Procesamiento del Lenguaje Natural, Revista nº 75, septiembre de 2025, pp. 199-210 recibido 05-04-2025 revisado 09-05-2025 aceptado 19-05-2025

Improving the Classification of Cybersecurity Attack Procedures using Retrieval Augmented Generation

Clasificación de Procedimientos de Ataques de Ciberseguridad mediante Generación Aumentada por Recuperación

Sonia Bilbao-Arechabala,[1] **Aitziber Atutxa,**[2] **Javier Del Ser**[1,3]

[1]TECNALIA, Basque Research & Technology Alliance (BRTA), 48160 Derio, Spain
[2]Department of Languages and Computer Systems
University of the Basque Country (UPV/EHU), 48013 Bilbao, Spain

[3]Department of Mathematics
University of the Basque Country (UPV/EHU), 48940 Leioa, Spain
{sonia.bilbao, javier.delser}@tecnalia.com, aitziber.atucha@ehu.eus

Abstract: Understanding the tactics (*why*), techniques (*how*) and procedures (*methods*) behind a cybersecurity attack is paramount to develop defenses against them or to mitigate their effects. However, this task requires a high-level of technical expertise, is time-consuming and error prone. In this work we verify that open-source Llama 3.1 LLMs (Large Language Models) cannot automatically identify which of the 625 MITRE techniques is used within a cybersecurity attack procedure. We evaluate two RAG (Retrieval Augmented Generation) approaches to enhance the classification accuracy. Our experiments show the importance of the embedding model in information retrieval. Moreover, our analysis shows that selecting appropriate examples helps the language model reduce ambiguity. Specifically, a dynamic few-shot learning strategy performs best for larger models, whereas a multiple-choice strategy is more appropriate for smaller models. In contrast, corrective RAG techniques fail to provide significant enhancements, highlighting current methodological limitations and the inherent complexity of this task.
Keywords: Cyber-security, RAG, open-source LLM, text embedding.

Resumen: Comprender las tácticas (*por qué*), técnicas (*cómo*) y procedimientos (*métodos*) de un ciberataque es clave para establecer defensas o mitigar sus efectos. Sin embargo, esta tarea requiere conocimientos técnicos avanzados, es tediosa y propensa a errores. En este trabajo verificamos que los grandes modelos de lenguaje de código abierto (LLMs), como Llama 3.1, no son capaces de identificar automáticamente cuál de las 625 técnicas MITRE se emplea en un ataque. Evaluamos dos enfoques RAG (Generación Aumentada por Recuperación) para mejorar la clasificación automática de procedimientos. Nuestros experimentos destacan la importancia de los *embeddings* para la recuperación de información. Además, mostramos que una buena selección de ejemplos reduce la ambigüedad: el few-shot learning es más eficaz en modelos grandes, mientras que la opción de respuesta múltiple resulta más adecuada para modelos pequeños. Las técnicas RAG correctivas no aportaron mejoras significativas, reflejando la dificultad técnica inherente a este problema.
Palabras clave: Ciberseguridad, RAG, LLM abiertos, embeddings de texto.

1 Introduction

Digital transformation in enterprises is increasing the potential of cyber threats and hence, the necessity of cyber security defense. It is critical for enterprises to understand, detect, prevent and react to attacks or cyber threats. In this regard, the Cybersecurity and Infrastructure Security Agency (CISA) works to defend against today's threats by, among others, publishing Cyber Threat Intelligence (CTI) reports[1], which provide insights into current and emerging

[1]CTI Reports, `www.cisa.gov/news-events/cybersecurity-advisories` [April 4th, 2025].

ISSN 1135-5948 DOI 10.26342/2025-75-15
©2025 Sociedad Española para el Procesamiento del Lenguaje Natural

cyber threats to an organization's security.

In addition, MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) is a comprehensive framework of adversary tactics, techniques, and procedures based on real-world observations (Al-Sada, Sadighian, and Oligeri, 2024). It is used globally by cybersecurity professionals to understand and defend against cyber threats. Tactics represent the adversary's goals or the 'why' behind an attack. Techniques describe 'how' adversaries achieve their goals. Techniques are grouped in two levels of detail with parent techniques and sub-techniques offering more detailed descriptions of specific actions within a technique. Finally, procedures describe the specific implementations or methods, adversaries use to carry out techniques and sub-techniques (see Figure 1). Moreover, MITRE ATT&CK is organised into different matrices for various environments, such as enterprise networks[2], mobile devices, and industrial control systems (ICS).

Tactical Threat Intelligence, i.e., comprehending the Tactics, Techniques, and Procedures (TTPs) used by threat actors is of paramount importance for security teams to understand cybersecurity attacks, to develop defenses against them, or to mitigate their effects. However, this task requires a high level of technical expertise, being both time-consuming and error prone. It is worth noting that the enterprise matrix in the MITRE ATT&CK framework alone outlines 14 tactics, 202 parent techniques, 423 active sub-techniques, and 12,999 documented examples of attack procedures.

In this paper we experimentally evaluate the potential of Large Language Models (LLMs) and Retrieval Augmented Generation (RAG) methodologies to tackle the challenge of attack procedure classification. Our experiments are designed to provide evidence to the following research questions (RQ):

- **RQ1:** Can open-source LLMs effectively perform the task under zero-shot (off-the-shelf) or few-shot learning regimes?
- **RQ2:** Does the implementation of a RAG approach enhance the overall classification accuracy? If so, which embeddings are best for information retrieval?
- **RQ3:** Can corrective RAG techniques further improve performance?

Our overarching goal is to provide insights into the efficacy of these advanced models for the classification of attack procedures. In doing so, our main contributions include the development of an annotated dataset comprising 532 multiple-choice questions specifically designed for attack procedure classification, as well as the implementation of a RAG-based system utilizing customized prompts tailored to this task.

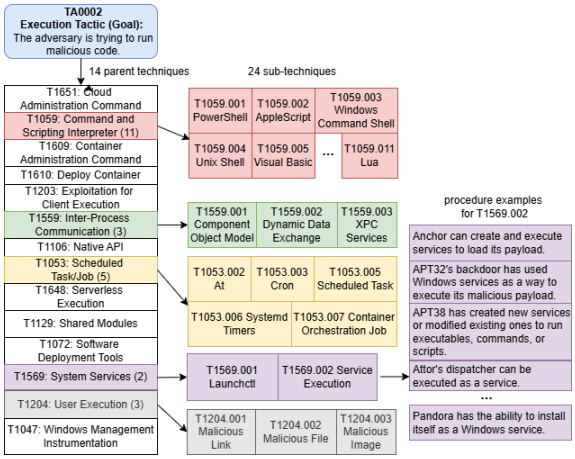


Figure 1: Example of a tactic, its techniques and some procedures. Execution tactic with 14 parent techniques and 24 sub-techniques.

The manuscript is structured as follows. Section 2 first revisits the related literature to put in context the contribution of the paper, whereas Section 3 details the dataset, models, embeddings, RAG frameworks and metrics utilized in our study. Section 4 presents and discusses the obtained results for each RQ. Finally, Section 5 concludes the paper with a discussion of the main findings and prospects for future research.

2 *Related Work*

As highlighted previously, a fundamental challenge in cybersecurity is the accurate prediction of the MITRE ATT&CK technique given a description of an attack procedure. Timely attack identification relies heavily on the vast amount of security information that exists in unstructured textual form (e.g., blogs, social media, Telegram messages, threat intelligence reports). Consequently, several approaches have been documented in the literature for training models and developing NLP (Natural Language Pro-

[2]MITRE matrix for enterprise networks, `attack.mitre.org/matrices/enterprise/` [April 4th, 2025].

cessing) and LLM-based techniques to classify attackers' TTPs from unstructured text. First approaches like Sauerwein and Pfohl (2022) proposed a combination of NLP traditional techniques like tokenization, POS tagging, IoC replacement, lemmatization, one-hot encoding, binary relevance and Support Vector Machine for performing TTP classification. These techniques worked well for classifying tactics, but they were not able to classify techniques properly.

Since the emergence of the Transformer architecture (Vaswani et al., 2017), language models have evolved from encoder-only models (BERT, RoBERTa), to encoder-decoder models (T5, BART) and finally to decoder-only models (GPT, Llama, Qwen). Encoder-based architectures are designed to understand and process the entire input sequence, making them particularly effective for tasks that demand a thorough understanding of context, like text classification. This is the reason why the task of tactic and technique classification has been addressed in the past using fine-tuned BERT-based models (Devlin et al., 2019) models. Alves, Filho, and Gonçalves (2022) tackled the problem by training encoder models, specifically 11 different BERT family models, and fine-tuning their hyperparameters. This time, the scope was reduced to the 253 most common techniques and sub-techniques achieving 82.64% accuracy on the test dataset (which contained attack procedures from MITRE) with the RoBERTa Large model and 78.75% on the inference dataset (which contained manually extracted procedures from CTI reports) with the BERT Large Cased model. The Threat Report ATT&CK Mapper (TRAM) open-source platform developed by MITRE uses a fine-tuned SciBERT model (Beltagy, Lo, and Cohan, 2019) to identify up to 50 ATT&CK techniques. Li, Huang, and Chen (2024) classify, at sentence level and using the DistilBERT model, procedures that were extracted from TRAM training dataset and from the MITRE ATT&CK website. When classifying according to the tactic used in the procedure, the model achieved an accuracy of 81% whereas, when identifying the MITRE technique, the accuracy was 67%.

Since cybersecurity information is inherently dynamic as new attacks emerge continuously, a significant limitation of BERT-based solutions is their reliance on existing training data, restricting their ability to predict new attack techniques. BERT models require supervised training, which is time-consuming. Furthermore, such models by themselves cannot predict unseen techniques. To incorporate new techniques, these models must undergo retraining. Hence, current approaches try to evolve encoder models to prompt-based interactions with decoder-only based models.

Decoder-only based models are autoregressive models that generate text by predicting the next token based on the current and previous tokens. The size of these models has increased steadily over the years, reaching billions of parameters. Their capabilities have grown as well, further trained to be able to follow instructions and carry out tasks without the need for further fine-tuning (Dong and others, 2024). Within this framework, *in-context learning* refers to the ability of LLMs to follow instructions or prompts and perform tasks under the guidance of examples (i.e., few-shot learning) or by providing a suitable augmented context for the task at hand. These huge models are trained on extensive corpora of textual data, often by big players such as OpenAI, Anthropic, Google, Meta or Microsoft. However, the specifics of the training datasets are often undisclosed. Therefore, it is crucial to assess the domain-specific knowledge captured by these models, particularly in specialized fields like cybersecurity. In this context, Fayyazi and Yang (2023) analysed and compared the direct use of LLMs (e.g., GPT-3.5) versus supervised fine-tuning of small-scale LLMs (e.g., BERT) to study their potential in predicting ATT&CK tactics. The study is limited as it focuses only on tactic classification achieving a 44% accuracy with GPT-3.5 and 31% with Bard.

Even though LLMs have powerful modeling capabilities, they still suffer from hallucinations, particularly for knowledge-intensive tasks, and require fine-tuning for knowledge updates. In this sense, Retrieval-Augmented Generation (RAG) has demonstrated to be very useful, as it allows retrieving information from external knowledge databases to provide context to the LLM. Thanks to this augmented context, the LLM can extract the answer to a given query (Gao and others, 2024) for general domain tasks. The community working in LLM-powered cyber-

security has become increasingly interested in RAG in recent times. For instance, in (Fayyazi, Taghdimi, and Yang, 2024) the authors use RAG for ATT&CK tactics classification and highlight the importance of retrieving relevant context for the LLM. Using only the prompt (i.e., relying on GPT-3.5's pre-trained knowledge) resulted in poor performance in interpreting the ATT&CK tactics with an average F1 score of 0.60 for the 14 tactics. In contrast, when using RAG by providing the exact URL of the procedure to retrieve context, this score significantly improved to 0.95, representing the upperbound of a RAG-based solution. In a more realistic scenario (RAG with top-3 similar procedures, without any URL), the performance although smaller (0.68 average F1 Score) improved 8% with respect to the non-RAG solution.

Despite their potential, RAG systems face significant development challenges and multiple points of failure (Barnett et al., 2024). Regarding classification problems, such critical points are related to the relevance or similarity of the retrieved information to the item being classified (Salemi and Zamani, 2024), namely, a) failure to retrieve the most similar items from the knowledge base; b) relevant items are not ranked top, causing confusion to the LLM when searching for a response; and/or c) missing content, as there are no similar procedures in the knowledge base.

Contribution While previous studies have focused on fine-tuning encoder models and on identifying tactics with commercial LLMs such as GPT3.5, to our knowledge our study is the first examining the capabilities of current open-source LLMs to accurately classify attack procedures according to the attack technique used. To this end, we provide a dataset for attack procedure classification and describe a methodology for evaluating new models proposed to tackle this task. Moreover, we propose a RAG-based approach to improve accuracy, and we explain the source of information retrieval errors to motivate follow-up studies aimed to circumvent them effectively.

3 Resources and Methods

We now proceed by describing the dataset (Section 3.1), LLMs (Section 3.2), embeddings (Section 3.3), RAG frameworks (Section 3.4) and performance metrics (Section 3.5) considered in our experiments.

3.1 TTPEval Dataset

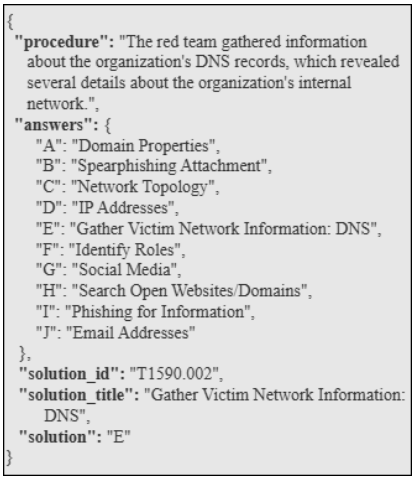
To the best of our knowledge, there is no publicly available dataset to evaluate the performance of LLMs for Tactical Threat Intelligence and attack procedure classification. Available datasets are focused on quantifying LLM security risks and capabilities, e.g., CyberSecEval 3 (Wan and others, 2024). Even though the CyberMetric Dataset (Tihanyi et al., 2024) consists of 10,000 questions, only 196 of them mention techniques, tactics or MITRE. Moreover, such questions are too broad for evaluating the classification task, as they focus on general attack types, such as phishing or vulnerability attacks, without mapping them to the technique level.

We therefore created a dataset of 532 questions and the corresponding responses, coined as the TTPEval dataset[3], from the information available in 30 CISA reports in the period from 20th July 2023 to 11th July 2024. We extracted the attack procedure used and its mapping to a MITRE tactic and technique. The dataset has been constructed to support multiple evaluation settings and usage scenarios. Given a *query* describing a specific attack procedure, the dataset includes the *correct answer* (associated MITRE ATT&CK technique name), enabling the assessment of LLM outputs in a free-text format using manual evaluation or/and automatic metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and BERTScore (Zhang et al., 2020). Additionally, the dataset provides a set of *candidate answers* for each query, facilitating evaluation in a multiple-choice question answering (MCQA) setting. To do so, given the correct answer technique we randomly selected other 4 to 9 techniques and sub-techniques grouped under the same tactic. Finally, we shuffled the alternatives to guarantee that the correct response is free from positional biases. Figure 2 shows a JSON sample of the 10 possible options of technique names created for a procedure. An additional dataset was also constructed using technique identifiers rather than technique names (Figure 3).

3.2 Open-source LLMs

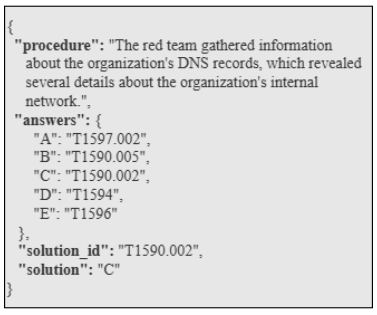
Our framework uses the Llama 3.1 open-source models (Grattafiori and others, 2024), with experiments comparing the 8B and 70B

[3]https://github.com/soniabilbao/TTPEval-dataset



```
{
 "procedure": "The red team gathered information
   about the organization's DNS records, which revealed
   several details about the organization's internal
   network.",
 "answers": {
    "A": "Domain Properties",
    "B": "Spearphishing Attachment",
    "C": "Network Topology",
    "D": "IP Addresses",
    "E": "Gather Victim Network Information: DNS",
    "F": "Identify Roles",
    "G": "Social Media",
    "H": "Search Open Websites/Domains",
    "I": "Phishing for Information",
    "J": "Email Addresses"
 },
 "solution_id": "T1590.002",
 "solution_title": "Gather Victim Network Information:
     DNS",
 "solution": "E"
}
```


Figure 2: Answers with technique *names*.



```
{
 "procedure": "The red team gathered information
   about the organization's DNS records, which revealed
   several details about the organization's internal
   network.",
 "answers": {
    "A": "T1597.002",
    "B": "T1590.005",
    "C": "T1590.002",
    "D": "T1594",
    "E": "T1596"
 },
 "solution_id": "T1590.002",
 "solution": "C"
}
```


Figure 3: Answers with technique *identifiers*.

parameter versions. Open-source LLMs offer several advantages for enterprise use over commercial models. They ensure data privacy and security by keeping input data under organizational control, eliminate licensing costs, and provide transparency and flexibility, allowing for architectural modifications and optimization for specific needs.

3.3 Embeddings

The embedding model is essential for accurately capturing the semantics of each procedure: low-quality embeddings result in poor retrieval. Embedding models come in three main types: sparse, dense, and multi-vector. Sparse embeddings like SPLADE (Formal et al., 2021; Lassance et al., 2024) are high-dimensional with few non-zero values, highlighting only relevant information. They are effective for rare words or specialized terms. Dense embeddings, such as sentence transformers (Reimers and Gurevych, 2019), are lower-dimensional but information-rich, capturing overall semantic meanings in a single vector. Multi-vector embeddings, like ColBERT (Santhanam et al., 2022), perform query-document interaction after independent encoding, enabling fine-grained matching. BGE-M3 (Chen et al., 2024) is a versatile model that supports sparse, dense, and multi-vector retrieval by assigning relevance scores to each. Our study considers three embedding models: 1) dense embeddings with the BGE-M3 model; 2) ColBERT multi-vector embeddings; and 3) ATT&CK BERT[4], a cybersecurity-specific dense model based on sentence transformers.

3.4 RAG and Corrective RAG

We develop the attack procedure classification system using two methods: RAG and Corrective RAG (CRAG). RAG enhances generation by incorporating external knowledge sources (Gao and others, 2024), but its effectiveness relies on the relevance of retrieved documents. CRAG addresses this by assessing the confidence and quality of retrieved content, improving robustness (Yan et al., 2024). Web searches can further enrich retrieval, but in our experiments CRAG is limited to evaluating document relevance, without web search, to assess the ability of LLMs based solely on their internal knowledge.

3.5 Metrics

As noted in the introduction, the 625 techniques in the MITRE ATT&CK Enterprise Matrix are organized hierarchically into 202 parent techniques and 423 sub-techniques. Using standard classification metrics such as recall and F1-score, which assume a flat label space, would penalize the model unnecessarily, leading to an artificial decrease in recall or F1, even if the model's predictions are semantically close to the true label. Hence, we propose these two metrics to manually evaluate the accuracy of LLM-generated free-text responses: *Exact Match Accuracy (EMA)* and *Partial Match Accuracy (PMA)*. The LLM outputs the name or identifier of a technique, which is then compared to the dataset reference. The ratio of correct answers produces such accuracy metrics.

Given a procedure p_i, with predicted technique t_i and ground truth t_j, we define:

- Exact Match (EM): $t_i = t_j$.
- Child Match (CM): t_i is subtechnique of t_j.
- Parent Match (PM): t_i is a parent of t_j.

[4]ATT&CK BERT, `huggingface.co/basel/ATTACK-BERT` [April 4th, 2025].

• Sibling Match (SM): t_i and t_j are subtechniques of the same parent.

Based on these definitions, EMA considers only EM cases as correct answers. PMA includes all matches, i.e., EM+CM+PM+SM.

4 *Experiments and Results*

We now present the experimental results for each of the RQ posed in the introduction:

4.1 RQ1: Zero-/Few-shot Attack Procedure Classification

Using prompt engineering (see Figure 4), we evaluate the capability of open-source LLMs to answer questions related to MITRE ATT&CK. We compare results using Llama 2 (7B, 70B), and Qwen1.5-72B. All LLMs are aware of what MITRE ATT&CK is, i.e., a knowledge base of cyber attacks and threat techniques. However, they are not able to answer questions such as *what the MITRE identifier is, given the name of a tactic, or technique*, or *what a tactic or technique identifier, such as TA0006, means*. When asked about what MITRE ATT&CK technique is used in a given attack procedure, the LLM is not able to answer, answers incorrectly, or responds with the name of a tactic instead of a technique. The performance is so low that it is not worthwhile to consider them as a viable classification model for identifying techniques.

```
What MITRE Attack technique was used in the
following attack? Attack: CISA's red team used
open source tools and services to probe the
organization's internet-facing presence and
gather information, including names, roles,
and contact information.
Options: A) Spearphishing Link,
B) Determine Physical Locations,
C) Scan Databases,
D) Search Victim-Owned Websites,
E) Spearphishing Voice
Choose the correct answer (A, B, C, D or E)
only. Always return in this format: 'ANSWER:X'
```

Figure 4: Example of a prompt.

With Llama 3.1 8B and 70B, capabilities improve significantly. We conduct experiments (Table 1) using the TTPEval Dataset described in Section 3.1, obtaining an EMA between 21.9% and 25.8% when the LLM has to choose from a list of identifiers (Figure 3), and an EMA between 79.5% and 86.5% when the LLM is provided with the names of techniques (Figure 2). When we query Llama 3.1 70B to classify a procedure extracted from a CISA report without examples, the EMA is 27.63%, while Llama 3.1 8B achieves only 4.89% and, in some cases, responds with a tactic name instead of a technique.

Multiple ID choice (5 options)	
Llama 3.1 8B	24.497%
Llama 3.1 70B	21.938%
Multiple name choice (10 options)	
Llama 3.1 8B	79.525%
Llama 3.1 70B	86.472%
Zero-shot free-text scenario	
Llama 3.1 8B	4.887%
Llama 3.1 70B	27.632%

Table 1: EMA (dataset of 532 procedures).

Although the LLMs cannot consistently classify correctly on their own, accuracy reaches around 80% when selecting from a limited set of options in the MCQA setting. The challenge lies in identifying a small subset of techniques—out of the 625 available—that includes the correct one. To address this, we consider that retrieving relevant context using a RAG system could help the LLM improve classification accuracy.

4.2 RQ2: Embedding Model Selection and RAG Evaluation

We answer this second RQ by evaluating the accuracy of a RAG approach for the procedure classification problem. RAG systems are designed to identify an appropriate context for the language model, enabling it to perform the task accurately. However, developing an effective RAG system requires selecting an appropriate embedding model to ensure that the most relevant information is retrieved and ranked top, thereby avoiding confusion for the model. In the subsequent subsections we describe the process followed for this purpose.

4.2.1 Selection of the embedding model for information retrieval

The goal of this step is to ensure the retrieval system prioritizes procedures from our knowledge base that use the same attack technique as the one being classified. Our vector database stores embeddings for 12,999 attack procedures from the MITRE ATT&CK Matrix for Enterprise (Footnote 2). We must therefore choose an embedding model that accurately captures each procedure's meaning. The hypothesis is that procedures with similar meaning share the same technique. Thus, cosine distance should be

closer to 0 for procedures using the same technique, ensuring they rank highest in retrieval.

Using the TTPEval Dataset, we evaluate the three embedding models described in Section 3.3: BGE M3 dense embeddings, ColBERT multi-vector embeddings, and ATT&CK BERT embeddings. For each dataset sample, we retrieve the 100 most similar procedures and record the position (top-k) of the first procedure using the same attack technique as the query.

Model	NF (%)	Top 1 (%)	Top 2 (%)	Top 10 (%)	Top 20 (%)
BGE-M3 Dense	20.86	28.20	36.09	55.26	66.17
ColBERT	18.05	31.20	38.53	57.33	66.73
ATT&CK BERT	20.30	34.21	42.67	64.29	70.86

Table 2: Search results (NF: Not Found).

The best results (Table 2) are obtained with the ATT&CK BERT model where in 34.21% of the 532 samples, the most similar procedure uses the same attack technique, in 42.67% the correct technique is ranked in the first two positions, in 64.29% of the cases the correct technique is among the top 10 results and in 70.86% among the top 20 results. However, in 20.30% cases none of the first 100 retrieved procedures shares the same attack technique as the input procedure.

We further analyze the procedures ranked first using ATT&CK BERT embeddings. As shown in the IR column of Table 3, in 182 out of the 532 cases, the result and the procedure being classified share the same attack technique. In 23 cases, the retrieved result corresponds to a technique which is a child (e.g., T1548.003) of that of the procedure being classified (e.g. T1548). In 16 cases, the result corresponds to a parent technique. Finally, in 24 cases the two procedures share sibling techniques (e.g., T1548.001 and T1548.003). Exact matches (EM) are 34.21%, whereas partial matches (i.e., EM+CM+PM+SM) amount to 46.05%.

4.2.2 Why does information retrieval fail for some attack procedures?

As mentioned previously, in this work we posit that attack procedures classified under the same MITRE technique should exhibit semantic coherence, resulting in a clustered distribution of their meanings. However, when performing information retrieval, there are cases in which the procedures ranked top do not share the same attack technique. This clustering experiment seeks to provide a means to evaluate embedding models and to further delve into the reasons for classification mismatches, demonstrating that accurately classified samples in the TTPEval Dataset correspond to techniques with procedures characterized by low semantic variability and high disjointedness from procedures of other techniques. Conversely, we aim to expose that incorrectly classified samples exhibit one of the following patterns: (i) high semantic variability within the technique's procedures, (ii) semantic similarity between the technique's procedures and those of other techniques, or (iii) the embeddings fail to capture subtle differences in meaning as the cybersecurity domain is very specific. To this end, we use DBSCAN (Density-Based Spatial Clustering of Applications with Noise) to cluster the embeddings within our database of procedures to analyze why some cases fail. DBSCAN requires the specification of two key parameters: ϵ, which sets the similarity threshold for points to be considered part of the same cluster, and minsamples, which determines the minimum number of points within the ϵ-neighborhood of a point required for it to be classified as a core point. A smaller ϵ value indicates that points must be more similar (i.e., exhibit higher cosine similarity) to be grouped together.

We first examine four MITRE techniques (T1053, T1134, T1490, and T1203) for which the sample procedures in the TTPEval Dataset have been accurately classified. Out of the 12,999 procedures cataloged in the MITRE framework, a subset of 317 procedures are specifically associated with these four techniques. The results of this clustering analysis are presented in Table 4, which shows the number of procedures clustered together and their corresponding techniques for different ϵ values. Applying DBSCAN to the ATT&CK BERT embeddings of the 317 procedures, we observe that a small ϵ value yields only two clusters. However, as ϵ increases, the number of outlier samples (i.e. those that do not fulfill the conditions to be clustered as per ϵ and the minimum number of points at lower distance than this threshold) decreases, and the procedures are ultimately grouped into four distinct clusters without overlap.

We proceed by performing a similar analysis on five MITRE techniques (T1552, T1110,

	IR	Llama 3.1 8B				Llama 3.1 70B			
	Top 1	**Zero-shot**	**RAG Few-shot**	**RAG MCQ**	**CRAG MCQ**	**Zero-shot**	**RAG Few-shot**	**RAG MCQ**	**CRAG MCQ**
EM	182	26	222	207	210	147	280	211	213
CM	23	5	13	17	20	10	25	24	22
PM	16	20	21	23	25	73	21	35	38
SM	24	9	14	16	13	14	9	9	10
Errors	287	472	262	269	264	288	197	253	249
EMA (%)	34.21	4.89	41.73	38.91	39.47	27.63	52.63	39.66	40.04
PMA (%)	46.05	11.28	50.75	49.44	50.38	45.86	62.97	52.44	53.20

Table 3: Accuracy results with ATT&CK BERT embeddings, Llama 3.1 8B and 70B.

ϵ	**Cluster 0**	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Outliers (no cluster)**
0.1	T1053: 58	T1490: 24	-	-	T1053: 115, T1134: 57, T1490: 12, T1203: 51
0.2	T1053: 108	T1490: 34	T1134: 15	T1203: 24	T1053: 65, T1203: 27, T1134: 42, T1490: 2
0.3	T1053: 146	T1134: 46	T1203: 37	T1490: 35, T1053: 1	T1134: 11, T1053: 26, T1203: 14, T1490: 1
0.4	T1053: 161	T1134: 51, T1053: 1	T1203: 49	T1490: 35, T1053: 2	T1134: 6, T1490: 1, T1053: 9, T1203: 2

Table 4: Clustering results of ATT&CK BERT embeddings over the 317 procedures associated to MITRE techniques T1053, T1134, T1490 and T1203 (minimum samples for a cluster: 5).

T1078, T1555, and T1005) that exhibited high classification error rates for the sample procedures in the TTPEval Dataset. Out of the 12,999 procedures catalogued in the MITRE framework, a subset of 541 procedures are associated with these five techniques. Clusters emerge using a smaller ϵ value of 0.05 (Table 5), indicating that the procedures from different techniques expose higher similarity. Furthermore, as ϵ increases, there is always one cluster that contains diverse techniques, proving the significant overlap between the embeddings of the procedures associated to these techniques.

Finally, we repeat the experiments by vectorizing the procedures using the BGE-M3 dense embeddings model, which yielded lower classification accuracy. This indicates that this model is not capable of capturing the nuanced distinctions between cybersecurity terms, resulting in less distinct cluster separation. This is shown in Table 6, which reveals that for these embeddings, clustering is less effective with smaller ϵ values than in Table 4, suggesting that procedures belonging to the same technique are less accurately semantically represented. For the techniques exhibiting high classification error, Table 7 also shows that a larger ϵ value than in Table 5 is necessary to cluster samples.

4.2.3 Evaluation of dynamic few shot learning

We employ a few-shot learning approach wherein the 20 most semantically similar procedures to the target classification instance are selected dynamically and provided as contextual information to the LLM (see Figure 5 for an example prompt). The results with Llama 3.1 8B and Llama 3.1 70B are shown in Table 3 in the columns labeled as "RAG Few-shot".

This approach yields significant improvements in classification accuracy in comparison to the zero-shot and information retrieval (IR) approaches shown in the columns labeled as such in the same Table 3.

In the case of Llama 3.1 8B, the EMA metric increases from 4.89% to 34.21% in IR using cosine similarity and to 41.73% with dynamic few-shot RAG. The issue encountered when utilizing Llama 3.1 8B with few-shot RAG is that the technique name generated as a response is typically accurate or semantically similar to the precise name. However, the associated identifier does not often correspond to the correct technique. Consequently, the manual verification of the provided identifier is necessary. In many instances, reassignment to the correct identifier is required. The accuracy of 41.73% considers the answer as correct when either the name or the identifier of the technique provided are correct. With larger model size

ϵ	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Outliers (no cluster)
0.05	T1555: 13	T1005: 5	T1555: 15	T1005: 9	T1005: 9	-	T1552: 63, T1078: 99 T1005: 173, T1555: 105, T1110: 50
0.1	T1555: 73 T1552: 10 T1078: 1 T1110: 5	T1005: 18	T1005: 31	T1005: 10	T1005: 6	T1552: 2, T1555: 4	T1078: 98, T1005: 131 T1555: 56, T1552: 51, T1110: 45
0.2	T1005: 134	T1555: 122, T1110: 35 T1552: 38, T1005: 5, T1078: 10	T1078: 6	T1078: 5	T1078: 4	-	T1005: 57, T1552: 25 T1078: 74, T1555: 11, T1110: 15
0.3	T1005: 163 T1555: 1	T1110: 48, T1552: 50 T1555: 130, T1078: 80, T1005: 6	T1005: 5	T1552: 5	-	-	T1110: 2, T1552: 8 T1005: 22, T1078: 19, T1555: 2

Table 5: Clustering results of ATT&CK BERT embeddings over the 541 procedures associated to MITRE techniques T1552, T1110, T1078, T1555 and T1005 (minimum cluster samples: 5).

ϵ	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Outliers (no cluster)
0.1	-	-	-	-	T1203: 51, T1134: 57 T1053: 173, T1490: 36
0.2	T1053: 8	T1490: 5	-	-	T1203: 51, T1134: 57 T1053: 165, T1490: 31
0.3	T1053: 107, T1134: 1	T1490: 26	T1134: 7	T1203: 36, T1134: 1	T1203: 15, T1134: 48 T1053: 66, T1490: 10
0.4	T1490: 36, T1203: 50 T1134: 54, T1053: 173	-	-	-	T1134: 3, T1203: 1

Table 6: Clustering results of BGE-M3 dense embeddings over the 317 procedures associated to MITRE techniques T1053, T1134, T1490 and T1203 (minimum cluster samples: 5).

(Llama 3.1 70B), accuracy improvements are also remarkable, achieving an improvement from 27.63% in zero-shot learning to 52.63% with dynamic few-shot RAG (see Table 3).

System prompt: You are a security expert that classifies attack procedures by Mitre attack technique used.

User prompt: Classify the Mitre attack technique that was used in the following procedure.
Examples:
Procedure: XXX.
Answer: XXX
...
Procedure: XXX.
Answer:

Figure 5: Prompt example for dynamic few-shot learning.

4.2.4 RAG approach evaluation

In this case, instead of providing procedure-answer pair examples to the LLM, we expand the prompt with a list of possible answers simulating a Multiple Choice Question (MCQ) answering setting. Specifically, we retrieve the 20 most semantically similar procedures (as in the previous experiment) and extract the pertinent attack techniques, which comprise the multiple-choice options. We then ask the LLM to choose the most appropriate technique from the provided options. Additionally, we introduce a last option, "None of the above", to accommodate instances where none of the suggested techniques is correct. An example of a user prompt used to implement this MCQ-based approach is shown in Figure 6.

The results with Llama 3.1 8B and Llama 3.1 70B are shown in Table 3, in the columns labeled as "RAG MCQ". For smaller models, the accuracy achieved is comparable to that of dynamic few-shot RAG. The primary advantage is that the technique names are consistently precise, eliminating the need for manual verification. However, for larger models, the accuracy diminishes significantly. This suggests that the knowledge acquired during model training is utilized, and the language model does not confine its responses to the provided examples.

Classify the Mitre attack technique that was used in the following procedure.
Procedure: {procedure}
Options:
A) Disable Windows Event Logging
B) Employee Names
C) Spearphishing Link
D) None of the above
Choose the correct answer (A,B,C or D) only.
Always return in this format: 'ANSWER: X'

Figure 6: Example of a user prompt used in multiple-choice approach with RAG.

4.3 RQ3: CRAG Evaluation

These experiments investigate whether the robustness of the classification system can be augmented by incorporating a relevance ranking mechanism to the procedures re-

ϵ	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Outliers (no cluster)
0.05	-	-	-	-	-	T1078: 99, T1555: 133 T1005: 196, T1110: 50 T1552: 63
0.1	-	-	-	-	-	T1078: 98, T1005: 131 T1555: 56, T1552: 51 T1110: 45
0.2	T1005: 37, T1078: 2 T1552: 1	T1552: 1, T1005: 1 T1078: 1, T1555: 1 T1110: 1	T1005: 5	T1078: 3 T1005: 1 T1555: 2	T1555: 1 T1078: 3 T1005: 1	T1005: 151, T1555: 129 T1552: 61, T1078: 90 T1110: 49
0.3	T1078: 72, T1555: 108 T1005: 156, T1110': 27 T1552: 39	T1552: 4, T1078: 1	-	-	-	T1552: 20, T1078: 26, T1110: 23, T1555: 25, T1005: 40

Table 7: Clustering results of BGE-M3 dense embeddings over the 541 procedures associated to MITRE techniques T1552, T1110, T1078, T1555 and T1005 (minimum cluster samples: 5).

trieved using cosine similarity. To this end, we increase the retrieval set from 20 to the top 50 most semantically similar procedures. Each procedure's relevance is assessed using the prompt shown in Figure 7, and the procedures are subsequently ranked according to the produced scores. We then select the top 20 procedures from the re-ranked list and repeat the multiple-choice experiments.

```
You are an expert at evaluating attack procedures relevance.

Rate on a scale from 0 to 1 how relevant the two attack procedures
are considering the attack technique used.
0 means completely irrelevant because both procedures use
different attack techniques, 1 means perfectly relevant because
both of them use the same attack technique.

Provide ONLY the score as a float between 0 and 1.
```

Figure 7: User prompt for the evaluation of the relevance of the retrieved procedure.

The results with Llama 3.1 8B and 70B are shown in Table 3 in the columns labeled as "CRAG MCQ". The implementation of this technique does not yield any substantial improvements to the results for any of the language models. Besides, it incurs in significant computational overheads and inference latencies due to the numerous LLM calls required for the ranking process.

5 Discussion and Future Work

While LLMs have demonstrated substantial improvements in their capabilities, they continue to exhibit limitations when tasked with domain-specific applications that demand precise technical understanding and nuanced vocabulary comprehension, such as cybersecurity attack procedure classification.

To enhance the classification accuracy achieved through semantic similarity, we introduce a RAG architecture that combines the strengths of retrieval and generative models. RAG techniques have proven to be very useful in classification tasks where the number of available categories is large, which can lead to increase confusion in LLMs. Selecting appropriate examples helps the LLM reduce ambiguity, improving the exact match accuracy from 4.89% (zero-shot) to 41.73% (RAG few shot) in the case of Llama 3.1 8B and from 27.63% to 52.63% in the case of Llama 3.1 70B. When considering partial match accuracy, our results range from 11.28% (zero-shot) to 50.75% (RAG few shot) for Llama 3.1 8B, and from 45.86% to 62.97% for Llama 3.1 70B. Furthermore, for smaller models, a multiple-choice approach can be an effective strategy to mitigate errors in generation and the subsequent need for manual validation. In contrast, CRAG techniques fail to provide significant enhancements for procedure classification when compared to the dynamic few-shot RAG approach.

The ATT&CK Bert embedding model was found to precisely represent the meaning of the procedures. We also show that a clustering analysis can systematically unveil the semantic variability in the samples and the disjointness of the classification categories when using this embedding model, thereby assessing whether it effectively captures the meaning of a vocabulary or technical domain.

Recent advances have introduced LLMs with enhanced reasoning capabilities, such as DeepSeek R1 (Guo and others, 2025), which generate intermediate steps that make their thought process more transparent. As a future research direction we plan to explore whether incorporating intermediate reasoning steps can improve the accuracy of attack procedure classification.

Acknowledgements

This work has been partially supported by the European Commission (CertifAI project, Ref. Ares(2022)8966412, funded under HORIZON-CL3-2022-CS-01-04). J. Del Ser also acknowledges funding support from the Basque Government through the consolidated research group MATHMODE (IT1456-22). And A. Atutxa from the HiTZ Center and the Basque Government, Spain (Research group funding IT1570-22) as well as by MCIN/AEI/10.13039/5011 00011033 Spanish Ministry of Universities, Science and Innovation by means of the project: EDHIA PID2022-136522OB-C22 (also supported by FEDER, UE).

References

Al-Sada, B., A. Sadighian, and G. Oligeri. 2024. MITRE ATT&CK: State of the art and way forward. *ACM Computing Surveys*, 57(1).

Alves, P. M. M. R., G. P. R. Filho, and V. P. Gonçalves. 2022. Leveraging BERT's power to classify TTP from unstructured text. In *2022 Workshop on Communication Networks and Power Systems (WCNPS)*, pages 1–7.

Barnett, S., S. Kurniawan, S. Thudumu, Z. Brannelly, and M. Abdelrazek. 2024. Seven failure points when engineering a retrieval augmented generation system. In *IEEE/ACM 3rd International Conference on AI Engineering-Software Engineering for AI*, pages 194–199.

Beltagy, I., K. Lo, and A. Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620.

Chen, J., S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu. 2024. BGE M3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *2019 Conference of the North American chapter of the Association for Computational Linguistics: Human Language Technologies, volume 1 (long and short papers)*, pages 4171–4186.

Dong, Q. et al. 2024. A survey on in-context learning. In *2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128.

Fayyazi, R., R. Taghdimi, and S. J. Yang. 2024. Advancing TTP analysis: harnessing the power of large language models with retrieval augmented generation. In *2024 Annual Computer Security Applications Conference Workshops (ACSAC Workshops)*, pages 255–261.

Fayyazi, R. and S. J. Yang. 2023. On the uses of large language models to interpret ambiguous cyberattack descriptions. *arXiv preprint arXiv:2306.14062*.

Formal, T., C. Lassance, B. Piwowarski, and S. Clinchant. 2021. SPLADE v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*.

Gao, Y. et al. 2024. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Grattafiori, A. et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Guo, D. et al. 2025. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Lassance, C., H. Déjean, T. Formal, and S. Clinchant. 2024. SPLADE-v3: New baselines for SPLADE. *arXiv preprint arXiv:2403.06789*.

Li, L., C. Huang, and J. Chen. 2024. Automated discovery and mapping ATT&CK tactics and techniques for unstructured cyber threat intelligence. *Computers & Security*, 140:103815.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *40th Annual Meeting of the Association*

for Computational Linguistics, pages 311–318.

Reimers, N. and I. Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-Networks. In *2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Salemi, A. and H. Zamani. 2024. Evaluating retrieval quality in retrieval-augmented generation. In *47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2395–2400.

Santhanam, K., O. Khattab, J. Saad-Falcon, C. Potts, and M. Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734.

Sauerwein, C. and A. Pfohl. 2022. Towards automated classification of attackers' TTPs by combining NLP with ML techniques. *arXiv preprint arXiv:2207.08478*.

Tihanyi, N., M. A. Ferrag, R. Jain, T. Bisztray, and M. Debbah. 2024. CyberMetric: a benchmark dataset based on retrieval-augmented generation for evaluating LLMs in cybersecurity knowledge. In *IEEE International Conference on Cyber Security and Resilience (CSR)*, pages 296–302.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Wan, S. et al. 2024. CYBERSECEVAL 3: Advancing the evaluation of cybersecurity risks and capabilities in large language models. *arXiv preprint arXiv:2408.01605*.

Yan, S.-Q., J.-C. Gu, Y. Zhu, and Z.-H. Ling. 2024. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*.

Zhang, T., V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.