} = (\mathsf{Task}, \mathsf{role}, Q, \{\mathcal{R}_i\}_{i=1}^n)$, where role includes user's background implicitly and $\mathcal{R}_i$ indicates its description here.
Inputting SPT to $\mathcal{M}$ and get $\{S_i\}_{i=1}^n = \mathcal{M}(\mathsf{SPT})$, where $S_i \in \{\mathsf{True}, \mathsf{False}\}$.
▷ **Procedure of Self-Querying Retrieval:**
for all $i = 1, \ldots, n$ **do**
  if $\mathcal{S}_i = \mathsf{True}$ **then**
    Execute self-querying retriever $\mathcal{R}_i$ and get documents $D_i = \mathcal{R}_i(Q)$
  else
    Set $D_i = \mathsf{NULL}$.
  end if
end for
▷ **Procedure of Guided Generation:**
Set prompt $\mathsf{UPT} = (\mathsf{Task}, \mathsf{role}, Q, \{\mathcal{D}_i\}_{i=1}^n)$
return $A = \mathcal{M}(\mathsf{UPT})$.

"Knowledge": "The consumer may not have formal technical training but are familiar with using IoT devices for daily convenience such as smart home systems. The consumer has a basic understanding of device operation but may not be aware of the intricate security risks that exist."

"Goals": "The primary aim is to understand whether a device is secure and how to maintain or improve its security, ensure safety, security, and reliability of IoT devices within their homes or personal environments."

"Requirements": "The answers should be practical, easy to follow, and focused on actionable steps the general user can take."

Fig. 5: The background for consumer utilized to guide the ChatIoT.

tailored to the user's needs. The specific background for the general consumer is shown in Figure 5, demonstrating how we simplify content for easy understanding. The background specifications of other user types can be referred to Appendix B. Formally, we summarize and show the workflow of **IoT-RG** in Algorithm 1.

4.2 Data Processing Toolkit

The raw data of the collected datasets are of different formats, *e.g.*, PDF and JSON. These formats are not suitable for RAG processing directly, and thus we develop our end-to-end data processing **DataKit**as Figure 6:

① **Parse Raw Data.** The initial step involves parsing raw data into distinct content elements such as text, tables, figures, and code. Leveraging existing tools like the unstructured library[7] helps in extracting textual content from threat reports, while JSON library[8] is useful for handling structured data from sources like VARIoT and MITRE ATT&CK.

[7] https://pypi.org/project/unstructured/
[8] https://python.readthedocs.io/en/v2.7.2/library/json.html

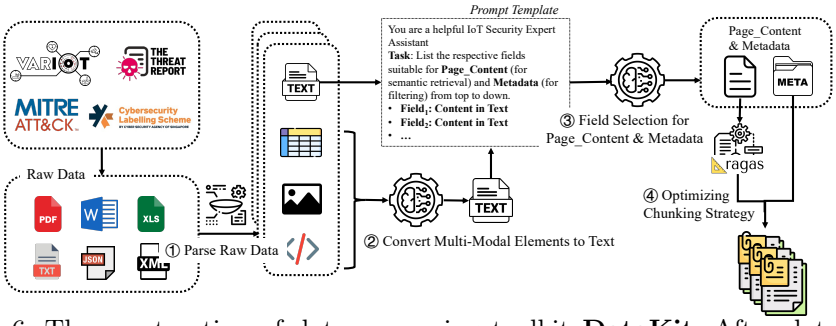

Fig. 6: The construction of data processing toolkit **DataKit**. After datasets collection, **DataKit** first parses the raw data to get elements of multi-modal (step ①), and then converts the multi-modal elements into text by utilizing LLM (step ②). Finally, **DataKit** uses LLM to select fields for the page_content and metadata of documents (step ③), and optimizes the chunking strategy (step ④).

② **Convert Multi-Modal Elements to Text.** Once parsed, any multi-modal elements (*e.g.*, tables, figures, code) must be converted into text descriptions for further processing. LLMs like LLaVA (for images) [35–37], LLaMA3:8B (for tables), and CodeLlama (for code) [46] are employed to generate these descriptions. This modular approach allows for easy integration of other LLMs to handle different types of multi-modal content.

③ **Field Selection for Page_Content & Metadata.** In structured formats like JSON, content is often stored across multiple fields. Instead of using all fields, it is crucial to identify and utilize the most relevant ones for retrievers. This is done by sampling example items from each field and using an LLM (*e.g.*, LLaMA3:8B) to intelligently select fields that best represent the document's page_content and metadata, and the prompt is shown in Figure 6. The selected fields for page_content and metadata of each dataset are shown in § 5.2.

④ **Optimize Chunking Strategy.** The final step involves selecting an appropriate chunking strategy, including the chunking size, overlap, and splitting method. The Ragas library [20] is used to optimize this process, and the details are shown in algorithm 2.

Using the optimized chunking strategy, we split the documents into small chunks and use the all-MiniLM model for chunked text embedding. The documents are composed of chunked text, embedding, and metadata. This approach ensures that the IoT security and threat datasets are processed efficiently and ready for further analysis or use in LLM.

Remark 1. These tools are well-established and widely-used in their respective domains[9], we foucs on putting them all together to develop an end-to-end data processing toolkit, which might be of independent interest and useful in practical applications. Also, as these tools are combined modularly, any future advancements can be integrated easily. Manual downloading is a minor

[9] Their detailed performance evaluations can be accessed on the Internet easily.

Algorithm 2 Optimize Chunking Strategy

Require: Documents D, chunking $sizes$, $overlaps$, and $splitters$ functions, $metrics = [precision, recall]$, and LLM $\mathcal{M}$.
Ensure: Optimized $(size^*, overlap^*, splitter^*)$.

for all $splitter \in splitters$ **do**
  for all $size \in chunk\_sizes$ **do**
    for all $overlap \in overlaps$ **do**
      if $overlap < size$ **then**
        ▷ **Split** D **as small chunks:**
        $\{d_i\}_i = splitter(D, size, overlap)$.
        $(p, r) =$ Ragas.evaluate$(\{d_i\}_i, \mathcal{M}, metrics)$.
      else
        `break.` ▷ **Go to next chunk size.**
      end if
    end for
  end for
end for
return chunking $(size^*, overlap^*, splitter^*)$ with a optimized trade-off between (p, r).

```
"User Role": {
     "description": "Role of the user, such as Consumer.",
     "type": "enum string"
}
"Background": {
     "description": "Specifies the knowledge, goals, and requirements of user.",
     "type": "enum string"
}
"Actions": {
     "description": "Tasks the user can perform.",
     "type": "list of strings"
}
"Example Query": {
     "description": "Provides a sample question/query.",
     "type": "string"
}
```

Fig. 7: Use case specializations. We define role, goal, actions, and example query.

implementation issue that can be easily resolved, i.e., using API. Moreover, while some IoT datasets are publicly available, they require API keys to access and responses are often paginated (*e.g.*, 10 records per call [30]). So, this problem can be accepted in a prototype of concept (PoC) and collaboration with data providers is essential for updating ChatIoT from a PoC to a practical application.

4.3 Use Cases Specialization

In this section, we define five specialized use cases of ChatIoT. Each use case is defined by *four* fundamental properties: *User Role*, *Background*, *Actions*, and

Table 2: Actions and example query for each kind of user role in ChatIoT.

Role	Actions	Example Query
Consumer	i) Assess the security of IoT devices before purchase or installation; ii) Monitor ongoing security status and updates for existing devices; iii) Make informed decisions based on security reports provided by ChatIoT.	Is it secure to use Signify Smart Lighting in home?
Security Analyst	i) Identify and evaluate security threats and vulnerabilities in IoT devices; ii) Recommend mitigation strategies based on threat intelligence and analysis; iii) Provide detailed security reports to stakeholders.	Conduct a security assessment for TP-Link AX6000 Wi-Fi 6 Router.
Technical Officer	i) Ensure that IoT devices are deployed securely and operate within compliance guidelines; ii) Oversee the application of security updates and patches; iii) Monitor the security posture of the IoT ecosystem within their organization.	Check the security labeling of the company's WiFi Routers, including TP-Link, D-Link, and ASUS in Singapore.
Developer	i) Design and develop secure IoT products by adhering to best practices and security standards; ii) Continuously update products to address new vulnerabilities and threats; iii) Provide accurate security documentation and updates to customers.	Develop a security enhancement roadmap for the next generation of TP-Link Wi-Fi routers.
Trainer	i) Develop and deliver training programs on IoT security; ii) Guide users and organizations on how to secure IoT devices and respond to incidents; iii) Provide up-to-date information on IoT security trends and best practices.	Explain the importance of cybersecurity labeling for smart locks like the August Smart Lock.

Example Query, within the IoT security domain. The detailed specifications are illustrated in Figure 7. The roles include *Consumer*, *Security Analyst*, *Technical Officer*, *Developer*, and *Trainer*.

Recall that we have discussed background in § 4.1. Table 2 highlights the key actions associated with each user role, such as assessing the security of IoT devices, deploying security patches, or developing training programs on IoT security. Additionally, it provides example queries for each role, demonstrating how ChatIoT can be utilized to address the unique needs of various users. This structured approach ensures that ChatIoT caters to a diverse range of users, offering tailored assistance and enhancing IoT security management across different scenarios. Note that while we provide five use cases, they are not rigid or fixed. The use cases can be easily extended by defining new user roles, specifying the background (including knowledge, goals, and requirements), and outlining actions. Example queries can also be added to further clarify the context and functionality.

5 Experimental Evaluation

We implement ChatIoT and study the effectiveness of ChatIoT, and answer the following questions:

- **Q1:** How does **DataKit** extract appropriate field selection from IoT threat datasets and convert them into well-structured documents for retrieval and LLM analysis? What are the optimal chunking strategies? (§ 5.2)
- **Q2:** What are the advantages of our system? Can ChatIoT effectively generalize and improve the capabilities of the most advanced LLMs available for IoT security? (§ 5.3)

– **Q3:** Can ChatIoT be an useful IoT assistant over LLM alone? How about the human evaluation? (§ 5.4)

5.1 Setup

Testbed. We implement ChatIoT in Python 3.10.13, utilizing large language models LLaMA3:8B & 70B and LLaMA3.1:8B & 70B provided by Groq[10], GPT-4o-mini and 4o provided by OpenAI[11]. All these LLMs are utilized by calling their APIs. For building the vector store, we employed Elasticsearch 8.13.2 [1], running on Docker Desktop 4.29.0 [2]. All components were integrated using the LangChain library (version 0.2.5) [3]. The WebApp was developed using Streamlit (version 1.33.0) [4]. Experiments were conducted on a MacBook Pro equipped with an Apple M3 Pro CPU (11 cores) and 18 GB of RAM, running macOS 14.6.1 with the Darwin 23.6.0 kernel.

Data Sources. We collect five kinds of IoT security and threat datasets from the public Internet:

– VARIoT vulnerabilities [30]: This dataset catalogs known vulnerabilities in various IoT devices, offering detailed information about the potential risks associated with each vulnerability.
– VARIoT exploits [30]: This dataset contains exploits targeting IoT devices, providing insights into the techniques and methods attackers use to compromise these systems.
– MITRE ATT&CK ICS TTPs [49]: This dataset outlines the tactics, techniques, and procedures (TTPs) employed by adversaries specifically in industrial control systems (ICS), which often include IoT-related TTPs as well.
– Threat reports: We collect 17 public threat reports from VXUG[12] about emerging threats and vulnerabilities, offering analysis and recommendations for mitigating risks.
– Cybersecurity labelling schemes[13] [14] [15] [16]: These schemes provide information on the security posture of various IoT products, helping consumers and organizations assess the security standards and certifications of specific devices.

Remark 2. The datasets are from authoritative and widely recognized cybersecurity repositories, such as VarIoT Vulnerabilities and Exploits, MITRE ATT&CK, and Singapore's Cybersecurity labelling schemes, which maintain well-established curation and verification workflows. These include expert-reviewed vulnerability submissions, standardized CVSS scoring procedures, cross-referencing among multiple reporting channels, and continuous update processes to correct errors. Moreover, we perform additional data-cleaning steps, including deduplication and removal of incomplete records, to ensure reliable and high-quality IoT datasets.

[10] https://chat.groq.com/
[11] https://platform.openai.com/docs/models/gpt-4o
[12] https://vx-underground.org/
[13] https://www.csa.gov.sg/our-programmes/certification-and-labelling-schemes
[14] https://tietoturvamerkki.fi/en/products
[15] https://www.nemko.com
[16] https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.02042022-2.pdf

Table 3: LLMs-based selected fields for page_content and metadata.

Dataset	Page_Content	Metadata
VARIoT Vulns.	title, description	id, products
VARIoT Exps.	title, description, exploit	id, products
ICS	name, description	stixId
Threat Report	Report's content	title
CLS	NULL	All Fields

5.2 Fields Selection & Chunking Strategy

This section shows the experimental evaluation for field selection and chunking strategy optimization.

Fields for Page_Content & Metadata Recall that for each dataset, we should first determine the fields for page_content and metadata for documents before building self-querying retrievers (see § 4.2). For each dataset, we sample 3 items, list the fields' names, and instruct the LLM to select the suitable fields. The results are as follows:

- We use three LLMs: LLaMA3:8B, LLaMA3.1:70B, and GPT4-o, to select fields for VARIoT vulnerabilities, exploits, and MITRE ATT&CK ICS. The experimental results are summarized in Appendix C. While there are slight variations in their selections, there is consensus on the crucial decisions: For instance, in the case of the VARIoT vulnerabilities, all LLMs select `title` and `description` for page_content, and `id` and `products` for metadata. Similar selections are observed for the VARIoT exploits and MITRE ATT&CK ICS.
- For threat reports, which are typically unstructured, we use the report's content as page_content and the title as metadata (Note we do not use self-querying retrieval for threat reports). The CLS schemes consist solely of metadata with no descriptive content, so we leave page_content blank and utilize the metadata for self-querying retrievers. Table 3 shows the selected fields.

Chunking Evaluation To optimize the chunking strategy for documents' page_content, we utilize Ragas library [20] in conjunction with all-MiniLM[17] (for embedding) and LLaMA3:8B (for evaluation) to search most suitable chunking size, overlap, and splitting method for each dataset. We use content *precision* and *recall* as key metrics:

- Precision measures whether all relevant items retrieved by the model are ranked higher than the irrelevant items;

[17] https://ollama.com/library/all-minilm

Table 4: Optimized chunking strategy for VARIoT vulnerabilities, exploits, ICS, and threat reports. `RecurChar` is short for `RecursiveCharacter`.

Dataset	Size	Overlap	Splitter Method
`VARIoT Vulns.`	500	100	`RecurChar`
`VARIoT Exps.`	1000	150	`TokenText`
`ICS`	1000	200	`Character`
`Threat Reports`	500	200	`TokenText`

– Recall measures how much of relevant content is retrieved based on annotated answers and the retrieved context.

Both precision and recall are evaluated within the range $[0, 1]$, where a higher score indicates better performance. As the datasets contain a huge number of samples, for practical efficiency, we select a subset of $1,000$ samples from each dataset except threat reports[18], generate a testset of 50 items, and conduct evaluations based on the subset and testset. This might not result in the optimal chunking size, overlap, and splitter method, but is enough to get a reasonable and useful chunking strategy for our practical applications. As shown in Table 6, we test the following commonly used configurations: 1) chunk sizes: $\{500, 1000, 1500, 2000\}$; 2) overlaps: $\{50, 100, 150, 200\}$; 3) splitters: `Character`, `RecursiveCharacter`, and `TokenText`. Our objective is to achieve high precision and recall simultaneously, ensuring that the system retrieves as many relevant documents as possible while minimizing irrelevant content. From the experimental results in Table 6 (c.f. Appendix A), it is easy to see that using the `RecursiveCharacter` splitter with a chunk size of 500 and an overlap of 100 is the most effective strategy for the VARIoT vulnerabilities, offering the best trade-off between precision and recall. Similarly, we can choose the suitable chunking strategies for VARIoT exploits, MITRE ATT&CK ICS, and threat reports. The details are illustrated in Table 4.

5.3 LLMs-based Evaluation of Outputs

As there is no public Question-Answer dataset about IoT security and threat intelligence, we synthesize 50 common IoT security-related questions (10 questions for each kind of user). The synthetic role-based questions are constructed from classic IoT threat scenarios reports (e.g., VXUG), device-specific vulnerabilities and attacks from authoritative repositories such as MITRE ATT&CK, and common IoT requirements for each user role derived from our survey, ensuring that the questions are realistic, representative, and role-appropriate.

To evaluate our improvements, we compare ChatIoT's outputs with the answers generated by the underlying LLM alone (denoted as LLM-A), which is not equipped with our IoT data sources. For automatic evaluation, we employ another LLM as the evaluator and measure the output quality using four key metrics: *Reliability*, *Relevance*, *Technical*, and *Friendliness*:

[18] We use all collected threat reports for generating testset and evaluation.

`Task`: You are an expert IoT security assistant. Your task is to evaluate the answers to a question posed by a user with {role};

`Background`: The background of {role} is {background};

`Question`: {question};

`Answers`:

1. ChatIoT_answer: {ChatIoT_answer};
2. LLM-A_answer: {LLM-A_answer};

`Instructions`:

- **Criteria**: {The descriptions about Reliability, Relevance, Technical, and User-friendliness.}
- **Score**: i) Provide a score for each answer across the five metrics above. Scores should range from 0 to 5, with 5 being the highest and 0 being the lowest. ii) Scores should reflect how well each answer meets the criteria, particularly in alignment with the user role's background and needs.
- **Output Format**: i) Present a table that includes the names of all answers and their scores for each metric. You can score differently for different metrics. ii) Give explanations for scores.

Fig. 8: The prompt template for LLM-based evaluation of outputs. We provide the role and background of the user, instructions about the criteria and score for mitigating bias in LLM-based evaluation.

- Reliability: The trustworthiness and reliability of each answer, ensuring it is plausible and aligns with known IoT best practices and standards.
- Relevance: Assess how well the answer addresses the specific question and meets the user's needs, considering their role and context in the IoT ecosystem.
- Technical: Judge the appropriateness and precision of technical language, including IoT research, standards, protocols, and relevant technical aspects. Ensure the answer demonstrates a solid understanding of IoT technologies.
- Friendliness: Measure how easy the answer is to comprehend, focusing on clarity and whether the answer provides actionable steps/solutions or not.

All scores are in $[0, 5]$, where 5 is the highest. Among these metrics, *Relevance* aligns with standard RAG evaluation criteria. We introduce *Reliability* and *Technical* as domain-specific extensions tailored to IoT security, and *Friendliness* as an additional dimension for assessing user role–aware generation. For each question, the answers generated by ChatIoT and the corresponding LLM-A are inputted into the evaluator simultaneously.

We develop six versions of ChatIoT using LLaMA3:8B, LLaMA3.1:8B, LLaMA3.1:70B, GPT-4o-mini, GPT-4o, and DeepSeek-R1 to cover a broad range of model capabilities and sizes. We include both open-source models (LLaMA series and DeepSeek-R1) and closed-source models (GPT-4o-mini, GPT-4o), which provide varying levels of reasoning ability and context-handling strength. We use LLaMA3.1:70B as the evaluator due to its strong instruction-following ability and high-quality reasoning. The prompt for evaluation is shown in Figure 8. The above settings ensure that all generated answers are evaluated within the same internal state of the evaluator, *a.k.a.*, LLaMA3:70B, which reduces the impact of LLM randomness and enables fair comparisons between i) ChatIoT and LLM-A with the same LLM and ii) ChatIoT versions with different LLMs as much as possible.

Table 5: Comparison of ChatIoT with LLM alone method (LLM-A). The experimental results are for the most advanced LLMs LLaMA3.1:70B, GPT-4o, and DeepSeek-R1. We use LLaMA3:70B as the evaluator for all experiments.

Role	Metric	LLaMA3.1:70B		GPT-4o		DeepSeek-R1	
		ChatIoT	LLM-A	ChatIoT	LLM-A	ChatIoT	LLM-A
Consumer	Reliability	4.40 (+0.70)	3.70	4.55 (+0.55)	4.00	4.90 (+0.98)	3.92
	Relevance	4.90 (+0.90)	4.00	4.85 (+0.75)	4.10	4.98 (+1.03)	3.95
	Technical	4.50 (+0.70)	3.80	4.40 (+0.25)	4.15	4.94 (+0.90)	4.04
	Friendliness	4.70 (+1.00)	3.70	4.80 (+0.75)	4.05	4.40 (+0.98)	3.42
Security Analyst	Reliability	4.70 (+0.65)	4.05	4.80 (+1.29)	3.51	4.80 (+0.72)	4.08
	Relevance	4.93 (+0.81)	4.12	4.83 (+1.18)	3.65	4.92 (+0.72)	4.20
	Technical	4.82 (+0.84)	3.98	4.83 (+1.18)	3.65	4.96 (+0.72)	4.24
	Friendliness	4.09 (+0.80)	3.29	4.14 (+0.61)	3.45	4.47 (+0.71)	3.76
Technical Officer	Reliability	4.45 (+0.37)	4.08	4.85 (+0.61)	4.24	5.00 (+1.10)	3.90
	Relevance	4.71 (+0.51)	4.20	4.88 (+0.63)	4.25	5.00 (+1.10)	3.90
	Technical	4.55 (+0.31)	4.24	4.88 (+0.65)	4.23	5.00 (+1.00)	4.00
	Friendliness	4.13 (+0.27)	3.86	4.49 (+0.61)	3.88	4.30 (+1.00)	3.30
Developer	Reliability	4.35 (−0.14)	4.49	4.75 (+0.93)	3.82	4.95 (+1.03)	3.92
	Relevance	4.54 (+0.02)	4.52	4.81 (+0.97)	3.84	4.98 (+0.93)	4.05
	Technical	4.52 (+0.02)	4.52	4.86 (+0.87)	3.99	4.99 (+0.93)	4.06
	Friendliness	3.96 (−0.24)	4.30	4.11 (+0.63)	3.48	4.12 (+1.02)	3.10
Trainer	Reliability	4.30 (−0.31)	4.61	4.40 (+0.17)	4.23	4.50 (+0.30)	4.20
	Relevance	4.62 (−0.08)	4.70	4.54 (+0.39)	4.15	4.80 (+0.30)	4.50
	Technical	4.31 (−0.23)	4.54	4.53 (+0.21)	4.32	4.90 (+0.50)	4.40
	Friendliness	4.25 (−0.40)	4.65	4.46 (+0.22)	4.24	4.60 (+0.30)	4.30

Table 5 presents the results for more advanced LLMs: LLaMA3.1:70B, GPT-4o, and DeepSeek-R1, and computes our improved scores over LLM-A. From these results, several key observations can be made:

- ChatIoT significantly enhances the moderate LLM's performance in the IoT security domain. As shown in Table 5, ChatIoT achieves higher scores across most metrics for use cases *Consumer*, *Security Analyst*, *Technical Officer*, and *Developer*. This is expected, as ChatIoT integrates domain-specific IoT security knowledge, *e.g.*, vulnerabilities, and tailors responses to be more user-friendly and relevant.
- However, ChatIoT does not always outperform the baseline LLMs. Taking the use case *Trainer*, when using LLaMA series models, ChatIoT even performs slightly worse; when using GPT-4o(-mini) and DeepSeek-R1, the improvements achieved by ChatIoT are much less than the other cases. This is likely due to the external data introduced in ChatIoT focusing mainly on vulnerabilities, exploits, and TTPs, while lacking sufficient information on course training materials. So, ChatIoT excels at producing technical and security-centric content over broader aspects like training programs.

The above analysis also highlights the importance of incorporating external knowledge to bolster LLMs in specialized domains. Fortunately, additional infor-

mation, such as training materials, can easily be integrated into ChatIoT using our **DataKit** toolkit.

5.4 Analysis of Human Evaluation

We recruit 13 participants for each of the five user roles defined. Participants were selected from non-technical officers, graduate students, academic researchers, and our industry partners, aligned with the expertise spectrum of the five roles (c.f. Appendix B for details). This recruitment strategy ensures coverage from non-technical users to domain experts, matching the intended role-aware design of our system. This approach prioritizes practicality and ecological validity, ensuring that our evaluation aligns with how users interact with security tools in real-world settings. By focusing on direct user preferences rather than rigid scoring, we aim to capture the true utility of ChatIoTin assisting diverse users with IoT security tasks.

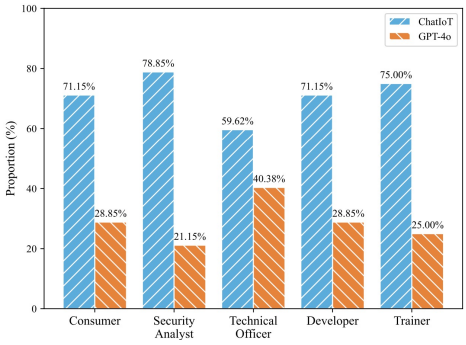

Fig. 9: Human evaluation of ChatIoT and LLM-A. The LLM is GPT-4o.

Figure 9 presents the results of human evaluations comparing ChatIoT with GPT-4o, noting that ChatIoT is built on top of GPT-4o in this experiment. From the experimental results, it is clear that ChatIoT consistently outperforms GPT-4o across all use cases. This aligns with expectations, as ChatIoT integrates additional IoT-specific intelligence into the LLM. Notably, ChatIoT demonstrates the greatest improvement for Security Analyst and the least for Technical Officer. The former result aligns with Table 5, where the comparison for Security Analyst shows the most significant difference. While Table 5 suggests that the least improvement is for Trainer, the human evaluation indicates that Technical Officer experiences the smallest gains because: i) The improvements for Technical Officer, though better than those for Trainer, particularly in top metric scores, may not be as easily discernible to humans as other use cases, making it harder for them to identify notable differences; ii) The Q&A tasks for Technical Officer are generally more complex and technical than Trainer, making it easier to select a better answer in Trainer case.

Remark 3. We acknowledge that LLM-based evaluation and Q&A survey may have inherent biases, as i) it is impossible to cover all LLMs and use cases. ii) participants' prior knowledge and personal preferences may also influence the results. We have made a strong effort to mitigate these biases by selecting representative LLMs, designing role-aware prompt templates that adapt to user backgrounds, and curating common and representative Q&A pairs for evaluation among users of diverse backgrounds. We believe this approach adequately reflects the improvements achieved.

5.5 Empirical Comparison to Vanilla RAG

Vanilla RAG typically relies on fixed, dense retrievers and does not support user role-aware generation, whereas ChatIoT introduces both adaptive retrieval and role-conditioned generation. Consequently, for a given query from a specific user, ChatIoT activates only the necessary retrievers and generates outputs tailored to the user's IoT expertise.

In Figure 11 (Appendix D), we compare ChatIoT with vanilla RAG for the query *"Can you provide a WiFi router with security CLS $\geq$ 1?"* by *Consumer*. The vanilla RAG outputs unnecessary technical WiFi router vulnerabilities because it indiscriminately activates all retrievers. In contrast, ChatIoT activates only the CLS-related retrievers and returns the requested devices, better aligning with the Consumer's needs and expertise level.

6 Related Work

Internet of Things (IoT) has seen rapid advancements in recent years, becoming an integral part of various domains, such as smart industries and homes, and serving as a key enabler in modern society. However, IoT continues to face numerous security challenges, prompting significant research efforts aimed at improving IoT security. With the rise of artificial intelligence (AI), machine learning (ML) and deep learning (DL)-based approaches have become increasingly popular in designing defense mechanisms for IoT devices, including malicious traffic classification [38, 48], malware detection [11, 13, 52], vulnerability discovery [42], and [8, 43, 51]. More recently, inspired by the success of large language models (LLMs), researchers have begun exploring the potential of LLMs to enhance IoT-related security tasks. For instance, LLMs have been applied to existing IoT security challenges such as threat detection and fuzzing. Ferrag *et al.* [22] introduced a BERT-based model, SecurityBERT, to achieve better cyber threat detection accuracy over traditional ML and DL-based methods. Similarly, Ma *et al.* [39] and Wang *et al.* [54] proposed LLM-assisted fuzzing methods to uncover hidden bugs in IoT devices, enabling the detection of complex vulnerabilities that traditional techniques might miss. Additionally, Yang *et al.* [58] combined LLMs with static code analysis using prompt engineering to create a cost-effective solution for IoT vulnerability detection. [31] collected cybersecurity raw texts to train cybersecurity LLM to augment the analysis of cybersecurity events, and [26] made use of a larger LLM to build knowledge graphs from public threat intelligence and use GPT to create datasets to fine-tune a smaller LLM to extract entities and TTPs from attack description. Ferraris *et al.* [24] utilized ChatGPT to enhance IoT trust semantics, aligning with W3C Web of Things (WoT) recommendations to extend TrUStAPIS [23].

Beyond the above tasks, LLMs have been employed in other IoT challenges. Meyuhas *et al.* [40] used LLMs to address labeling previously unseen IoT devices. [14, 45] explored leveraging LLMs to control IoT devices and facilitate effective collaboration among them. Mo *et al.* [41] collected IoT sensor-natural language paired data and trained IoT-LM to interpret and interact with physical IoT

sensors. Xu *et al.* [57] employed ChatGPT to interpret IoT sensor data and reason over tasks in the physical realm, introducing novel ways of integrating human knowledge into cyber-physical systems. Recently, Deldari *et al.* [16] proposed AuditNet, a conversational AI-based security assistant, which is most similar to ChatIoT and also augmented by external knowledge. However, AuditNet focused on standards, policies, and regulations of portable document format (PDF), and aimed to reduce the manual effort of security experts involved in compliance checks of IoT. Su et.al. [50] employed a hybrid RAG to improve their quality and complete specific tasks in terms of healthcare data management. [18] fine-tunes LLMs on cyber threat prediction to empower IoT security. [29] integrates RAG in IoT orchestration to enhance automation efficiency, personalization, and resilience. [28] leverage RAG to retrieve the specifications of individual IoT devices, and then assess the suitability of each vulnerability assessment item for every IoT device. On the other hand, we integrate IoT threat intelligence of various sources into ChatIoT and can assist different users. Besides, we provide an end-to-end toolkit to process data in various formats, including but not limited to PDF, JSON, etc. In this way, ChatIoT aims to narrow the time gap between emerging IoT security threats and broad, actionable responses.

Together, these studies indicate that LLMs have great potential to improve the security of IoT systems in various domains. By integrating LLMs with IoT-specific threat intelligence, these models can be guided to meet the unique challenges posed by the IoT ecosystem.

7 Conclusion

We propose ChatIoT, an LLM-based IoT security assistant, and conduct extensive evaluations on several common use cases. We leverage the advanced language understanding and reasoning capabilities of LLM and IoT security and threat information to provide IoT security assistance and develop an easy-to-use data toolkit to process different kinds of IoT datasets. With our design, ChatIoT is easily scalable to integrate different kinds of IoT security and threat intelligence. For future work, we plan to develop fully automated data processing toolkits, enhancing ChatIoT through LLM re-training/fine-tuning specifically for IoT security, like Sec-Gemini v1, and integrate more advanced evaluation approaches.

Acknowledgements

This research is supported by the National Research Foundation, Singapore, under its National Satellite of Excellence Programme "Design Science and Technology for Secure Critical Infrastructure: Phase II" (Award No: NRFNCR25-NSOE05-0001). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Research Foundation, Singapore.

References

1. Elasticsearch. software, version **6**(1) (2018)
2. Docker desktop, version 4.29.0. https://www.docker.com/products/docker-desktop (2024)
3. Langchain. https://github.com/langchain-ai/langchain (2024), version 0.2.5
4. Streamlit, version 1.33.0. https://streamlit.io (2024), accessed: 2024-09-02
5. Abdul-Ghani, H.A., Konstantas, D.: A comprehensive study of security and privacy guidelines, threats, and countermeasures: An iot perspective. Journal of Sensor and Actuator Networks **8**(2), 22 (2019)
6. Ahmad, R., Alsmadi, I.: Machine learning approaches to iot security: A systematic literature review. Internet of Things **14**, 100365 (2021)
7. Ahmed, S., Khan, M.: Securing the internet of things (iot): A comprehensive study on the intersection of cybersecurity, privacy, and connectivity in the iot ecosystem. AI, IoT and the Fourth Industrial Revolution Review **13**(9), 1–17 (2023)
8. Al-Garadi, M.A., Mohamed, A., Al-Ali, A.K., Du, X., Ali, I., Guizani, M.: A survey of machine and deep learning methods for internet of things (iot) security. IEEE communications surveys & tutorials **22**(3), 1646–1685 (2020)
9. Alrawi, O., Lever, C., Valakuzhy, K., Snow, K., Monrose, F., Antonakakis, M., et al.: The circle of life: A {large-scale} study of the {IoT} malware lifecycle. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 3505–3522 (2021)
10. Appalaraju, S., Jasani, B., Kota, B.U., Xie, Y., Manmatha, R.: Docformer: End-to-end transformer for document understanding. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 993–1003 (2021)
11. Aung, Y.L., Ochoa, M., Zhou, J.: Atlas: A practical attack detection and live malware analysis system for iot threat intelligence. In: International Conference on Information Security. pp. 319–338. Springer (2022)
12. Bou-Harb, E., Neshenko, N.: Cyber threat intelligence for the internet of things. Springer (2020)
13. Chaganti, R., Ravi, V., Pham, T.D.: Deep learning based cross architecture internet of things malware detection and classification. Computers & Security **120**, 102779 (2022)
14. Cui, H., Du, Y., Yang, Q., Shao, Y., Liew, S.C.: Llmind: Orchestrating ai and iot with llm for complex task execution. IEEE Communications Magazine (2024)
15. CVE Program: CVE® Program Mission. https://www.cve.org/ (nd)
16. Deldari, S., Goudarzi, M., Joshi, A., Shaghaghi, A., Finn, S., Salim, F.D., Jha, S.: Auditnet: A conversational ai-based security assistant. arXiv preprint arXiv:2407.14116 (2024)
17. Deogirikar, J., Vidhate, A.: Security attacks in iot: A survey. In: 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC). pp. 32–37. IEEE (2017)
18. Diaf, A., Korba, A.A., Karabadji, N.E., Ghamri-Doudane, Y.: Bartpredict: Empowering iot security with llm-driven cyber threat prediction. In: GLOBECOM 2024-2024 IEEE Global Communications Conference. pp. 1239–1244. IEEE (2024)
19. Dragomir, D., Gheorghe, L., Costea, S., Radovici, A.: A survey on secure communication protocols for iot systems. In: 2016 international workshop on Secure Internet of Things (SIoT). pp. 47–62. IEEE (2016)
20. Es, S., James, J., Espinosa-Anke, L., Schockaert, S.: Ragas: Automated evaluation of retrieval augmented generation. arXiv preprint arXiv:2309.15217 (2023)

21. Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.S., Li, Q.: A survey on rag meeting llms: Towards retrieval-augmented large language models. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 6491–6501 (2024)
22. Ferrag, M.A., Ndhlovu, M., Tihanyi, N., Cordeiro, L.C., Debbah, M., Lestable, T., Thandi, N.S.: Revolutionizing cyber threat detection with large language models: A privacy-preserving bert-based lightweight model for iot/iiot devices. IEEE Access **12**, 23733–23750 (2024). https://doi.org/10.1109/ACCESS.2024.3363469
23. Ferraris, D., Fernandez-Gago, C.: Trustapis: a trust requirements elicitation method for iot. International Journal of Information Security **19**(1), 111–127 (2020)
24. Ferraris, D., Kotis, K., Kalloniatis, C.: Enhancing trustapis methodology in the web of things with llm-generated iot trust semantics. In: 26th International Conference on Information and Communications Security (ICICS 2024). Springer, Mytilene, Lesvos, Greece (2024)
25. Görmüş, S., Aydın, H., Ulutaş, G.: Security for the internet of things: a survey of existing mechanisms, protocols and open research issues. Journal of the Faculty of Engineering and Architecture of Gazi University **33**(4), 1247–1272 (2018)
26. Hu, Y., Zou, F., Han, J., Sun, X., Wang, Y.: Llm-tikg: Threat intelligence knowledge graph construction utilizing large language model. Computers & Security **145**, 103999 (2024)
27. Iacovazzi, A., Wang, H., Butun, I., Raza, S.: Towards cyber threat intelligence for the iot. In: 2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT). pp. 483–490. IEEE (2023)
28. IKEGAMI, Y., HASEGAWA, K., HIDANO, S., FUKUSHIMA, K., HASHIMOTO, K., TOGAWA, N.: Prioritizing vulnerability assessment items for iot devices based on suitability evaluation using llms. IEICE Transactions on Information and Systems p. 2024EDP7325 (2025)
29. Jahanbakhsh, N., Vega-Barbas, M., Pau, I., Elvira-Martín, L., Moosavi, H., García-Vázquez, C.: Leveraging retrieval-augmented generation for automated smart home orchestration. Future Internet **17**(5), 198 (2025)
30. Janiszewski, M., Felkner, A., Lewandowski, P., Rytel, M., Romanowski, H.: Automatic actionable information processing and trust management towards safer internet of things. Sensors **21**(13) (2021). https://doi.org/10.3390/s21134359, https://www.mdpi.com/1424-8220/21/13/4359
31. Ji, H., Yang, J., Chai, L., Wei, C., Yang, L., Duan, Y., Wang, Y., Sun, T., Guo, H., Li, T., et al.: Sevenllm: Benchmarking, eliciting, and enhancing abilities of large language models in cyber threat intelligence. arXiv preprint arXiv:2405.03446 (2024)
32. Kim, G., Hong, T., Yim, M., Nam, J., Park, J., Yim, J., Hwang, W., Yun, S., Han, D., Park, S.: Ocr-free document understanding transformer. In: European Conference on Computer Vision. pp. 498–517. Springer (2022)
33. Kouicem, D.E., Bouabdallah, A., Lakhlef, H.: Internet of things security: A top-down survey. Computer Networks **141**, 199–221 (2018)
34. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., et al.: Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems **33**, 9459–9474 (2020)
35. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning (2023)
36. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: Llava-next: Improved reasoning, ocr, and world knowledge (January 2024)

37. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning (2023)
38. Luo, Y., Chen, X., Ge, N., Feng, W., Lu, J.: Transformer-based malicious traffic detection for internet of things. In: ICC 2022-IEEE International Conference on Communications. pp. 4187–4192. IEEE (2022)
39. Ma, X., Luo, L., Zeng, Q.: From one thousand pages of specification to unveiling hidden bugs: Large language model assisted fuzzing of matter IoT devices. In: 33rd USENIX Security Symposium (USENIX Security 24). pp. 4783–4800. USENIX Association, Philadelphia, PA (Aug 2024)
40. Meyuhas, B., Bremler-Barr, A., Shapira, T.: Iot device labeling using large language models. arXiv preprint arXiv:2403.01586 (2024)
41. Mo, S., Salakhutdinov, R., Morency, L.P., Liang, P.P.: Iot-lm: Large multisensory language models for the internet of things. arXiv preprint arXiv:2407.09801 (2024)
42. Neshenko, N., Bou-Harb, E., Crichigno, J., Kaddoum, G., Ghani, N.: Demystifying iot security: An exhaustive survey on iot vulnerabilities and a first empirical look on internet-scale iot exploitations. IEEE Communications Surveys & Tutorials **21**(3), 2702–2733 (2019)
43. Otoum, Y., Liu, D., Nayak, A.: Dl-ids: a deep learning–based intrusion detection framework for securing iot. Transactions on Emerging Telecommunications Technologies **33**(3), e3803 (2022)
44. Radford, A., Narasimhan, K.: Improving language understanding by generative pre-training (2018)
45. Rong, B., Rutagemwa, H.: Leveraging large language models for intelligent control of 6g integrated tn-ntn with iot service. IEEE Network **38**(4), 136–142 (2024). https://doi.org/10.1109/MNET.2024.3384013
46. Roziere, B., Gehring, J., Gloeckle, F., Sootla, S., Gat, I., Tan, X.E., Adi, Y., Liu, J., Sauvestre, R., Remez, T., et al.: Code llama: Open foundation models for code. arXiv preprint arXiv:2308.12950 (2023)
47. Saurabh, K., Sharma, V., Singh, U., Khondoker, R., Vyas, R., Vyas, O.: Hms-ids: Threat intelligence integration for zero-day exploits and advanced persistent threats in iiot. Arabian Journal for Science and Engineering pp. 1–21 (2024)
48. Shafiq, M., Tian, Z., Bashir, A.K., Du, X., Guizani, M.: Corrauc: a malicious bot-iot traffic detection method in iot network using machine-learning techniques. IEEE Internet of Things Journal **8**(5), 3242–3254 (2020)
49. Strom, B.E., Applebaum, A., Miller, D.P., Nickels, K.C., Pennington, A.G., Thomas, C.B.: Mitre att&ck: Design and philosophy. In: Technical report. The MITRE Corporation (2018)
50. Su, C., Wen, J., Kang, J., Wang, Y., Su, Y., Pan, H., Zhong, Z., Hossain, M.S.: Hybrid rag-empowered multi-modal llm for secure data management in internet of medical things: A diffusion-based contract approach. IEEE Internet of Things Journal (2024)
51. Tambe, A., Aung, Y.L., Sridharan, R., Ochoa, M., Tippenhauer, N.O., Shabtai, A., Elovici, Y.: Detection of threats to iot devices using scalable vpn-forwarded honeypots. In: Proceedings of the Ninth ACM Conference on Data and Application Security and Privacy. pp. 85–96 (2019)
52. Vasan, D., Alazab, M., Venkatraman, S., Akram, J., Qin, Z.: Mthael: Cross-architecture iot malware detection based on neural network advanced ensemble learning. IEEE Transactions on Computers **69**(11), 1654–1667 (2020)
53. Wagner, T.D., Mahbub, K., Palomar, E., Abdallah, A.E.: Cyber threat intelligence sharing: Survey and research directions. Computers & Security **87**, 101589 (2019)

54. Wang, J., Yu, L., Luo, X.: Llmif: Augmented large language model for fuzzing iot devices. In: 2024 IEEE Symposium on Security and Privacy (SP). pp. 881–896. IEEE Computer Society, Los Alamitos, CA, USA (may 2024). https://doi.org/10.1109/SP54263.2024.00211
55. Williams, R., McMahon, E., Samtani, S., Patton, M., Chen, H.: Identifying vulnerabilities of consumer internet of things (iot) devices: A scalable approach. In: 2017 IEEE International Conference on Intelligence and Security Informatics (ISI). pp. 179–181. IEEE (2017)
56. Wright, E., Lindsay, D., Wilkinson, G.: Regulating to protect security and privacy in the internet of things (iot): Draft report (2022)
57. Xu, H., Han, L., Yang, Q., Li, M., Srivastava, M.: Penetrative ai: Making llms comprehend the physical world. In: Proceedings of the 25th International Workshop on Mobile Computing Systems and Applications. pp. 1–7 (2024)
58. Yang, Y.: Iot software vulnerability detection techniques through large language model. In: International Conference on Formal Engineering Methods. pp. 285–290. Springer (2023)
59. Zhuge, M., Gao, D., Fan, D.P., Jin, L., Chen, B., Zhou, H., Qiu, M., Shao, L.: Kaleido-bert: Vision-language pre-training on fashion domain. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12647–12657 (2021)

A Chunking Strategies and LLMs-based Evaluation

Table 6 presents the chunking strategies experimental results and LLMs-based evaluation results of are in Table 7.

B Background Specifications and Participants

We present the background specifications for Security Analyst, Technical Officer, Developer, and Trainer in Figure 10. And the description of participants in our survey is as follows:

- Consumers: non-technical officers and general users with limited security background.
- Security Analysts: graduate students and junior researchers with foundational security knowledge.
- Technical Officers: school and laboratory officers familiar with operational or technical policy environments.
- Developers: participants from our industry partners and research labs with hands-on IoT/engineering experience.
- Trainers: lab and industry practitioners involved in IoT security education/outreach.

C Field Selection, Metadata, and Examples

Table 8 shows the field selection of VARIoT Exploits and MITRE ATT&CK ICS. The metadata information and examples for VATIoT exploits, MITRE ATT&CK ICS, and CLS are shown in Figure 12.

Table 6: Context (precision, recall) of different chunking configurations for VARIoT Vulnerabilities, Exploits, MITRE ATT&CK ICS, and Threat Reports. The best (precision, recall) in our experimental settings are marked in bold.

(Size, Overlap)	VARIoT Vulnerabilities			VARIoT Exploits		
	Character	RecurChar	TokenText	Character	RecurChar	TokenText
(500,50)	(0.875, 0.906)	(0.981, 0.898)	(0.936, 0.896)	(0.937, 0.903)	(0.927, 0.922)	(0.927, 0.896)
(500,100)	(0.876, 0.921)	**(0.986, 0.973)**	(0.902, 0.925)	(0.918, 0.902)	(0.883, 0.908)	(0.910, 0.935)
(500,150)	(0.917, 0.909)	(0.983, 0.955)	(0.929, 0.917)	(0.918, 0.895)	(0.912, 0.823)	(0.924, 0.884)
(500,200)	(0.929, 0.891)	(0.913, 0.935)	(0.937, 0.897)	(0.927, 0.913)	(0.913, 0.867)	(0.901, 0.955)
(1000,50)	(0.905, 0.867)	(0.955, 0.892)	(0.915, 0.902)	(0.799, 0.898)	(0.871, 0.811)	(0.937, 0.920)
(1000,100)	(0.883, 0.883)	(0.923, 0.897)	(0.888, 0.843)	(0.794, 0.908)	(0.878, 0.753)	(0.941, 0.920)
(1000,150)	(0.891, 0.897)	(0.916, 0.959)	(0.892, 0.873)	(0.908, 0.751)	(0.920, 0.722)	**(0.943, 0.941)**
(1000,200)	(0.891, 0.918)	(0.913, 0.921)	(0.899, 0.883)	(0.892, 0.846)	(0.804, 0.864)	(0.935, 0.925)
(1500,50)	(0.895, 0.882)	(0.893, 0.922)	(0.918, 0.829)	(0.851, 0.920)	(0.922, 0.861)	(0.917, 0.962)
(1500,100)	(0.828, 0.881)	(0.891, 0.913)	(0.901, 0.831)	(0.872, 0.862)	(0.898, 0.904)	(0.934, 0.923)
(1500,150)	(0.894, 0.855)	(0.913, 0.852)	(0.891, 0.847)	(0.824, 0.904)	(0.917, 0.898)	(0.910, 0.943)
(1500,200)	(0.842, 0.865)	(0.922, 0.896)	(0.901, 0.835)	(0.899, 0.845)	(0.880, 0.934)	(0.943, 0.913)
(2000,50)	(0.929, 0.849)	(0.913, 0.886)	(0.899, 0.841)	(0.897, 0.896)	(0.924, 0.934)	(0.897, 0.943)
(2000,100)	(0.903, 0.835)	(0.886, 0.882)	(0.879, 0.865)	(0.926, 0.893)	(0.918, 0.932)	(0.925, 0.918)
(2000,150)	(0.913, 0.853)	(0.861, 0.890)	(0.916, 0.855)	(0.863, 0.937)	(0.912, 0.938)	(0.938, 0.905)
(2000,200)	(0.859, 0.881)	(0.907, 0.906)	(0.923, 0.831)	(0.913, 0.915)	(0.950, 0.929)	(0.899, 0.954)

(Size, Overlap)	MITRE ATT&CK ICS			Threat Report		
	Character	RecurChar	TokenText	Character	RecurChar	TokenText
(500,50)	(0.923, 0.912)	(0.903, 0.922)	(0.901, 0.918)	(0.842, 0.869)	(0.867, 0.856)	(0.925, 0.921)
(500,100)	(0.918, 0.915)	(0.893, 0.911)	(0.880, 0.925)	(0.869, 0.890)	(0.915, 0.853)	(0.936, 0.942)
(500,150)	(0.920, 0.916)	(0.952, 0.888)	(0.887, 0.932)	(0.862, 0.869)	(0.930, 0.863)	(0.880, 0.930)
(500,200)	(0.918, 0.876)	(0.902, 0.910)	(0.929, 0.884)	(0.858, 0.864)	(0.946, 0.860)	**(0.961, 0.948)**
(1000,50)	(0.954, 0.905)	(0.927, 0.900)	(0.887, 0.929)	(0.934, 0.915)	(0.910, 0.925)	(0.885, 0.858)
(1000,100)	(0.946, 0.935)	(0.932, 0.924)	(0.889, 0.929)	(0.937, 0.900)	(0.913, 0.921)	(0.847, 0.833)
(1000,150)	(0.962, 0.922)	(0.901, 0.927)	(0.870, 0.946)	(0.923, 0.875)	(0.920, 0.912)	(0.844, 0.932)
(1000,200)	**(0.942, 0.958)**	(0.923, 0.886)	(0.887, 0.929)	(0.894, 0.934)	(0.944, 0.842)	(0.782, 0.806)
(1500,50)	(0.894, 0.898)	(0.905, 0.922)	(0.896, 0.925)	(0.965, 0.812)	(0.957, 0.866)	(0.885, 0.781)
(1500,100)	(0.917, 0.886)	(0.925, 0.923)	(0.924, 0.913)	(0.898, 0.888)	(0.915, 0.918)	(0.823, 0.907)
(1500,150)	(0.918, 0.893)	(0.923, 0.933)	(0.922, 0.898)	(0.910, 0.862)	(0.925, 0.883)	(0.833, 0.849)
(1500,200)	(0.913, 0.898)	(0.911, 0.928)	(0.884, 0.932)	(0.894, 0.934)	(0.944, 0.842)	(0.782, 0.806)
(2000,50)	(0.912, 0.880)	(0.901, 0.901)	(0.911, 0.901)	(0.791, 0.916)	(0.872, 0.879)	(0.832, 0.867)
(2000,100)	(0.904, 0.886)	(0.892, 0.904)	(0.884, 0.935)	(0.871, 0.890)	(0.797, 0.884)	(0.913, 0.839)
(2000,150)	(0.906, 0.908)	(0.927, 0.899)	(0.931, 0.888)	(0.929, 0.935)	(0.872, 0.841)	(0.903, 0.891)
(2000,200)	(0.920, 0.891)	(0.909, 0.891)	(0.908, 0.904)	(0.925, 0.898)	(0.870, 0.901)	(0.891, 0.842)

Table 7: Comparison of ChatIoT with LLM alone method (LLM-A). The experimental results are for moderate LLMs: LLaMA3:8B, LLaMA3.1:8B, and GPT-4o-mini. We use LLaMA3:70B as the evaluator for all experiments.

Role	Metric	LLaMA3:8B		LLaMA3.1:8B		GPT-4o-mini	
		ChatIoT	LLM-A	ChatIoT	LLM-A	ChatIoT	LLM-A
Consumer	Reliability	4.10 (+0.11)	3.99	4.50 (+0.80)	3.70	4.70 (+0.80)	3.90
	Relevance	4.90 (+0.65)	4.25	4.90 (+0.80)	4.10	5.00 (+1.00)	4.00
	Technical	4.50 (+0.47)	4.03	4.40 (+0.80)	3.60	4.45 (+0.55)	3.90
	Friendliness	4.30 (+0.30)	4.00	4.70 (+0.90)	3.80	4.90 (+1.00)	3.90
Security Analyst	Reliability	4.30 (+0.26)	4.04	4.75 (+0.67)	4.08	4.85 (+1.19)	3.66
	Relevance	4.63 (+0.33)	4.30	4.95 (+0.87)	4.08	4.91 (+1.21)	3.70
	Technical	4.47 (+0.23)	4.24	4.78 (+0.79)	3.99	4.89 (+1.18)	3.71
	Friendliness	4.04 (+0.01)	4.03	4.02 (+0.87)	3.15	4.92 (+1.43)	3.49
Technical Officer	Reliability	4.45 (+0.63)	3.82	4.45 (+0.37)	4.08	4.80 (+0.76)	4.04
	Relevance	4.78 (+0.73)	4.05	4.65 (+0.47)	4.18	4.90 (+0.80)	4.10
	Technical	4.59 (+0.83)	3.76	4.48 (+0.29)	4.19	4.83 (+0.68)	4.15
	Friendliness	3.92 (+0.44)	3.48	4.22 (+0.27)	3.95	4.67 (+0.62)	4.05
Developer	Reliability	4.43 (+0.63)	3.80	4.40 (+0.10)	4.30	4.80 (+0.79)	4.01
	Relevance	4.66 (+0.83)	3.83	4.60 (+0.30)	4.30	4.86 (+0.84)	4.02
	Technical	4.67 (+0.62)	4.05	4.50 (+0.10)	4.40	4.94 (+0.92)	4.02
	Friendliness	4.01 (+0.09)	3.92	3.90 (−0.50)	4.40	4.18 (+0.48)	3.70
Trainer	Reliability	4.10 (−0.14)	4.24	4.30 (−0.19)	4.49	4.55 (+0.29)	4.26
	Relevance	4.56 (+0.16)	4.40	4.40 (−0.11)	4.51	4.64 (+0.24)	4.40
	Technical	4.11 (+0.07)	4.04	4.42 (−0.07)	4.49	4.58 (+0.36)	4.22
	Friendliness	4.02 (−0.03)	4.05	4.08 (−0.48)	4.56	4.33 (+0.11)	4.22

Background of Security Analyst

"Knowledge": "Security Analyst is an expert in identifying vulnerabilities, analyzing threats, and ensuring IoT devices are secure from cyber threats. Security Analyst possesses in-depth technical knowledge of security protocols, vulnerabilities, and exploits, and are proficient in interpreting complex security data."

"Goals": "The primary aim is to conduct in-depth analyses of IoT security threats, vulnerabilities, and exploits, contributing to the development of secure IoT systems, and provide deep insights into potential attack vectors, technical analysis, and mitigation strategies."

"Requirements": "Security Analyst requires detailed information about the vulnerabilities, exploits, and technical configurations of IoT devices."

Background of Technical Officer

"Knowledge": "Technical Officer is familiar with security patch management, ensuring devices adhere to organizational security standards, and handling technical troubleshooting. "

"Goals": "Technical Officer is responsible for overseeing the implementation and maintenance of secure IoT systems within an organization, applying security patches, enforcing security policies, and troubleshooting security issues."

"Requirements": "Technical Officer's focus is on implementing security measures within the organization's infrastructure. You need practical steps to deploy security updates and verify compliance with security standards."

Background of Developer

"Knowledge": "Developer works on the technical design and architecture of IoT devices, with a focus on incorporating security into product design. Developer has a deep understanding of device security, encryption protocols, and compliance with security regulations."

"Goals": "Developer is responsible for ensuring IoT products meet industry security standards and are resilient against known threats, and designing and developing secure IoT devices."

"Requirements": "Developer needs insights into current vulnerabilities, designs best practices, and how to avoid common security pitfalls in future product iterations."

Background of Trainer

"Knowledge": "Trainer creates educational material or conducts training sessions to teach IoT security to a broader audience, including technical and non-technical participants. Trainer understands both technical and pedagogical aspects of IoT security and can explain complex concepts in a simplified manner."

"Goals": "Trainer aims to guide others in the best practices for IoT security, helping to raise awareness and improve security practices across different user groups."

"Requirements": "Trainer needs information that can be used in a training environment, with clear examples, case studies, and simplified explanations for different levels of learners."

Fig. 10: The background specifications for Security Analyst, Technical Officer, Developer, and Trainer, utilized to guide the ChatIoT to generate answers.

D Use Case Study and Compared to Vanilla RAG

First, we compare ChatIoT to GPT-4o only for the Security Analyst case study in Figure 13 and 14. Our key advantages are as follows: **Reliability**: GPT-4o provides general vulnerabilities and examples but lacks specific details, such as CVE references for many vulnerabilities. It misses certain critical vulnerabilities. ChatIoT offers precise information about vulnerabilities, including affected models and specific risks. It references real-world CVEs, enhancing factual reliability. **Relevance**: GPT-4o covers a broad range of vulnerabilities but does not specifically target the needs of a security analyst, making it less relevant for professionals. ChatIoT's answer is tailored for a security analyst, focusing on vulnerabilities impacting network security. It provides detailed exploit analysis and mitigation strategies, making it highly relevant. **Technical**: GPT-4o is a general LLM, so it lacks comprehensive technical details about each vulnerability

Table 8: Large Language Models-based field selection for page_content and metadata of VARIoT Vulnerabilities, Exploits, and MITRE ATT&CK ICS. ✔ is for page_content, ◯ denotes metadata, and ✘ indicates unused fields.

Datasets	Fields	LLaMA3:8B	LLaMA3.1:70B	GPT-4o	Datasets	Fields	LLaMA3:8B	LLaMA3.1:70B	GPT-4o
Vulnerabilities	cve	✘	◯	◯	MITRE ATT&CK ICS	stixId	◯	◯	◯
	id	◯	◯	◯		name	✔	✔	✔
	credit	✔	✘	◯		parentName	✔	✘	✔
	description	✔	✔	✔		description	✔	✔	✔
	title	✔	✔	✔		lunrRef	✘	✘	◯
	products	◯	◯	◯		id	◯	✘	◯
	vulns.-config.	◯	✔	◯		url	✘	✘	✘
	cvss-score	✘	✘	◯		is_enterprise	✘	◯	◯
	cvss-string	✘	◯	✔		type	✘	◯	◯
	reference	✘	✔	✔		relatedTech.	✘	◯	✘
	published	✘	◯	✘		is_ics	✘	✘	✘
	modified	✘	◯	✘		deprecated	✘	✘	◯
Exploits	id	◯	◯	◯	Exploits	credit	✘	◯	◯
	description	✔	✔	✔		exploit	✔	✔	✔
	title	✔	✔	✔		cve-id	◯	◯	◯
	reference	✘	◯	✔		products	◯	◯	◯

I'd be happy to help you find a secure WiFi router with CLS $\geq$ 1.
Based on the vulnerabilities and exploits I analyzed, there are serious WiFi router threats as follows:

- A firmware issue in some models of D-Link (specifically the DIR-852) allows remote command injection via malformed arguments to /gena.cgi.
- For Tenda routers — e.g. the AC21 model — multiple buffer-overflow vulnerabilities were discovered (e.g. via /goform/SetSysTimeCfg, or setPptpUserList, or SetVirtualServerCfg) that allow denial-of-service (DoS) or potentially remote code execution if exploited.

Examples WiFi routers with CLS $\geq$ 1:

- D-Link AX3000 Dual-Band Wi-Fi 6: This router has a CLs Level 1 rating and supports WPA2-PSK (AES)encryption. It's a great option for home users who want a secure and reliable WiFi connection.
- TP-Link BE18000 Whole Home Mesh Wi-Fi 7 AP: This router also has a CLs Level 1 rating and supports WPA2-PSK (AES) encryption. It's a great option for larger homes or ofices that reguire a strong andsecure WiFi signal.

Disclaimer
The recommendations provided in this report are based on the information available at the time of writing. We are not legally responsible for any actions taken based on these recommendations. Always consult with a professional security expert before implementing any security measures.

I'd be happy to suggest some secure WiFi router options with CLS $\geq$1:

- D-Link AX3000 Dual-Band Wi-Fi 6: This router has a CLs Level 1 rating and supports WPA2-PSK (AES)encryption. It's a great option for home users who want a secure and reliable WiFi connection.
- ASUS RT-AX82U WiFi 6 Router: Part of the line of ASUS WiFi 6 routers; ASUS has several models with CLS Level 4 (highest cert level under CLS).

Disclaimer: A router model has had CLS certification doesn't guarantee perpetual security — the label remains valid for a limited period (manufacturers must continue issuing firmware updates, etc.).

Fig. 11: Answers of vanilla RAG (above) and ChatIoT (below) for (User Role=Consumer, Query="Can you provide a WiFi router with CLS $\geq$ 1").

and does not explain them in depth or provide mitigation strategies. ChatIoT offers detailed technical specifics for each vulnerability, including descriptions, risks, and mitigation strategies. The structured analysis of exploits is suitable for technical audiences. **Friendliness**: GPT-4o uses straightforward language, making it easy for a general audience to understand, but lacks engagement for detailed insights. Our approach maintains a professional tone while being informative. The structured layout enhances readability, making it user-friendly for professionals seeking specific information.

In Figure 11, we compare ChatIoT to vanilla RAG using GPT-4o.

```
# metadata fields information and examples for the self-querying retriever corresponding to dataset VARIoT exploits.
exploits_metadata_field_info = [
    AttributeInfo(
        name="products",
        description="The affected products.",
        type="string or list[string]",
    ),
    AttributeInfo(
        name="id",
        description="The ID of the entry in the VarIoT Exploits dataset. ",
        type="string",
    ),
]
exploits_examples = [
    (
        "Provide exploits related to D-Link routers.",
        {
            "query": "exploits related to D-Link routers",
            "filter": "contain(\"products\", \"D-Link routers\")",
        },
    ),
    (
        "Describe the exploits with ID VAR-E-201704-0089.",
        {
            "query": "content with ID VAR-E-201704-0089",
            "filter": "eq(\"id\", \"VAR-E-201704-0089\")",
        },
    ),
]

# metadata fields information and examples for the self-querying retriever corresponding to MITRE ATT&CK ICS
ics_metadata_field_info = [
    AttributeInfo(
        name="stixId",
        description="The stixId of the entry in the MITRE ICS dataset.",
        type="string",
    ),
]
ics_examples = [
    (
        "Provide the TTPs with stixId attack-pattern--008b8f56-6107-48be-aa9f-746f927dbb61.",
        {
            "query": "ICS TTPs with stixId attack-pattern--008b8f56-6107-48be-aa9f-746f927dbb61",
            "filter": "eq(\"stixId\", \"attack-pattern--008b8f56-6107-48be-aa9f-746f927dbb61\")"
        },
    ),
]

cls_metadata_field_info = [
    AttributeInfo(
        name="category",
        description="The category of products, for example, WiFi Router",
        type="string",
    ),
    AttributeInfo(
        name="manufacturer",
        description="The manufacturer of the product, for example, TP-Link.",
        type="string",
    ),
    AttributeInfo(
        name="Model",
        description="The Model No. of product, for example, AP5100W-D171",
        type="string",
    ),
    AttributeInfo(
        name="CLS Level",
        description="The CLS level of product, should be [1,2,3,4], and higher is better.",
        type="integer",
    ),
]
cls_examples = [
    (
        "Provide WiFi Mesh with Model AP5100W-D171.",
        {
            "query": "Category and Model",
            "filter": "and(contain(\"category\", \"WiFi Mesh\"), contain(\"Model\", \"AP5100W-D171\"))",
        },
    ),
    (
        "Provide WiFi router of D-Link with CLS level no less than 2.",
        {
            "query": "category of a manufacturer with CLS level",
            "filter": "and(contain(\"category\", \"WiFi router\"), contain(\"manufacturer\", \"D-Link\"), gte(\"CLS Level\",
            ↪  2))",
        },
    ),
]
```

Fig. 12: The metadata field information and examples for VARIoT exploits and MITRE ATT&CK ICS.

TP-Link routers, like many other networking devices, have been subject to various vulnerabilities and exploits over the years. These vulnerabilities can range from simple misconfigurations to more complex software flaws that can be exploited by attackers. Here are some notable vulnerabilities and exploits associated with TP-Link routers:

1. Default Credentials and Weak Passwords
Many TP-Link routers come with default administrative credentials (e.g., admin/admin). If these credentials are not changed, attackers can easily gain access to the router's administrative interface. Weak passwords can also be brute-forced.

2. Remote Code Execution (RCE)
Several TP-Link router models have been found to have vulnerabilities that allow remote code execution. For example:

- **CVE-2019-7406**: A vulnerability in the TP-Link WR940N allows remote attackers to execute arbitrary code via a crafted HTTP request.
- **CVE-2020-12109**: A buffer overflow vulnerability in the TP-Link Archer C5 allows remote attackers to execute arbitrary code.

3. Cross-Site Scripting (XSS)
Cross-Site Scripting vulnerabilities can allow attackers to inject malicious scripts into the web interface of the router. For example:

- **CVE-2019-7405**: An XSS vulnerability in the TP-Link WR940N allows remote attackers to inject arbitrary web script or HTML.

4. Cross-Site Request Forgery (CSRF)
CSRF vulnerabilities can allow attackers to perform actions on behalf of the authenticated user without their consent. For example:

- **CVE-2018-11714**: A CSRF vulnerability in the TP-Link Archer C1200 allows remote attackers to change the router's settings.

5. Command Injection
Command injection vulnerabilities allow attackers to execute arbitrary commands on the router's operating system. For example:

- **CVE-2017-13772**: A command injection vulnerability in the TP-Link WR1043ND allows remote attackers to execute arbitrary commands via the ping feature.

6. Information Disclosure
Information disclosure vulnerabilities can expose sensitive information about the router or the network. For example:

- **CVE-2019-7404**: An information disclosure vulnerability in the TP-Link WR940N allows remote attackers to obtain sensitive information via a crafted HTTP request.

7. Firmware Vulnerabilities
Outdated firmware can contain numerous vulnerabilities that have been patched in later versions. It's crucial to keep the router's firmware up to date to mitigate these risks.

8. Misconfigurations
Improper configurations, such as enabling remote management without proper security measures, can expose the router to external attacks.

Mitigation Strategies
To protect against these vulnerabilities, consider the following best practices:

- **Change Default Credentials**: Always change the default username and password to something strong and unique.
- **Update Firmware**: Regularly check for and apply firmware updates from TP-Link.
- **Disable Unnecessary Services**: Turn off services like remote management if they are not needed.
- **Use Strong Encryption**: Ensure that Wi-Fi networks use strong encryption methods like WPA3.
- **Network Segmentation**: Segment your network to limit the impact of a compromised device.
- **Monitor Network Traffic**: Use network monitoring tools to detect unusual activity.

Conclusion
TP-Link routers, like any other network devices, can be vulnerable to various types of attacks if not properly secured. Regular updates, strong passwords, and proper configuration can significantly reduce the risk of exploitation. Always stay informed about the latest vulnerabilities and apply necessary patches and mitigations promptly.

Fig. 13: Generated answers of GPT-4o for the inputs (User Role=Security Analyst, Query="Investigate the vulnerabilities and exploits associated with the TP-Link Routers").

Technical Analysis of Vulnerabilities and Exploits in TP-Link Routers

Overview
TP-Link routers have been identified with multiple security vulnerabilities that pose significant risks to users. This analysis focuses on the vulnerabilities, associated risks, and actionable security measures to mitigate these threats.

Identified Vulnerabilities

1. **Denial of Service (DoS) Vulnerability**
   - **Affected Models**: TP-Link Archer C2, Archer C20i
   - **Description**: This vulnerability allows attackers to disrupt the normal functioning of the router, rendering it unavailable to legitimate users.
   - **Risk**: High. A successful DoS attack can lead to network downtime, affecting all connected devices.
   - **Mitigation**: Implement rate limiting and anomaly detection to identify and block malicious traffic. Regularly update firmware to patch known vulnerabilities.
2. **Security Bypass Vulnerability**
   - **Affected Models**: TP-Link Archer C2, Archer C20i
   - **Description**: This vulnerability allows attackers to bypass security mechanisms, gaining unauthorized access to the router.
   - **Risk**: High. Unauthorized access can lead to data breaches and further exploitation.
   - **Mitigation**: Enable strong authentication mechanisms, such as multi-factor authentication (MFA). Regularly review and update access control lists (ACLs).
3. **Cross-Site Request Forgery (CSRF)**
   - **Affected Models**: TP-Link TL-WR1043N
   - **Description**: This vulnerability allows attackers to perform unauthorized actions on behalf of authenticated users by tricking them into clicking malicious links.
   - **Risk**: Medium. CSRF can lead unauthorized configuration changes, e.g., altering admin passwords.
   - **Mitigation**: Implement anti-CSRF tokens in web forms. Educate users about the risks of clicking on unknown links.
4. **Path Traversal Vulnerability**
   - **Affected Models**: TP-Link TL-R600VPN
   - **Description**: This allows attackers to access sensitive system files by manipulating URL.
   - **Risk**: High. Unauthorized access to system files leads to information disclosure and exploitation.
   - **Mitigation**: Validate and sanitize user inputs to prevent path traversal attacks. Regularly update firmware to patch known vulnerabilities.
5. **Command Injection Vulnerability**
   - **Affected Models**: TP-Link Archer C2, Archer C20i
   - **Description**: This vulnerability allows attackers to execute arbitrary commands on router.
   - **Risk**: Critical. Command injection can lead to complete system compromise.
   - **Mitigation**: Implement input validation and sanitization. Use security mechanisms like Web Application Firewalls (WAF) to detect and block malicious inputs.

Exploits Analysis

1. **Proof-of-Concept (PoC) Exploits**
   - **Description**: Researchers developed PoCs to demonstrate feasibility of exploiting vulnerabilities.
   - **Risk**: High. PoCs can be used by malicious actors to develop real-world attacks.
   - **Mitigation**: Monitor security advisories and apply patches promptly. Conduct regular security assessments to identify and mitigate potential vulnerabilities.
2. **Unauthenticated Reflected XSS**
   - **Affected Models**: TP-Link TD-W8951ND
   - **Description**: This vulnerability allows attackers to inject malicious scripts into the browser of an unauthenticated user.
   - **Risk**: Medium. XSS can lead to session hijacking and unauthorized actions.
   - **Mitigation**: Implement CSP and sanitize user inputs. Update firmware for known vulnerabilities.
3. **Authenticated Reflected XSS**
   - **Affected Models**: TP-Link TD-W8951ND
   - **Description**: This vulnerability allows attackers to inject malicious scripts into the browser of an authenticated user.
   - **Risk**: Medium. XSS can lead to session hijacking and unauthorized actions.
   - **Mitigation**: Implement CSP and sanitize user inputs. Educate the risks of clicking unknown links.
4. **Reset Admin Password CSRF**
   - **Affected Models**: TP-Link TD-W8951ND
   - **Description**: This vulnerability allows attackers to reset the admin password by tricking the administrator into clicking a malicious link.
   - **Risk**: High. Unauthorized access to the admin account can lead to complete system compromise.
   - **Mitigation**: Implement anti-CSRF tokens and educate the risks of clicking unknown links.

Actionable Security Measures

1. **Firmware Updates** Regularly update router firmware to patch known vulnerabilities. Enable automatic updates if available.
2. **Strong Authentication** Use strong, unique passwords for admin accounts. Enable multi-factor authentication (MFA) where possible.
3. **Network Segmentation** Segment the network to limit the impact of a compromised device. Use VLANs to isolate sensitive devices.
4. **Monitoring and Logging** Implement network monitoring to detect and respond to suspicious activities. Enable logging to maintain an audit trail of administrative actions.
5. **User Education** Educate users about the risks of phishing and social engineering attacks. Encourage users to report suspicious activities.

Disclaimer
The recommendations provided in this report are based on the information available at the time of writing. We are not legally responsible for any actions taken based on these recommendations. Always consult with a professional security expert before implementing any security measures.

Fig. 14: Answers of ChatIoT for (User Role=Security Analyst, Query="Investigate the vulnerabilities and exploits associated with the TP-Link Routers").