Regular Paper

Communications of the CCISA
Vol. 31 No. 3 Aug. 2025

Retrieval-Augmented Generation for Identifying ATT&CK Technique

Sheng-Shan Chen [1], Kai-Siang Cao [2], Chung-Kuan Chen[3], Chin-Yu Sun [4*]
[1,4] Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei, Taiwan
[2] Department of Innovation Frontier Institute of Research for Science and Technology, National Taipei University of Technology, Taipei, Taiwan
[3] CyCraft Technology, Taipei, Taiwan
[1]t111599004@ntut.edu.tw, [2]t112C72005, [4]cysun@ntut.edu.tw, [3]ck.chen@cycraft.com,

Abstract

Cyber Threat Intelligence (CTI) analysis faces significant challenges due to the scale and complexity of threat data. Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) offer promising solutions; however, existing approaches often struggle with limited accuracy and hallucination. We propose an enhanced RAG framework that incorporates fine-tuned BERT embeddings for semantic retrieval and technique annotation, coupled with structured prompt generation to guide LLMs toward more precise and context-aware threat analysis. Compared with traditional encoder-only architectures, our framework substantially improves both accuracy and efficiency. Experiments conducted on the MITRE ATT&CK database and recent open-source threat reports demonstrate that our model achieves an F1-score of 0.93, outperforming state-of-the-art baselines including GPT-4 and LLaMA-3. These results highlight the potential of advanced RAG architectures to enable scalable, accurate, and trustworthy automated CTI analysis.

Keywords: Cyber Threat Intelligence, Large Language Models, Retrieval-Augmented Generation, BERT Embeddings, MITRE ATT&CK, Semantic Similarity

* Corresponding author.

1. Introduction

With the rapid advancement of internet technologies, enterprises and organizations are increasingly exposed to escalating cyber threats. To mitigate these risks, cybersecurity analysts rely on Cyber Threat Intelligence (CTI) to anticipate potential attacks and develop defense strategies, thereby improving the defensive posture of organizations [1, 2]. To systematically identify and classify such threats, frameworks such as MITRE ATT&CK [3] are widely adopted. MITRE ATT&CK, developed by the MITRE Corporation, is a globally recognized knowledge base that categorizes cyber adversary behavior using the Tactics, Techniques, and Procedures (TTPs) taxonomy. Tactics describe the overarching objectives or strategic goals of an adversary; techniques specify the methods employed to achieve these objectives; procedures provide detailed, step-by-step implementations of techniques in real-world scenarios.

TTPs offer a comprehensive representation of adversarial attack patterns, enabling analysts to accurately pinpoint ongoing malicious activities and develop timely countermeasures. The rapid growth of the Internet and social media platforms has dramatically increased the volume of CTI [4]. However, discrepancies in CTI formats and quality across sources remain a significant challenge [5]. For instance, unstructured CTI sources such as AlienVault [6] and Twitter [7] often contain up-to-date findings from security analysts, but the content and scope depend heavily on the discoverer's focus. These inconsistencies increase the difficulty of analysis, potentially lengthening response times and reducing detection accuracy. Traditional manual classification of TTPs is a time-consuming task, requiring significant human expertise and time. As the volume and complexity of CTI reports grow, this manual effort risks overlooking critical attacker intent in real time, potentially leading to delayed or ineffective defensive actions [8]. To address these issues, automated AI-based classification models for techniques within TTPs can substantially reduce analyst workload, improve classification accuracy, and enhance the timeliness of threat detection. Such automation enables faster and more informed incident responses, thereby strengthening overall cybersecurity resilience.

The emergence of Large Language Models (LLMs) has opened new opportunities for CTI analysis. Models such as GPT-3.5 [9], built upon the Transformer architecture [10], have shown remarkable advances in semantic understanding and text generation. The Transformer's attention mechanism allows parallel processing across sequences, significantly improving efficiency. Encoder-based models (e.g., Bidirectional Encoder Representations from Transformers, BERT [11]) capture contextual semantics for text understanding, while decoder-based models (e.g., ChatGPT [12]) excel in natural language generation. BERT is trained through unsupervised pre-training tasks on large-scale general-domain corpora such as BooksCorpus [13] and Wikipedia [14], capturing fundamental semantic and syntactic structures

of natural language. However, to achieve optimal performance in domain-specific applications, fine-tuning is essential. This process adapts the model's general linguistic knowledge to specialized domains such as cybersecurity, medical text analysis, and sentiment detection. Despite their strengths, decoder-based models have been underutilized in CTI classification due to the risk of hallucination, the tendency to generate plausible but incorrect outputs with high confidence [15].

To address this limitation, Lewis et al. [16] introduced Retrieval-Augmented Generation (RAG), a hybrid approach that integrates external knowledge retrieval with generative models. By incorporating domain-specific evidence during generation, RAG mitigates hallucinations and produces more accurate, factually grounded outputs. Building on this paradigm, the present study proposes an enhanced RAG framework tailored for CTI classification, particularly focusing on TTPs. The key contributions of our approach are threefold: (1) fine-tuned BERT embeddings are employed to enrich contextual representations of cybersecurity-specific terms, thereby improving retrieval precision and classification accuracy; (2) LangChain [17] -based structured prompting is incorporated to guide response generation, ensuring that the model explicitly acknowledges uncertainty rather than producing misleading classifications; and (3) empirical validation using a dataset derived from the MITRE ATT&CK framework [18] demonstrates that the proposed method achieves an F1-score of 93%, outperforming conventional deep learning-based CTI classification methods.

The remainder of this paper is organized as follows. Section 2 reviews related work in CTI classification and compares existing methods with the proposed approach. Section 3 discusses the challenges of applying LLMs to CTI and explains how RAG addresses hallucination issues. Section 4 details the proposed method of the enhanced RAG framework, including architectural design, embedding strategies, and its application to CTI. Section 5 outlines the experimental setup, dataset preprocessing, and presents the results with performance analysis. Finally, Section 6 concludes with a summary of contributions.

2. Related Work

This section reviews the evolution of automated techniques for extracting and classifying TTPs in CTI. The discussion begins with traditional rule-based approaches, progresses through early machine learning methods, and concludes with the most recent advances in LLMs. Each category is examined in terms of its suitability for TTP extraction tasks, with particular attention to its strengths, limitations, and the challenges that motivate this study.

2.1 Traditional Rule-Based and Early Machine Learning Methods

Early research on TTP extraction and classification can generally be divided into rule-based and machine learning approaches. Rule-based methods rely on expert-crafted heuristics, such as regular expressions and domain-specific keyword lists, to identify TTPs from unstructured reports. For instance, Husari et al. [19] developed TTPDrill, which employed syntactic patterns and ontology mapping to align extracted behaviors with the MITRE ATT&CK framework. While these approaches achieve high precision and strong interpretability, they are costly to maintain and struggle to accommodate linguistic diversity and novel threats. To improve scalability, researchers later turned to supervised machine learning, leveraging features such as n-grams, contextual cues, and positional information to train classifiers. For example, Sharma et al. [20] introduced RADAR, an extensible TTP-based system that applies machine learning to network traffic analysis and malware detection, demonstrating the practical utility of TTP-aware models in real-world security tasks. These models can automatically learn patterns from data and adapt more flexibly to evolving threats; however, they tend to be less interpretable and remain highly dependent on data quality. The limitations of both rule-based and traditional machine learning approaches ultimately motivated the transition toward deep learning and semantic modeling as more robust solutions for TTP classification.

2.2 Applications of Deep Learning for TTPs Classification Tasks

To overcome the limitations of rule-based methods, recent studies have increasingly adopted deep learning approaches, particularly Transformer-based models [10]. BERT [11], with its strong capability for contextual semantic modeling, has been widely applied to extract TTPs from CTI reports. For instance, You et al. [21] proposed the TIM framework, which formulates TTP classification as a sentence-level task and integrates TCENet to capture contextual semantics, thereby enhancing classification performance. Nonetheless, the effectiveness of such models remains constrained by data quality and the comprehensiveness of semantic representations. To address class imbalance, Kim et al. [22] employed Easy Data Augmentation (EDA), which generates synthetic samples through operations such as synonym replacement, random insertion, word swapping, and deletion. While EDA increases diversity in low-resource categories, its surface-level transformations often introduce noise and fail to capture deeper semantic structures, thereby limiting its impact on classification accuracy. To improve domain relevance, researchers have developed specialized models such as SecureBERT [23], trained on cybersecurity corpora, and SecBERT [24], fine-tuned on datasets including APTnotes [25], Stucco-Data [26], CASIE [27], and SemEval [28]. These domain-

adapted models demonstrate superior performance compared to general-purpose language models.

With the advent of LLMs such as GPT-4 [29] and LLaMA-3 [30], deep learning has achieved significant advances in natural language understanding and generation. LLMs are capable of performing complex reasoning and knowledge transfer with minimal reliance on labeled data. In the context of TTP classification, LLMs can emulate the role of a security analyst by extracting and categorizing relevant TTPs from documents without the need for manual rule engineering or extensive annotation. The long-context decoder architecture of LLaMA-3 is particularly well-suited for cross-paragraph reasoning and semantic integration in CTI reports. Nevertheless, LLMs remain susceptible to hallucination [31], producing outputs that appear plausible but are factually inaccurate in the absence of explicit knowledge support. They also suffer from delayed knowledge updates, as their knowledge is fixed at the time of training. Moreover, domain-agnostic models may misinterpret cybersecurity-specific terminology and syntax, thereby reducing classification accuracy.

In addition to leveraging BERT for feature extraction, another line of research focuses on directly fine-tuning LLMs to internalize cybersecurity domain knowledge. For example, Fengrui et al. [32] proposed a TTP classification framework that combines few-shot learning with instruction-tuning strategies. Their method employed ChatGPT to generate diverse TTP descriptions for augmenting rare classes and subsequently fine-tuned Llama-2-7B for task-specific classification of MITRE ATT&CK techniques. While this approach relies solely on LLM-based classification without an external retrieval module, it provides valuable insights into TTP automation through data augmentation and fine-tuning. Nonetheless, such methods face notable challenges: (1) high computational costs, as fine-tuning multi-billion-parameter models requires substantial GPU resources, and (2) delayed knowledge updates, since retraining is necessary to incorporate new threat intelligence.

Table 1: TTPs Classification Research Method Capability Comparison Table.

Study	Domain-Specific Semantic Adaptability	Retrieval Capability	Long-Context Reasoning Ability	Unstructured Data Processing Ability	Classification Task Interpretability
Husari et al. [19]	✓	✗	✗	✓	✗
Sharma et al. [20]	✓	✗	✗	✓	✗
You et al. [21]	✓	✗	✗	✓	✗
Kim et al. [22]	✓	✗	✗	✗	✗
Aghaci et al. [23]	✓	✗	✗	✓	✗
Kun et al. [24]	✓	✗	✗	✓	✗
Fengrui et al. [32]	✓	✗	✓	✓	✓
Ours	✓	✓	✓	✓	✓

In contrast, the enhanced RAG framework proposed in this study fine-tunes only a lightweight BERT to improve retrieval accuracy, significantly reducing computational costs. The RAG architecture further enables immediate knowledge updates by incorporating new CTI documents into the vector database and offers traceable evidence for classification decisions, thereby mitigating hallucination risk. Table 1 compares common TTP classification methods with the proposed approach across dimensions such as semantic adaptability, retrieval capability, long-context reasoning, ability to process unstructured data, and interpretability, underscoring the comprehensive advantages of our framework.

3. Problem Statement

Recent advances in NLP have facilitated the adoption of RAG and LLMs in CTI classification tasks. Despite these developments, several challenges remain. General-purpose embeddings often fail to capture the domain-specific semantics of cybersecurity, thereby limiting classification accuracy. RAG architectures further demand embeddings that are both semantically precise and contextually aligned with threat intelligence data. In addition, generative models frequently struggle to produce reliable and interpretable outputs without carefully structured prompt guidance. To address these limitations, this study decomposes the

research problem into three sub-problems, each targeting a critical component of the overall solution.

Sub-problem 1: Development of an Embedding Model

The first challenge is to obtain embeddings that effectively capture the fine-grained semantics of TTPs. General-purpose embeddings often overlook cybersecurity-specific concepts, reducing retrieval and classification accuracy.

- Input: CTI documents
- Output: BERT Embedding
- Objective: Extract high-quality embeddings for TTP classification, maximizing retrieval accuracy and efficiency

Sub-problem 2: Construction of the Retrieval-Augmented Generation Model

The second challenge focuses on integrating a RAG framework to enhance contextual relevance and inference capabilities for contemporary threat scenarios.

- Input: BERT Embedding
- Output: Top three most relevant TTPs techniques retrieved
- Objective: Improve the accuracy of CTI report classification

Sub-problem 3: Integration with Large Language Models

The final challenge involves combining retrieved results with a generative model to strengthen contextual reasoning and improve inference in threat analysis.

- Input: Prompt + TTPs' procedure
- Output: Technique IDs within TTPs
- Objective: Improve the accuracy of TTPs extraction and classification.

The proposed solution establishes a novel application pathway for RAG in the cybersecurity domain and provides a robust technical foundation for its deployment in CTI classification tasks.

4. Proposed Method

To address the challenges outlined in Section 3, this study proposes a model that integrates

fine-tuned BERT embeddings with a RAG framework for extracting TTPs from CTI. Compared with conventional RAG designs, the proposed model demonstrates superior performance. Section 4.1 presents the overall enhanced RAG architecture, while Section 4.2 details the fine-tuning process of the BERT model that underpins the embedding component.

4.1 Enhanced RAG Architecture

In the proposed approach, fine-tuned embeddings derived from a BERT model trained on TTP classification are extracted and used to replace the generic embeddings within the RAG framework, as illustrated in Figure 1. These embeddings are stored in ChromaDB [33], an open-source vector database selected for its scalability and efficient similarity search capabilities, and are employed to retrieve the most relevant TTPs through semantic matching. ChromaDB is specifically designed for vector data management, offering fast retrieval, a simple API, scalability, and stability. It also supports customizable index structures and real-time updates, making it particularly suitable for precise and efficient retrieval in RAG workflows.

When a user submits a query, the system encodes it using the fine-tuned BERT embeddings trained on cybersecurity data, thereby capturing domain-specific terminology and contextual nuances. The resulting query vector is compared against the stored CTI embeddings in ChromaDB using semantic similarity measures, and the most relevant records are retrieved. The top three retrieved sentences are then passed to the retrieval module of the RAG architecture. These retrieved contents are incorporated into the prompt alongside the user's query and subsequently forwarded to the language model. This process is implemented using the LangChain framework [17], which dynamically integrates the retrieved content with the query context to provide the language model with comprehensive semantic information.

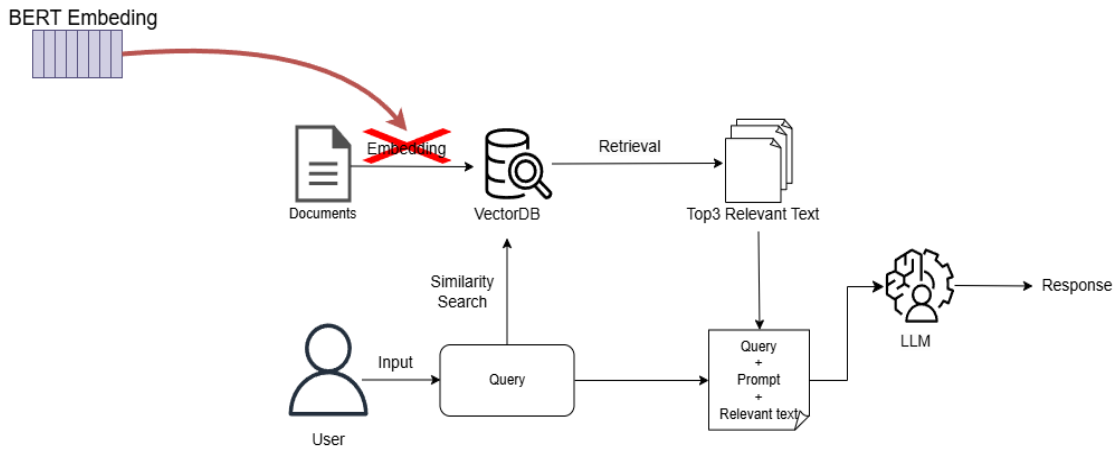

Figure 1: Our architecture of the enhanced RAG framework.

4.2 Fine-tuning BERT Embedding

As an integral component of the architecture introduced in Section 4.1, BERT is fine-tuned to function as the embedding model within the retrieval module. While the pre-trained BERT model exhibits strong semantic understanding, its direct application to specialized downstream tasks such as CTI analysis requires task-specific fine-tuning. This process adapts BERT's general linguistic knowledge to the semantic structures of the cybersecurity domain.

Before fine-tuning, the dataset undergoes standardized preprocessing to convert it into an acceptable input format for the model. Tokenization, a core step in preprocessing, is performed using the WordPiece algorithm. This technique balances vocabulary size and semantic preservation by starting from individual characters and iteratively merging frequently co-occurring subword units. As shown in Figure 2, the term "Drovorub" may be split into "Dor" and "##vor", enabling the model to infer semantics even for previously unseen terminology.

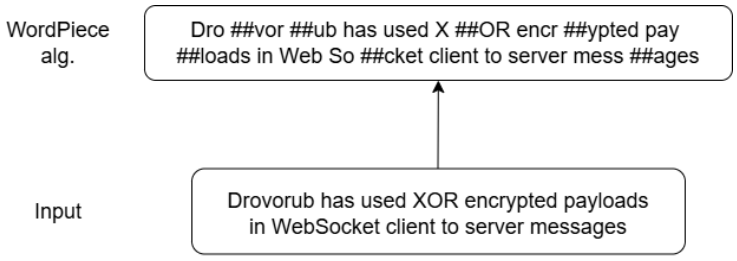

Figure 2: The example of wordpiece methodology.

Within the BERT architecture, multiple Transformer encoder layers progressively capture syntactic and semantic representations through multi-head self-attention and feed-forward neural networks. During fine-tuning, input sentences are tokenized into subwords using the WordPiece algorithm, ensuring robust handling of domain-specific and out-of-vocabulary terms. These tokenized sequences are embedded and propagated through the encoder layers, where bidirectional contextual dependencies are modeled. For the classification task, the hidden state corresponding to the [CLS] token serves as the aggregated sentence representation, which is passed through a fully connected layer and optimized with task-specific objectives. A Softmax layer then outputs normalized probabilities across TTP categories. Through this fine-tuning process, BERT's general linguistic capacity is effectively adapted to the cybersecurity domain, enabling more accurate interpretation of CTI semantics and improving downstream classification performance, as illustrated in Figure 3.

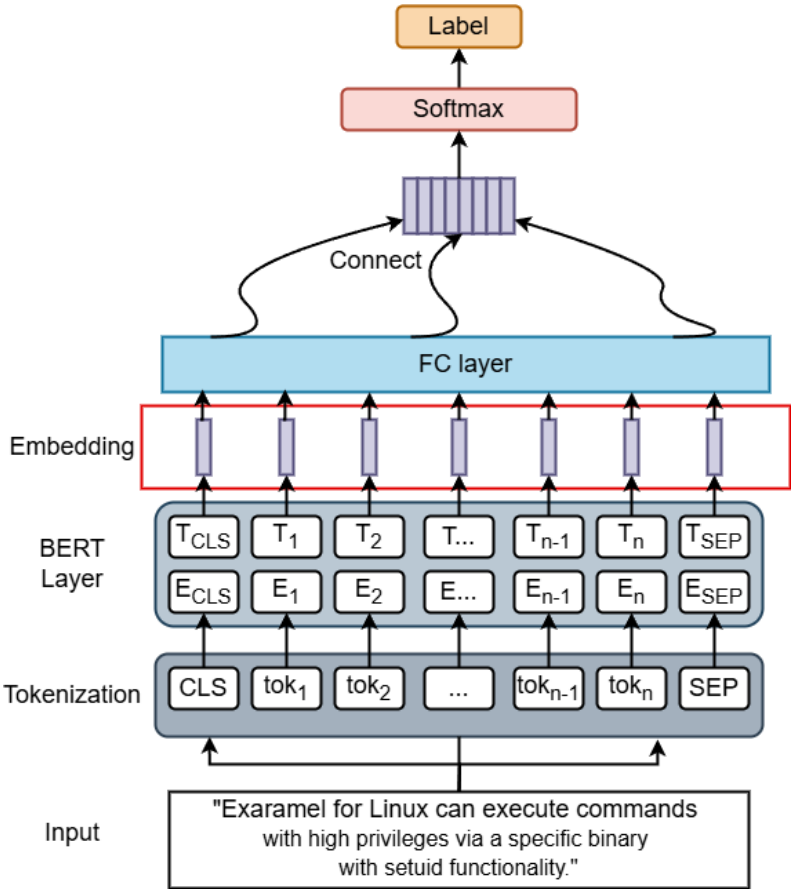

Figure 3: The architecture of the fine-tuned BERT model.

This purposed method introduces three key innovations: (1) adopting TTPs instead of IoCs for more stable and effective CTI classification; (2) leveraging fine-tuned BERT embeddings with RAG to improve semantic understanding and accuracy; and (3) incorporating a decoder to enhance classification performance and provide interpretable reasoning. These innovations form the core of the enhanced RAG threat model, which advances CTI classification by overcoming the limitations of traditional content classification methods, offering higher accuracy and flexibility.

5. Experimental Results

To assess the effectiveness of the proposed method, we conducted experiments covering embedding configurations, retrieval parameters, and generative model performance, using annotated TTPs data as benchmarks. Evaluation primarily employed the F1-score in Sections 5.2, while additional metrics, including Accuracy, Precision, and Recall, were reported in Section 5.3 for a more comprehensive comparison. We also examined key parameters such as Chunk_Size and Overlap_Size, where *Chunk_Size* defines the number of tokens in each text

segment and *Overlap_Size* specifies the portion shared between adjacent segments to preserve semantic continuity.

5.1 Dataset

The experimental dataset in this study is derived from the publicly available MITRE ATT&CK dataset [18] (ATT&CK v14.1, released in October 2023). To construct an embedding model capable of effectively identifying attack behaviors, we employed web scraping techniques to collect a total of 12,006 procedure examples from the database, each describing how a specific threat actor group or malware implements a given attack technique. These descriptions are essential for understanding the semantics of TTPs.

To illustrate the dataset structure and annotation format more clearly, Figure 4 presents an example entry containing a description of a specific attack behavior and its corresponding MITRE ATT&CK Technique ID. The dataset distribution is visualized in Figure 5 as a heatmap, where the horizontal and vertical axes represent different MITRE ATT&CK Tactics. The color intensity indicates the frequency of occurrence of each tactic in the dataset, with darker colors signifying higher frequencies. This visualization helps identify any imbalance in data distribution, guiding training data design and model evaluation strategies.

```
"text": "Transparent Tribe has used dynamic DNS services to set up C2.",
"technique_id": "T1568"
```

Figure 4: Dataset example.

For reproducibility, we performed dataset shuffling with a fixed random seed (set to 42) and split the data into training and test sets with a 9:1 ratio. This ensured that the model was exposed to a diverse range of attack techniques while avoiding overfitting.

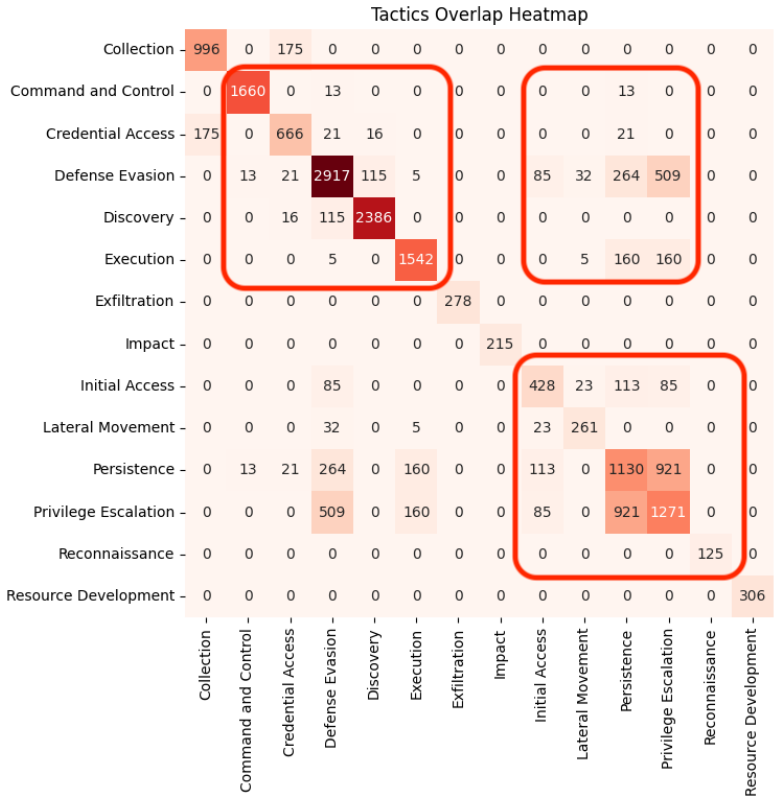

Figure 5: Dataset distribution map.

5.2 Experimental Setup

All experiments were conducted on an Ubuntu server equipped with an 11th Gen Intel® Core™ i7-11800H CPU (2.3 – 4.6 GHz, 8 cores), 16 GB RAM, and an RTX A5000 GPU with 24 GB VRAM (8,192 CUDA cores). We tested multiple configurations of the RAG architecture, varying both embedding models and LLMs.

For the embedding component, we compared OpenAI [34], all-MiniLM-L6-v2 [35], and BERT [11] to assess their suitability for CTI data. For the LLM component, we selected GPT-4 and LLaMA-3, as they represent the latest generation of large-scale models. Additionally, we experimented with document segmentation parameters to identify the optimal encoding strategy.

The key parameter tested was Chunk_Size, which defines the length of each document segment before embedding and retrieval. We evaluated Chunk_Size values from 100 to 500. As shown in Table 2, the highest F1-score of 0.90 was achieved when Chunk_Size was set to 500,

significantly outperforming other segment lengths. This indicates that longer contexts provide richer semantic information, improving both retrieval and generation accuracy. The Set Count column in Table 2 indicates the number of text segments produced at each setting, while Supports denotes the actual number of samples used for evaluation.

Table 2: Chunk_Size for F1-score test.

Set Count	F1-score	Supports
100	0.58	1,201
200	0.83	1,201
300	0.89	1,201
400	0.89	1,201
500	**0.90**	**1,201**

To further improve semantic continuity and answer accuracy, we considered Overlap_Size, which determines the proportion of overlapping context included in each retrieval query. Following Zhang et al. [36], who found that setting Overlap_Size to 20% of Chunk_Size optimizes coverage and efficiency, we fixed Overlap_Size at 100 tokens when using the optimal Chunk_Size of 500.

5.2.1 BERT Hyperparameter Configuration

Table 3: Hyperparameter for BERT.

Hyperparameter	Value
Optimizer	AdamW
Loss Func.	BCE
Activation Func.	Softmax
Batch Size	256
Epoch	100
Learning rate	1e-4

Table 3 lists the hyperparameters used during BERT fine-tuning. We employed AdamW [37] as the optimizer. The loss function was Binary Cross-Entropy (BCE) [38], suitable for single-label classification of TTP categories. The output layer used a Softmax activation to normalize class probabilities within the [0,1] range.

We set the batch size to 256 to balance memory efficiency with stable gradient estimation, and trained for 100 epochs to ensure sufficient learning without overfitting. The learning rate

was set to 1e-4, a common initial value for BERT fine-tuning, balancing convergence speed and stability. Figure 6 shows the F1-score performance of the BERT model after fine-tuning.

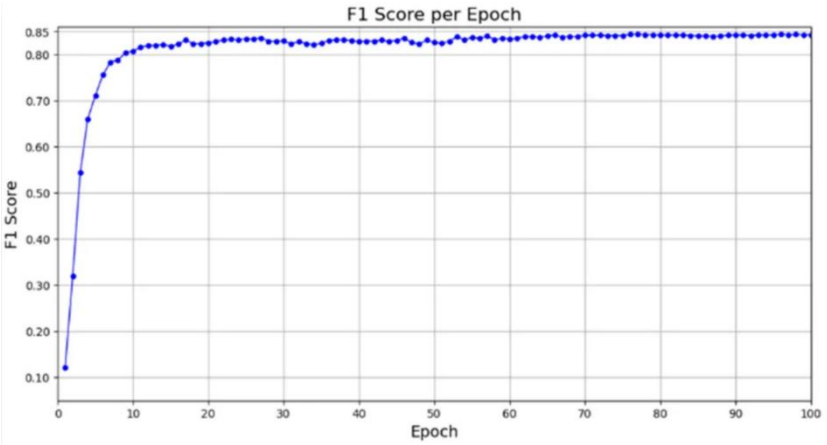

Figure 6: F1-score performance of the BERT model fine-tuning.

5.2.2 Comparison of BERT, SecureBERT, and SecBERT

We fine-tuned BERT, SecureBERT, and SecBERT to evaluate them as embedding layers within the RAG framework. As shown in Figure 7 and Table 4, BERT and SecureBERT both achieved an F1-score of **0.90**, while SecBERT obtained **0.87,** trailing by only 0.03 and still within an acceptable range. Given the minimal differences, all three models are viable in practice. We ultimately selected BERT for our RAG architecture due to its broader semantic adaptability, which better supports diverse CTI classification needs.

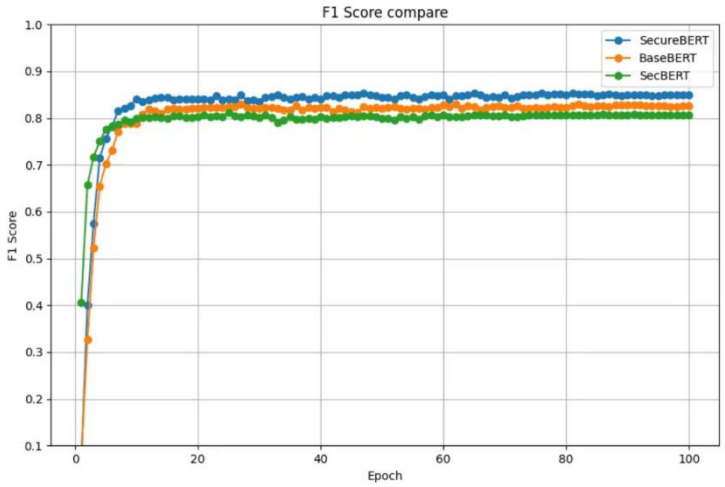

Figure 7: BERT, SecureBERT, and SecBERT training comparison.

Table 4: Comparison of BERT, SecureBERT, and SecBERT.

Model	F1-score	Supports
BERT	**0.90**	**1,201**
SecureBERT	0.90	1,201
SecBERT	0.87	1,201

5.3 Performance Evaluation

When a user submits an unannotated CTI sample, the proposed enhanced RAG model classifies it into one of the 197 Technique IDs defined in the MITRE ATT&CK framework. Model performance was evaluated using standard metrics, including Accuracy, Precision, Recall, and F1-score. To further validate effectiveness, we conducted comparative experiments across different embedding models and LLM configurations.

- **BERT** — Fine-tuned BERT embeddings + linear classifier (no LLM decoder);
- **OpenAI Embedding + GPT-4** — OpenAI embeddings with GPT-4 (March 2024);
- **OpenAI Embedding + LLaMA-3** — OpenAI embeddings with LLaMA-3 decoder;
- **all-MiniLM-L6-v2 + LLaMA-3** — Popular open-source embedding model with LLaMA-3;
- **Proposed BERT Embedding + LLaMA-3** — Our fine-tuned BERT embeddings with LLaMA-3 decoder.

In this study, the latest commercial version of GPT-4 was not included in the final evaluation due to data security concerns. According to OpenAI's policies, API input data may be retained in system logs, posing potential risks of exposing sensitive CTI that contains enterprise defense strategies or attack records. To comply with security requirements, we prioritized the use of locally hosted LLMs for our experiments.

According to the results presented in Figure 8 and Table 5, our proposed enhanced RAG architecture outperforms all comparative models, except for the fine-tuned BERT baseline, across all evaluation metrics, demonstrating particularly stable and superior accuracy in terms of the F1-score. Specifically, the fine-tuned BERT baseline achieved an F1-score of **0.94**, while RAG models using OpenAI embeddings with GPT-4, OpenAI embeddings with LLaMA-3, and all-MiniLM-L6-v2 embeddings with LLaMA-3 achieved **0.54**, **0.38**, and **0.37**, respectively. In contrast, our enhanced RAG model reached an F1-score of **0.93**, confirming its effectiveness and robustness in CTI classification tasks. This indicates that the fine-tuned BERT embeddings can more effectively capture the terminology and patterns within the context of cyberattacks.

When combined with the language model decoder, this capability further enhances the model's ability to capture semantic nuances and improve classification decisions

It is worth noting that, compared to the fine-tuned BERT baseline model, the proposed enhanced RAG architecture achieves comparable performance across all metrics, demonstrating its ability to match the classification effectiveness of the baseline. This finding suggests that, in the context of a purely classification-oriented task, a fine-tuned BERT can directly determine the category of an input sentence without the need for retrieval. While RAG integrates the advantages of retrieval and generation, its final decision heavily depends on the quality of the retrieved results. When the BERT embeddings yield correct classifications, RAG can retrieve highly relevant information, thereby maintaining stable accuracy and recall. However, when BERT embeddings misclassify an instance, the retrieved content may deviate from the correct context, potentially misleading the LLM's generative judgment and, in certain cases, resulting in slightly lower performance compared to the purely fine-tuned BERT model.

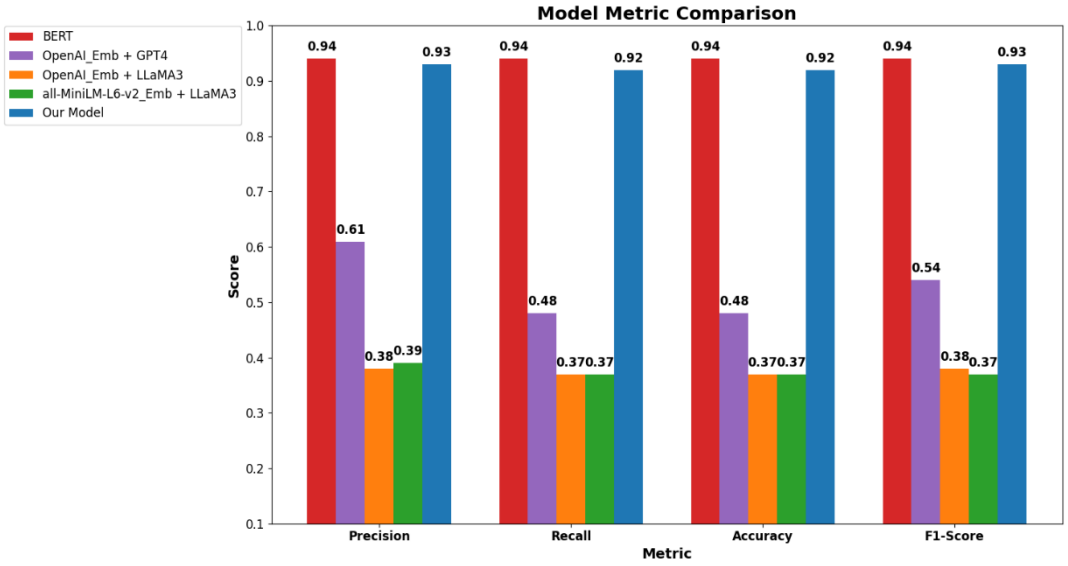

Figure 8: Models Metric Compare.

Table 5: Models Evaluation Comparison.

Sets	Precision	Recall	Accuracy	F1-score
BERT	0.94	0.94	0.94	0.94
OpenAI Embedding + GPT-4	0.61	0.48	0.48	0.54
OpenAI Embedding + LLaMA-3	0.38	0.37	0.37	0.38
all-MiniLM-L6-v2 Embedding + LLaMA-3	0.39	0.37	0.37	0.37
Ours	**0.93**	**0.92**	**0.92**	**0.93**

We also implemented a user-friendly interface to assist analysts in querying and classifying TTPs from CTI, as shown in Figure 9. Users can input descriptive threat information, and the system retrieves and ranks relevant Technique IDs along with their ATT&CK classification and technical summaries. This not only improves analysis efficiency but also lowers the learning curve for newcomers to the ATT&CK framework.

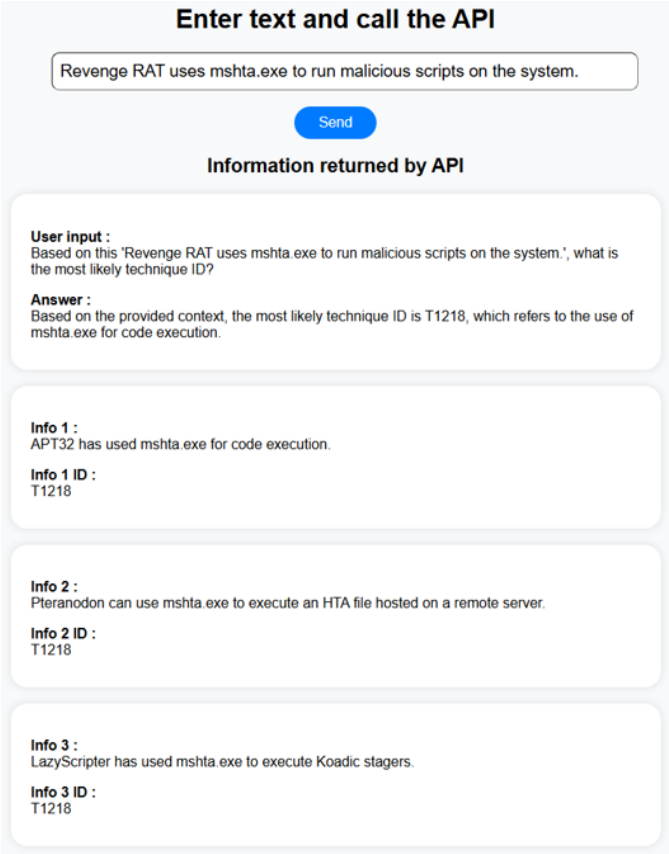

Figure 9: Example of the program we developed.

6. Conclusion

This study proposes an enhanced RAG model tailored for CTI classification tasks. Experimental results demonstrate that conventional RAG embedding models, largely trained on general-purpose datasets, struggle to capture the specialized semantics of cybersecurity. By leveraging data from the MITRE ATT&CK framework, we trained a domain-specific embedding model and integrated it into the RAG architecture. Our approach significantly outperformed RAG models using OpenAI and all-MiniLM-L6-v2 embeddings, with traditional models achieving F1-scores between 37% and 54%, compared to 93% for our proposed model. This improvement highlights the necessity of domain-optimized embeddings for effective CTI

classification. Moreover, incorporating an LLM enhanced the contextual understanding and interpretability of classification results, offering more reliable decision support. In summary, the proposed method provides a robust and scalable solution for advancing automated CTI analysis.

References

[1] B. Shin and P. B. Lowry. "A review and theoretical explanation of the 'Cyberthreat-Intelligence (CTI) capability' that needs to be fostered in information security practitioners and how this can be accomplished," In: *Computers & Security 92* (2020), p. 101761.

[2] H. Kure, and S. Islam. "Cyber threat intelligence for improving cyber security and risk management in critical infrastructure," In: *Journal of Universal Computer Science* 25.11 (2019), pp. 1478–1502.

[3] W. Xiong, E. Legrand, O. Åberg, and R. Lagerström. "Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix," In: *Software and Systems Modeling* 21.1 (2022), pp. 157–177.

[4] B. Cinar. "A study on cyber threat intelligence based on current trends and future perspectives," In: *Advances and Challenges in Science and Technology* (2023), p. 37.

[5] M. S. Abu, S. Ra. Selamat, A. Ariffin, and R. Yusof. "Cyber threat intelligence–issue and challenges," In: *Indonesian Journal of Electrical Engineering and Computer Science* 10.1 (2018), pp. 371–379.

[6] AlienVault. AlienVault - Open Threat Exchange. [Accessed 03-01-2024]. 2024. url: https://otx.alienvault.com.

[7] L. M. Kristiansen, V. Agarwal, K, Franke, and R. S. Shah. "Cti-Twitter: Gathering cyber threat intelligence from twitter using integrated supervised and unsupervised learning," In: *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 2299–2308.

[8] M. Büchel, T. Paladini, S. Longari, M. Carminati, S. Zanero, H. Binyamini, G. Engelberg, D. Klein, G. Guizzardi, M. Caselli, et al. "SoK: Automated TTP Extraction from CTI reports-Are we there yet?," (2025).

[9] J. Ye, X. Chen, N. Xu, C. Zu, Z. Shao, S. Liu, Y. Cui, Z. Zhou, C. Gong, Y. Shen, et al. "A comprehensive capability analysis of gpt-3 and gpt-3.5 series models," In: *arXiv preprint* arXiv:2303.10420 (2023).

[10] A. Vaswani. "Attention is all you need," In: *arXiv preprint* arXiv:1706.03762 (2017).

[11] J. Devlin. "Bert: Pre-training of deep bidirectional transformers for language un derstanding," In: *arXiv preprint* arXiv:1810.04805 (2018).

[12] ChatGPT — chatgpt.com. https://chatgpt.com/. [Accessed 15-06-2025].

[13] BookCorpus- Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/ BookCorpus. [Accessed 17-06-2025].

[14] Wikipedia, the free encyclopedia — wikipedia.org. https://www.wikipedia.org/. [Accessed 17-06-2025].

[15] L. Floridi and M. Chiriatti. "GPT-3: Its nature, scope, limits, and conse quences," In: *Minds and Machines 30* (2020), pp. 681–694.

[16] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks," In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 9459–9474.

[17] LangChain — langchain.com. https://www.langchain.com/. [Accessed 15-06-2025].

[18] MITRE ATT&CK® — attack.mitre.org. https://attack.mitre.org/. [Accessed 15-06-2025].

[19] G. Husari, E. Al-Shaer, M. Ahmed, B. Chu, and X. Niu. "Ttpdrill : Automatic and accurate extraction of threat actions from unstructured text of cti sources," In: *Proceedings of the 33rd annual computer security applications conference*. 2017, pp. 103–115.

[20] Y. Sharma, S. Birnbach, and I. Martinovic. "Radar : A TTP-based extensible, explainable, and effective system for network traffic analysis and malware detection," In: *Proceedings of the 2023 European Interdisciplinary Cyberse-curity Conference*. 2023, pp. 159–166.

[21] Y. You, J. Jiang, Z. Jiang, P. Yang, B. Liu, H. Feng, X. Wang, and N. Li. "TIM : Threat context-enhanced TTP intelligence mining on un structured threat data," In: *Cybersecurity 5.1* (2022), p. 3.

[22] H. Kim and H. Kim. "Comparative experiment on TTP classification with class imbalance using oversampling from CTI dataset," In: *Security and Com munication Networks 2022.1* (2022), p. 5021125.

[23] E. Aghaei, X. Niu, W. Shadid, and E. Al-Shaer. "Securebert : A domain specific language model for cybersecurity," In: *International Conference on Security and Privacy in Communication Systems*. Springer. 2022, pp. 39–56.

[24] GitHub - jackaduma/SecBERT: pretrained BERT model for cyber security text, learned CyberSecurity Knowledge — github.com. https://github.com/jackaduma/SecBERT. [Accessed 05-08-2025].

[25] GitHub - kbandla/APTnotes: Various public documents, whitepapers and articles about APT campaigns — github.com. https://github.com/kbandla/APTnotes. [Accessed 05-08-2025].

[26] J. Goodall. Stucco-Data — stucco.github.io. https://stucco.github.io/data/. [Accessed 05-08-2025].

[27] GitHub - Ebiquity/CASIE : CyberAttack Sensing and Information Extraction — github.com. https://github.com/Ebiquity/CASIE. [Accessed 05-08-2025].

[28] CodaLab-Competition competitions.codalab.org.https://competitions.codalab.org/competitions/17262. [Accessed 05-08-2025].

[29] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, FL. Aleman,D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila. Gpt-4 technical report. *arXiv preprint* arXiv:2303.08774. 2023 Mar 15.

[30] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, A. Goyal. The Llama 3 herd of models. *arXiv e-prints*. 2024 Jul:arXiv-2407.

[31] V. Rawte, A. Sheth, and A. Das. "A survey of hallucination in large foundation models," In: *arXiv preprint* arXiv:2309.05922 (2023).

[32] Y. Fengrui and Y. Du. "Few-Shot learning of TTPs classification using large language models," (2024).

[33] Chroma — trychroma.com. https://www.trychroma.com/. [Accessed 20-06-2025].

[34] OpenAI— openai.com. https://openai.com/, [Accessed 30-08-2025]

[35] sentence-transformers/all-MiniLM-L6-v2·HuggingFace--- huggingface.co. https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2, [Accessed 30-08-2025]

[36] R. Zhang, H. Du, Y. Liu, D. Niyato, J. Kang, S. Sun, X. Shen, and H. V. Poor. "Interactive AI with retrieval-augmented generation for next generation networking," In: *IEEE Network* (2024).

[37] I. Loshchilov, F. Hutter, et al. "Fixing weight decay regularization in adam," In: *arXiv preprint* arXiv:1711.05101 5.5 (2017), p. 5.

[38] U. Ruby, V. Yendapalli, et al. "Binary cross entropy with deep learning technique for image classification," In: *Int. J. Adv. Trends Comput*. Sci. Eng 9.10 (2020).