big data and cognitive computing

MDPI

Article

Labeling Network Intrusion Detection System (NIDS) Rules with MITRE ATT&CK Techniques: Machine Learning vs. Large Language Models

Nir Daniel [1,2,*], Florian Klaus Kaiser [3], Shay Giladi [1], Sapir Sharabi [1], Raz Moyal [1], Shalev Shpolyansky [1], Andres Murillo [4], Aviad Elyashar [2,5] and Rami Puzis [1,2]

[1] Department of Software and Information Systems Engineering, Ben-Gurion University, Beer Sheva 8410501, Israel; puzis@bgu.ac.il (R.P.)
[2] Cyber@BGU, Cyber Labs at Ben-Gurion University, Beer Sheva 8410501, Israel
[3] Agnostic Intelligence AG, 6300 Zug, Switzerland
[4] Fujitsu Ltd., Kawasaki 212-0014, Japan
[5] Department of Computer Science, Shamoon College of Engineering, Ashdod 77245, Israel
* Correspondence: nirdanie@post.bgu.ac.il

Abstract: Analysts in Security Operations Centers (SOCs) are often occupied with time-consuming investigations of alerts from Network Intrusion Detection Systems (NIDSs). Many NIDS rules lack clear explanations and associations with attack techniques, complicating the alert triage and the generation of attack hypotheses. Large Language Models (LLMs) may be a promising technology to reduce the alert explainability gap by associating rules with attack techniques. In this paper, we investigate the ability of three prominent LLMs (ChatGPT, Claude, and Gemini) to reason about NIDS rules while labeling them with MITRE ATT&CK tactics and techniques. We discuss prompt design and present experiments performed with 973 Snort rules. Our results indicate that while LLMs provide explainable, scalable, and efficient initial mappings, traditional machine learning (ML) models consistently outperform them in accuracy, achieving higher precision, recall, and F1-scores. These results highlight the potential for hybrid LLM-ML approaches to enhance SOC operations and better address the evolving threat landscape. By utilizing automation, the presented methods will enhance the analysis efficiency of SOC alerts, and decrease workloads for analysts.

Keywords: cyber threat intelligence; alerts investigation; natural language processing

check for updates

Academic Editor: Jun Wang

Received: 8 December 2024
Revised: 20 January 2025
Accepted: 23 January 2025
Published: 26 January 2025

Citation: Daniel, N.; Kaiser, F.K .; Giladi, S.; Sharabi, S.; Moyal, R.; Shpolyansky, S.; Murillo, A.; Elyashar, A.; Puzis, R. Labeling Network Intrusion Detection System (NIDS) Rules with MITRE ATT&CK Techniques. *Big Data Cogn. Comput.* **2025**, *9*, 23. https://doi.org/10.3390/bdcc9020023

1. Introduction

Network Intrusion Detection Systems (NIDSs), such as Snort (https://www.snort.org/) or Suricata (https://suricata.io/), analyze network traffic to identify and mitigate potential threats. In this way, NIDS rules are an effective way to identify specific attack methods [1]. NIDS-generated alerts must be interpreted by security analysts to extract relevant information to understand ongoing attacks. To address detected attacks, security analysts must develop accurate hypotheses regarding attack techniques and attackers' intentions, generating actionable insights [2]. Leveraging such actionable insights, security analysts are empowered to take appropriate countermeasures.

The increasing sophistication and diversity of cyber attacks necessitates a complex rule-based approach in NIDSs (i.e., leading to a significant number of rules that NIDSs operate on). The plethora of different NIDS rules that analysts need to understand and the lack of links between those rules (e.g., Snort) and extensive Cyber Threat Intelligence

(CTI) requires significant skills, attention, and cyber security expertise [3]. Given the scarcity of experienced security analysts, the need for labeled NIDS rules supporting (and partially automating) the process of identifying attacks, detecting adversary behavior, and hypothesizing about probable next steps is highlighted in recent publications [4].

To reduce human effort and help analysts stay ahead of malicious actors, organizations are continuously incorporating Artificial Intelligence (AI) into their security workflows [5]. A primary application of AI is automating the investigation of security events, which helps alleviate alert fatigue among analysts. Beyond this, AI presents numerous opportunities to enhance defensive security operations. Hereby, Large Language Models (LLMs) are being used to automate various cyber security tasks that previously relied on human effort [6,7]. By automating these processes, AI allows security professionals to focus on higher-level tasks, adding significant value to organizations and society.

In this paper, we leverage AI to associate NIDS rules with relevant MITRE ATT&CK Techniques (https://attack.mitre.org/), easing analysts' work in extracting actionable insights into ongoing attacks and craft hypotheses.

We investigate the ability of using three open-accessible LLMs to assist cyber security experts. The research problem targeted within this work is formulated by Guerra et al. [8] and Gjerstad et al. [9], demanding novel methods for labeling high volumes of network traffic-related data with high quality and speed. Namely, we explore, optimize, and evaluate the usage of ChatGPT (https://chatgpt.com/), Claude (https://claude.ai/), and Gemini (https://gemini.google.com/) to automate the labeling of Snort rules with MITRE ATT&CK tactics and techniques, thereby providing insights into the current state of a cyber attack.

Our main contributions are as follows:

- A dataset of 973 labeled Snort community NIDS rules;
- Formalizing NIDS rule labeling as a conditional text generation problem, treating it as a maximization task;
- Introducing a workflow for employing LLMs to label NIDS rules with MITRE ATT&CK tactics and techniques and generate an automated machine learning (ML)-based labeling procedure;
- Three ML models for labeling NIDS rules, automatically generated and trained by LLMs.

Leveraging LLMs to label NIDS rules offers two key benefits: Firstly, it is possible to justify each suggested technique. The provided explanations and reasoning given through the LLMs can be especially beneficial for analysts with limited expertise in cyber security (e.g., limited experience). Secondly, LLMs leverage their extensive training on diverse cyber security knowledge from a variety of data sources, which contains cyber security knowledge from a wide range of different data sources including incident reports but also hacker chatter, etc. ML models on the other side frequently provide superior performance within the tasks provided; however, results are frequently limited with respect to their explainability.

The rest of the paper is structured as follows. Section 2 gives the relevant background. Section 3 presents related work. In Section 4, the methods, dataset, and experiments are presented. In Section 5, we introduce the results. Section 6 provides the discussion. Section 7 summarizes the key findings and offers a perspective on future research directions.

2. Background

2.1. Cyber Threat Intelligence

Cyber Threat Intelligence (CTI) is a proactive approach in computer and network security [10], focusing on the collection and analysis of data to derive actionable insights into potential or ongoing attacks. These insights improve decision-making processes by enabling the selection of appropriate defensive measures [11]. As outlined by Chismon et al. [11], CTI can be categorized into four types based on focus and depth:

- *Strategic CTI*: Aimed at non-technical audiences [12], this high-level intelligence assists organizational leaders in understanding the broader implications of cyber activities, including potential risks and their impacts.
- *Operational CTI*: Provides detailed insights on impending attacks, consumed primarily by senior security staff, such as incident response teams leads. This intelligence supports day-to-day decision-making, but access is often limited to advanced organizations or nation states.
- *Tactical CTI*: Known as as Tactics, Techniques, and Procedures (TTPs), this intelligence details adversary methodologies and is used by Security Operations Centers (SOCs) to test and enhance defensive measures.
- *Technical CTI*: Provides raw data, such as IP addresses or hash values, which are time sensitive and must be used promptly, as they can quickly become obsolete.

Another categorization of CTI is based on its source, which can be either **network based** or **host based**.

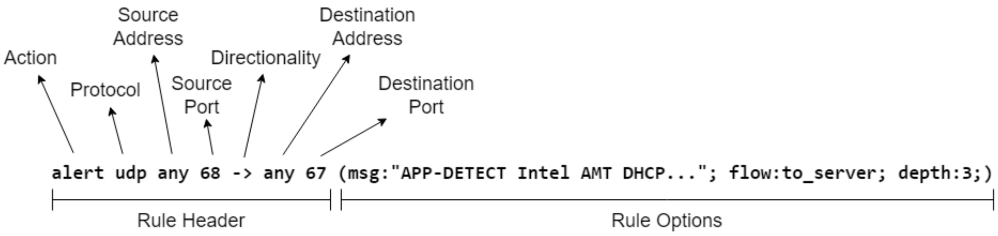
Low-level CTI, including Indicators of Compromise IoCs, is especially valuable for automating cyber security processes, enabling more efficient decision-making [13]. Tools such as Threat Hunting (TH), Endpoint Detection and Response (EDR), and Intrusion Detection System (IDS) often rely on technical and tactical CTI to identify and address potential threats [14]. A potential lies in expanding the automation levels [14], aiding organizations in gaining visibility of the threat landscape, identifying attacks and associated TTPs, and responding effectively [15].

The MITRE ATT&CK framework is an essential resource for CTI, providing a curated knowledge base of adversarial behavior, which includes detailed mappings of post-compromise tactics and techniques across various platforms [16]. According to Strom et al. [16], the ATT&CK framework centers on understanding how external adversaries infiltrate and act within computer networks. It therefore serves as an extensive database of tactical CTI, encompassing post-compromise adversary TTPs across various operating systems, including Windows, Linux, and macOS. Additionally, it spans multiple technological domains such as enterprise environments, mobile devices, cloud systems, and Industrial Control Systems (ICSs). As of November 2024, the ATT&CK framework included 203 attack techniques and 453 sub-techniques associated with 14 tactics for enterprise systems and 84 techniques linked to 12 tactics specific to ICSs.

2.2. Intrusion Detection Systems

IDSs can be differentiated to Host-based Intrusion Detection Systems, HIDSs, and NIDS [17]. The visibility coverage of different IDSs depends especially on the data sources they analyze. NIDSs, for example, offer only limited visibility inside the host machine [18]. Therefore, only a limited set of attack techniques is detectable with high confidence when relying on NIDS. Likewise, HIDS provides restricted visibility by being limited on the analysis of system logs. In an effort to systemize the analysis of data log source quality, visibility coverage, and detection coverage, Detect Tactics, Techniques and Combat Threats (*DeTT&CK*) was developed (https://github.com/rabobank-cdc/DeTTECT). *DeTT&CK* provides a useful means to analyze the visibility coverage of different intrusion detection systems and techniques identifiable through investigations of specific data sources.

Snort (https://www.snort.org/) is a lightweight NIDS built on the Libpcap library, offering efficient packet-filtering capabilities [19]. Snort rules, which are straightforward and easily interpreted, are divided into two primary components [20] (see Figure 1): The first component, known as the header, specifies attributes such as action, protocol, source and destination addresses, source and destination ports, and traffic direction. The second component, known as the rule options, contains a list of keyword and argument pairs enclosed within parentheses. These elements collectively enable Snort to detect and respond to specific network-based attack patterns effectively.



```
alert udp any 68 -> any 67 (msg:"APP-DETECT Intel AMT DHCP..."; flow:to_server; depth:3;)
```

Figure 1. Snort rule example.

2.3. Large Language Models

Generative Pre-trained Transformers (GPTs) are among the most advanced LLMs currently available, known for their ability to process and analyze complex tasks across various fields. This study leverages ChatGPT-4 along with other state-of-the-art LLMs, including Claude and Gemini, to enhance cyber security workflows. These models, with their advanced reasoning capabilities, have demonstrated potential in automating expert-driven processes [6].

In cyber security, LLMs are gaining recognition for their dual utility in both defensive and offensive applications [21]. On the offensive side, LLMs have raised concerns due to their ability to facilitate malicious activities, such as generating phishing emails or creating sophisticated attack scripts [22]. On the defensive side, however, these models have emerged as powerful tools for automating labor-intensive tasks [23]. Their vast training datasets, which include security-related knowledge, allow them to support cyber security experts by generating actionable insights and assisting in complex decision-making processes.

This work explores a further use case in analyzing the feasibility of using LLMs to label NIDS rules with MITRE ATT&CK techniques and furthermore provide contextual explanations for each label. By leveraging the comprehensive training and reasoning capabilities of models such as ChatGPT, Claude, and Gemini, we aim to evaluate and compare their effectiveness in enriching CTI workflows. This marks a significant advancement in the adoption of LLMs as a scalable and efficient means to assist security analysts in mitigating and understanding ongoing threats.

3. Related Work

Ishibashi et al. [24] highlight the need to implement AI-powered NIDSs and construct labeled training datasets. Gjerstad [9] points out the shortage of labeled cyber security datasets and encourages their creation (i.a., CTI labeled reports and labeled host-based, network-based data).

The growing abilities of LLMs are driving their use in many areas. With the fast pace and large volume of CTI sharing, automating labeling is essential to keep up with cyber threats.

3.1. Labeling Reports with CTI

Husari et al. [25] introduce TTPDrill, a tool for extracting TTPs unstructured text using a combination of natural language processing and information retrieval. Similarly, Legoy et al. [26] present rcATT, a tool that uses a Multi-Class Multi-Label Classifier (MCML-C) trained on reports linked to TTPs. Mendsaikhan et al. [27] adopt Multi-Label Classification

(ML-C) on vectors from vulnerability descriptions (reports) and correlate those vectors to adversary techniques. Extractor [28] can extract attack behaviors as provenance graphs from unstructured text. TIM [29] is a framework for mining TTPs intelligence with added threat context from unstructured data. Li et al. [30] develop AttacKG, a similar tool that extracts techniques from reports. The latest tool, TTPHunter [31], focuses on APT reports and uses SecureBERT [32].

The above methods are intended for extracting TTPs from unstructured text. However, it is unclear how they will perform on structured CTI sources and on input that demands a high level of expertise and context information such as NIDS rules (e.g., an understanding of benign and malicious network flows and link techniques that generate such network traffic).

3.2. Labeling of Host-Related Data with CTI

Landauer et al. [17] introduce a method targeting the labeling within HIDSs. The method proposed aims at labeling system logs.

Gabrys et al. [33] integrate LLMs and ML to map host-based Wazuh IDS rules to MITRE ATT&CK techniques, achieving high classification accuracy and enhancing alert interpretability. Their focus on host-based rules, which describe processes and endpoint events, aligns more directly with ATT&CK techniques compared to network-based rules.

Compared to those works presented, our work focuses on network-based intrusion detection rather than on host-based intrusion detection. Although the work presented by Gabrys et al. [33] is comparable with respect to the use of LLMs and ML, the entirely different focus on NIDS rules differentiates the works substantially.

3.3. Labeling of Network-Related Data with CTI

3.3.1. Labeling Network Packets

McPhee [34] explains that the network-based detection of ATT&CK techniques using network sensors often focuses on Windows-specific protocols and excludes other systems like ICS. To address this, McPhee [34] uses the Zeek (https://zeek.org/) monitoring system to detect techniques in these environments. However, this approach requires the manual definition of detection methods for each technique, making it time-consuming and reliant on expert knowledge. Arafune et al. [35] focus on ICS and create an automated TH system that detects attacks in network traffic using open-source tools. Their approach links network traffic to attack techniques but relies on a signature-based method, requiring a human analyst to label each signature once to identify the matching techniques.

Garcia and Valeros [36] introduce a tool to label Zeek network flows with parts of the MITRE ATT&CK framework. Also, Masumi et al. [37] focus on labeling network packets to attack information. Gjerstad [9] proposes a method for labeling network datasets using MITRE CALDERA, a tool for testing system security. The method matches techniques labeled in CALDERA reports to simulated attack traffic. Furthermore, Bagui et al. [4] introduce a dataset created using Zeek containing network traffic which is labeled using the MITRE ATT&CK framework. The mapping is based on a rule-based mapping process, where links to a specific ATT&CK technique are generated using pre-configured mappings (i.e., already labeled mission logs) while not specifying the labeling process. RADAR [38] is a tool for detecting malicious behavior in network traffic. It identifies TTPs using both feature-based and heuristic-based detection rules. However, its limitation is that feature-based rules are manually created for each technique, so the system currently supports only 17 techniques.

Jüttner et al. [39] propose ChatIDS, a system that employs LLMs to translate IDS alerts into intuitive explanations and actionable countermeasures for non-expert users, aiming to improve cyber security in home and small network environments. By focusing on accessibility, ChatIDS simplifies alerts to help users understand threats and respond appropriately without requiring technical expertise. In contrast, our work uses LLMs to label NIDS rules with MITRE ATT&CK techniques, providing detailed, actionable insights for professional analysts. While ChatIDS emphasizes ease of use and comprehensibility, our solution is designed to enhance the efficiency and precision of high-level CTI workflows in complex, large-scale cyber security operations (labeling with standardized machine readable labels and explainable mappings). The evaluation methods further highlight these differences. ChatIDS relies on qualitative feedback from interdisciplinary experts to assess usability and practical applicability for non-technical users. In contrast, our study uses quantitative metrics—precision, recall, and F1-score—to evaluate the accuracy and scalability of LLMs, reflecting its focus on meeting the demands of expert-driven, operational environments.

3.3.2. Network Intrusion Detection Rules Labeling

The work closest to ours is that presented by Lin et al. [3]. They propose a mechanism which uses text mining and machine learning (especially Decision Tree (DT), K-Nearest Neighbor (KNN), Support Vector Machines (SVMs), and Random Forest (RF)) for labeling NIDS rules with 12 attack tactics, in order to aid experts during the TH process. They reach a F1-score of approximately 0.9. In their work, the labeling of rules is restricted to tactics, significantly limiting its usability. Furthermore, labeling rules with tactics is easier than those with techniques because there are only 26 tactics compared to over 203 techniques (see Section 2.1). Furthermore, Daniel et al. [40] provide a proof of concept demonstrating the potential of ChatGPT as a tool for enriching NIDS rules with MITRE ATT&CK techniques.

We expand the approach presented by Daniel et al. [40] by (1) expanding the dataset of labeled rules, and provide it as a benchmarking, training, or comparing tool for other research studies in the field, (2) incorporating additional LLMs, specifically, Claude and Gemini, to evaluate their effectiveness in performing similar tasks, (3) expanding the labeling workflow to cover NIDS rules labeling, comprehensively relying on MITRE ATT&CK (labeling NIDS rules with tactics and techniques). Furthermore, we formalize the task of labeling NIDS rules as a conditional text generation problem and contribute a comprehensive prompt engineering approach.

Besides investigating the ability of LLMs to label NIDS rules, we introduce three ML models for NIDS rule labeling. This broader investigation enables a comparative analysis of LLM and ML-model capabilities and provides deeper insights into the applicability of AI for automating cyber security processes.

Table 1 provides an overview of the related work, highlighting some key extensions we incorporate within our work (for increasing readability, we use the following abbreviations: Attack Type (AT), Explanation (E), Lowest Common Ancestor (LCA) Named Entity Recognition (NER), Tactic (TA), and Technique (T)).

Table 1. Related work.

Citation	Labeling			Methodology		
	Input	Output	Evaluation	LLM	ML	Other
Husari et al. [25]	Reports	TA; T	Quantitative	-	SVM	TF-IDF
Legoy et al. [26]	Reports	TA; T	Quantitative	-	SVM; ML-C; MCML-C	TF; TF-IDF; Word2Vec
Mendsaikhan et al. [27]	Reports	T	Quantitative	-	ML-C	TF-IDF; Bag of Words
Satvat et al.[28]	Reports	T	Quantitative	BERT	-	-
You et al. [29]	Reports	TA; T	Quantitative	-	-	TF-IDF

Table 1. *Cont.*

Citation	Labeling			Methodology		
	Input	**Output**	**Evaluation**	**LLM**	**ML**	**Other**
Rani et al. [31]	Reports	TA; T	Quantitative	SecureBERT	-	-
Li et al. [30]	Reports	T	Quantitative	-	-	Regex; NER; LCA
Landauer et al. [17]	Host-based logs	TA; T	Qualitative	-	-	Rule-based
Gabrys et al. [33]	HIDS-rules	TA; T	Quantitative	BERT	-	-
McPhee [34]	Network traffic (Zeek)	T	Quantitative	-	-	Rule-based
Arafune et al. [35]	Network traffic	TA; T	Quantitative	-	SVM	-
Garcia and Valeros [36]	Network traffic	T	Qualitative; Quantitative	-	-	Rule-based
Masumi et al. [37]	Network traffic	AT	Quantitative	-	-	Correlation analysis
Gjerstad [9]	Network traffic	T	Qualitative	-	-	Rule-based
Bagui et al. [4]	Network traffic	TA; T	Not specified	-	-	Rule-based
Sharma et al. [38]	Network traffic	TA; T	Quantitative	-	-	Rule-based
Jüttner et al. [39]	IDS alerts (Surricata, Snort, Zeek)	E	Qualitative	ChatGPT	-	-
Lin et al. [3]	NIDS-rules	TA	Quantitative	-	SVM; RF; DT; KNN	TF-IDF
Daniel et al. [40]	NIDS-rules	T	Quantitative	ChatGPT	-	Keyword-based
Our work	NIDS-rules	TA; T	Quantitative	ChatGPT; Claude; Gemini	SVM, RF, GBM	TF-IDF

4. Materials and Methods

Figure 2 presents an overview of the stages discussed in this section. We present two approaches for labeling NIDS rules with MITRE ATT&CK techniques. In the first approach, we use LLMs and test different combinations of prompting strategies as explained in Section 4.1. In the second approach (see Section 4.2), we test ML models generated by the LLMs themselves. Based on the technique mapping, we label the NIDS rule with the relevant ATT&CK tactic(s) associated with the ATT&CK technique(s). While the aim of this work is to label NIDS rules with ATT&CK techniques, this procedure allows to compare with related work. The rest of this section covers the dataset collection, experiments, and evaluation metrics used to compare the performance of the two approaches.

4.1. Prompt Engineering for the NIDS Labeling Task

4.1.1. Problem Definition

In the realm of using LLMs, prompt engineering is a critical task that involves crafting effective cues to guide LLMs in generating the desired text outputs. Given the probabilistic nature of outputs generated by LLMs a fundamental challenge in prompt engineering is understanding the probability of a language model generating a specific text given a particular prompt. This probability can be formalized as a conditional probability. The Large Language Model (LLM)-based labeling can hence be regarded as a conditional text generation problem.

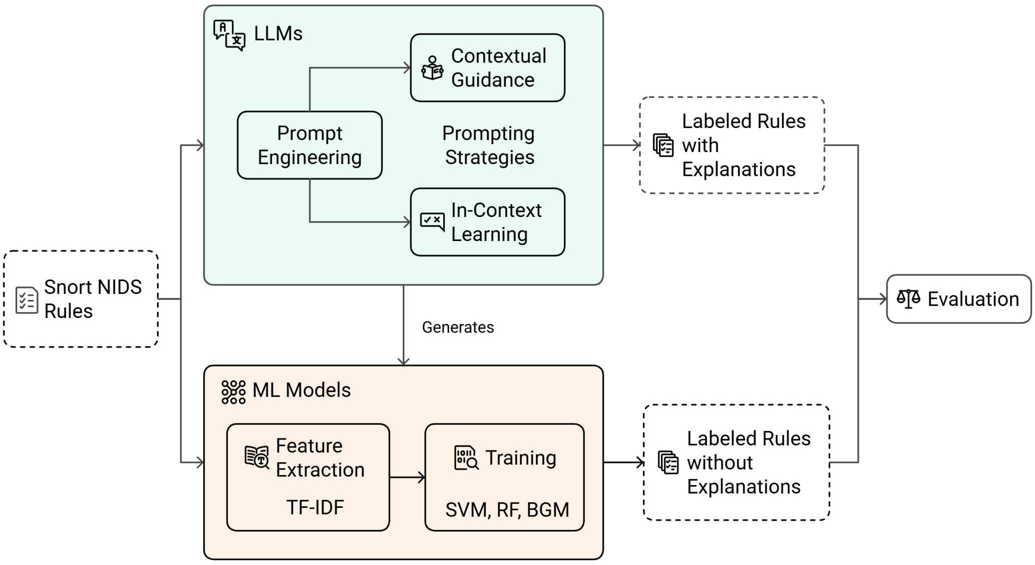


Figure 2. Materials and methods overview.

Let p be a prompt. The goal of engineering a prompt for LLM-based labeling in the context of labeling NIDS rules is to create a well-defined and efficient prompt p (cue or input given to the LLM) from an input space P that enables the accurate labeling of NIDS rules (output). With respect to this, the output of the LLM can be modeled as a function $f(p \in P, x \in X)$ where $x \in X$ is the NIDS rule to be labeled from a set of NIDS rules X. The problem is formulated as a maximization problem as follows:

$$max_{X,V} g(f(p,x),v) \tag{1}$$

where g describes the applicable evaluation metric (e.g., F1-score for the labeling task) and V is the ground truth. g is thereby defined as the labeling score between the predicted label $v_i^h \in V$ and the ground truth $v_i \in V$ for a given input x_i and prompt p.

The evaluation is based on the dataset D, where $x, v \in D$ is an instance of the dataset with the label x and the true label v, where the evaluation metric is calculated based on the average labeling score:

$$\frac{1}{|D|} \sum_{x,v \in D} g(f(p,x),v) \tag{2}$$

4.1.2. Prompt Template Generation

The function describing the output of the LLM can be specified as a function $Pr_{LLM}(y|p)$, where the probability Pr of generating a text y is modeled as a conditional probability depending on p, where p can be decomposed to its key components. The components of p are thereby represented by the task specification S, contextual information C, and methodological guidance M:

$$Pr_{LLM}(y|p) = Pr_{LLM}(y|S,C,M) \tag{3}$$

Each component is optimized based on prompt engineering techniques. The basic task specification S defines the desired output content. Direct questioning is employed to ask explicitly which MITRE ATT&CK technique(s) correspond to a given NIDS rule. We use the meta prompting technique of self-refinement prompting [41] to optimize the basic task specification. In this way, the prompts are refined iteratively based on the initial model responses to enhance clarity and reduce ambiguity.

Additionally, contextual guidance (C) is provided within the prompts to direct the model's focus toward relevant techniques to the NIDS rules. The techniques list (T) provided to the LLM is the complete list of techniques included in MITRE ATT&CK.

With regard to the methodological guidance (M), we test the use of competition questioning and few-shot ICL. Algorithm 1 describes competition questioning—this method involves dividing the list of MITRE ATT&CK techniques into smaller, more manageable batches and conducting multiple rounds of questioning to refine the models' labeling choices. The complete set of MITRE ATT&CK techniques is divided into 11 batches, and each LLM is tasked with labeling a given NIDS rule using only the techniques from one batch at a time. This strategy can be used to force the models to focus on a smaller set of techniques, reducing the cognitive load and potentially increasing the accuracy of their initial labeling. After the initial batch-based labeling, each model's answers are consolidated individually. The same model is then re-queried three additional times, but it is restricted to choose only from the techniques it has selected during the initial batch stage. This iterative process within each model aims to refine its previous selections, promoting convergence on the most likely techniques.

Algorithm 1 Competition questioning

```
Input:
•   LLM: Large Language Model instance
•   NIDS_Rule: Network Intrusion Detection System rule to be labeled
•   Techniques: Complete list of MITRE ATT&CK techniques
•   Batch_Size: Number of techniques per batch
Output:
•   Final_Labels: Refined set of techniques associated with the NIDS rule

Initialize:
$Selected\_Techniques \leftarrow \emptyset$
$Technique\_Batches \leftarrow Divide(\texttt{Techniques}, \texttt{Batch\_Size})$

Batch-wise Labeling:
for each $Batch \in Technique\_Batches$ do
    $Predicted\_Techniques \leftarrow Query\_LLM(\texttt{NIDS\_Rule}, Batch)$
    $Selected\_Techniques \leftarrow Selected\_Techniques \cup Predicted\_Techniques$
end for

Iterative Refinement:
for $i \leftarrow 1$ to R do                                   ▷ Number of refinement iterations
    $Refined\_Techniques \leftarrow Query\_LLM(\texttt{NIDS\_Rule}, Selected\_Techniques)$
    $Selected\_Techniques \leftarrow Refined\_Techniques$
end for

Output: return $Selected\_Techniques$
```

Furthermore, ICL [42,43] is applied by supplying examples of correctly labeled NIDS rules to demonstrate the desired output format and reasoning process. A key factor influencing the effectiveness of ICL is the quantity of examples provided. In general, a larger quantity of examples increases the effectiveness of the ICL jet, the marginal benefit of examples decreases, and a saturation can be observed [44]. Further factors influencing the effectiveness of ICL include the distribution of labels within the examples and the ordering of examples [44].

4.2. Machine Learning Approach

The machine learning approach uses the same dataset of Snort rules labeled with MITRE ATT&CK techniques as the LLM-based approach. However, in the machine learning experiments, both the training and test sets are utilized to build and evaluate the models. The dataset is split into train and test sets in a balanced manner to ensure a robust evaluation framework. The test set remains the same as the one used for evaluating the LLMs, allowing for a direct comparison of performance between the machine learning models and the LLM-based approach.

A novel aspect of this study is that the entire machine learning pipeline is designed and executed by the LLM itself. This process includes selecting suitable models, extracting features, writing code, and executing the machine learning tasks. The LLM has the autonomy to select the best models based on its own knowledge and experience, focusing on maximizing evaluation metrics with an emphasis on the F1-score.

Data preparation involves splitting the dataset into balanced training and test sets to ensure all classes are adequately represented. For feature extraction, the LLMs utilize Term Frequence Inverse Document Frequency (TF-IDF) to convert the Snort rules into numerical features. The MITRE ATT&CK technique IDs are binarized into multiple classes, allowing for multi-label classification. The LLMs autonomously identify TF-IDF as an appropriate feature extraction method and implement feature selection techniques to retain the most significant features relevant to the classification task.

The LLMs explore various machine learning classifiers suitable for multi-label classification, including RF, SVM, and Gradient Boosting Machine (GBM) in Algorithm 2.

Algorithm 2 Machine learning-based approach for Snort rule labeling

Input

- LLM: Large Language Model instance
- NIDS_Rule: Network Intrusion Detection System rule to be labeled
- Techniques: Complete list of MITRE ATT&CK techniques
- R: Number of iterations for model tuning
- Model_Type: Set of model types (RF, SVM, GBM)

Output

- Final_Model: Trained machine learning model with selected features

```
Initialize:
Dataset(NIDS_Rule, Techniques)
Selected_Features ← ∅
Model ← Null

Feature Extraction:
TF-IDF_Features ← Extract_TF-IDF(Dataset)
Selected_Features ← Feature_Selection(TF-IDF_Features)

Model Selection:
for each Model_Type do
    Model ← Train_Model(Model_Type, Selected_Features, Train_Set)
end for

Model Tuning:
for i ← 1 to R do
    Best_Model ← Select_Best_Model(Models)
    Refined_Model ← Refine_Model(Best_Model)
end for

Output: return Best_Model
```

4.3. Dataset Collection

To build a dataset comprising labeled NIDS rules (e.g., Snort), we collect Snort rules from the Snort community rules repository (https://www.snort.org/faq/what-are-community-rules), focusing on those rules that explicitly reference MITRE ATT&CK techniques. This involves identifying and extracting rules from their Snort webpage which contains a section of MITRE ATT&CK techniques. This approach ensures that the rules contain explicit references to MITRE ATT&CK techniques, providing a direct link between the detection rules and the adversarial techniques they are designed to identify. By selecting rules that include such references, we ensure that the dataset provides a relevant basis for evaluating the performance of various models in the task of automated labeling.

The final dataset comprises 973 rules, each mapped to one or more of 75 unique MITRE ATT&CK techniques.

4.4. Experimental Setup

The dataset is divided into train and test sets using an 80:20 split to facilitate the training and evaluation of machine learning models. The training set is used to train the models, while the test set is reserved for the final evaluation to ensure an unbiased performance assessment. The distribution of rules across these sets is balanced to ensure a representative sample of different MITRE ATT&CK techniques in each partition; however, some of the techniques appear very little in the dataset which makes them not feasible to be labeled by the ML models (necessitating a split in a test and training set). Therefore, we remove rules with fewer than 5 occurrences prior to the split, which leaves us with a total of 33 unique techniques across 900 rules—720 in the train set and 180 in the test. The performance of LLMs on the set of 42 rare techniques across 73 rules is also evaluated separately, using the prompt template that achieves the highest F1-score by each LLM on the test set.

Figure 3 presents the percentage of occurrences of each technique out of the total number of occurrences of techniques in the set—for both the train and the test sets. Figure 4 shows the percentage of occurrences of each tactic out of the total number of occurrences of tactics in the set—for both the train and the test sets.

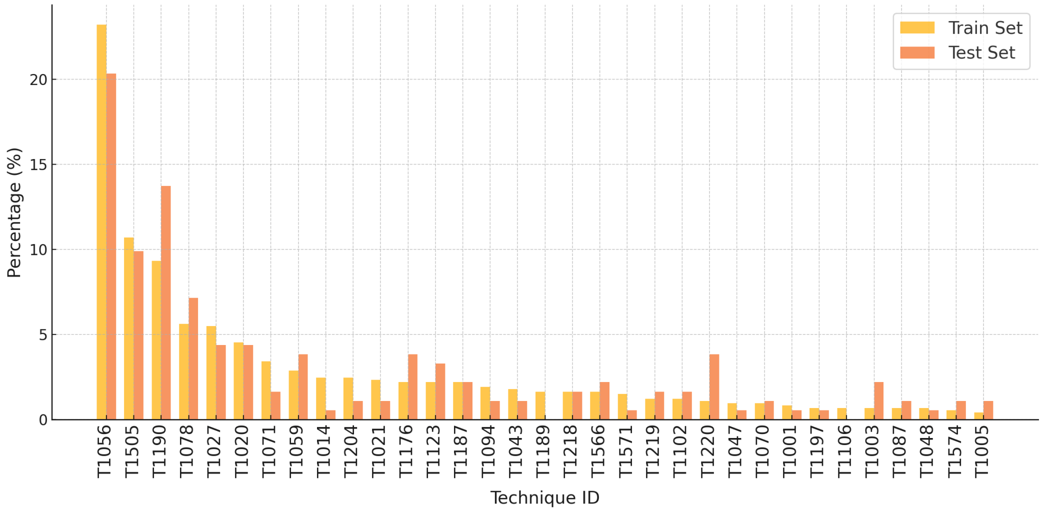


Figure 3. Distribution rules across ATT&CK techniques for both the train and the test sets.

Three LLMs—ChatGPT (chatgpt-4-turbo), Claude (claude-3-sonnet-20240229), and Gemini (gemini-1.0-pro)—are employed to assess their effectiveness in labeling NIDS rules with MITRE ATT&CK techniques. These models are chosen due to their advanced capabilities in processing complex language tasks, particularly in the context of cyber security.

To evaluate the effectiveness of both the LLMs and traditional ML approaches in labeling NIDS rules, a series of experiments is conducted under varying conditions. The experiments are designed to compare the performance of each approach using different configurations and scenarios.

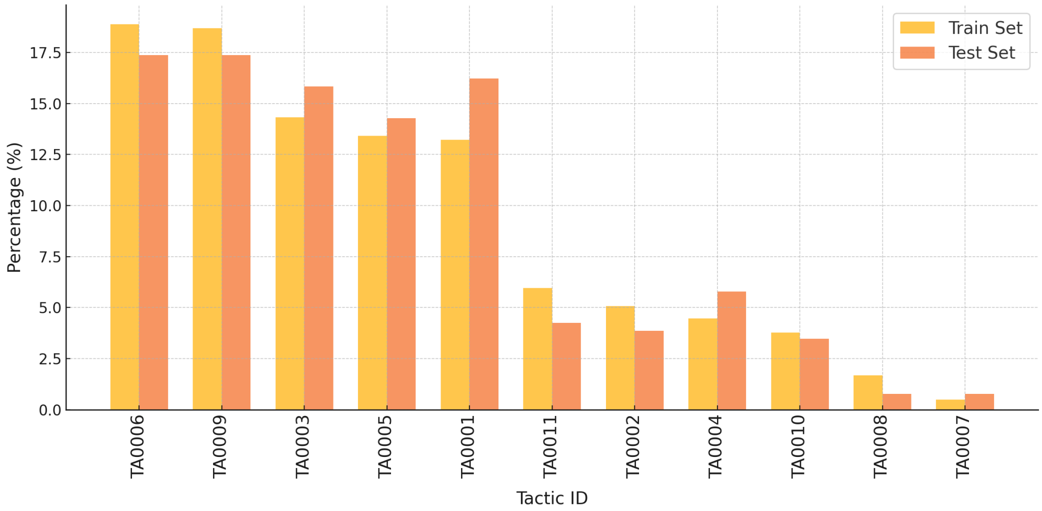


Figure 4. Distribution rules across ATT&CK tactics for both the train and the test sets.

4.4.1. LLM Experiments

The experiments conducted encompass a comprehensive set of scenarios to evaluate the performance of the LLMs across different conditions. These scenarios are combinations of using (T) or not using a technique's guide (no techniques, comprehensive set of techniques), providing methodological guidance by few-shot learning (zero, ICL_0, one, ICL_1, or two, ICL_2, examples). Each combination of these options is tested to thoroughly assess the strengths and weaknesses of each model under varying conditions.

4.4.2. Machine Learning Experiments

For the machine learning approach, both the training and test sets are utilized. The experiments involve training several classifiers, and Gradient Boosting Machines, which are selected and configured by the LLMs themselves. Feature extraction is performed using TF-IDF, and hyperparameter tuning is conducted before final testing on the test set.

4.5. Evaluation Metrics

The performance of both LLMs and machine learning models is assessed using precision, recall, and F1-score. These metrics are chosen to provide a balanced evaluation of the models' abilities to correctly label the rules (precision), cover the relevant techniques (recall), and balance both aspects (F1-score). Micro-averaging is employed for each evaluation metric described above:

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$\text{F1-score} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{6}$$

We include four baselines which enable to benchmark the results. Top-1 is defined as a frequency-based baseline. It describes a heuristic approach of selecting the most frequent technique as a label. Top-2 consistently describes the heuristic approach of selecting the two top most frequent techniques for each rule as a label. RT-1 is defined as a random baseline of selecting a random technique of the correct tactic. We hereby assume that an analyst is able to correctly identify the tactic detected by the NIDS rule (as there are models that assist in correctly labeling the NIDS rule). Out of the set of techniques related to the correctly identified tactic, the analyst needs to select a random technique, as no method is provided that assists in labeling. Likewise, RT-2 describes the baseline when selecting two techniques from the set of techniques corresponding to the correct tactic.

5. Results

The performance of both LLMs and ML models is evaluated in the task of mapping NIDS rules to MITRE ATT&CK techniques. Furthermore, we conduct a mapping to MITRE ATT&CK tactics to benchmark with the published work. Table 2 and Figure 5 summarize the performance of the tested LLMs (Gemini, Claude, and ChatGPT-4) in labeling NIDS rules with ATT&CK tactics and techniques across the different prompt template configurations, while Table 3 summarizes the precision, recall, and F1-score achieved by the ML models trained by the different LLMs on the same dataset. Furthermore, Table 4 provides the results of the LLMs in labeling rare techniques, where MLs are not applicable due to a lack of training data. The performance variations of the chosen baseline heuristics across the test set (see Table 2) and the rare techniques set (see Table 4) highlight the different underlying complexities of the tasks involved in labeling NIDS rules. The baselines demonstrate differing effectiveness depending on the structure and characteristics of the set tested on. Given this, the test results must be evaluated with these underlying complexities in mind.

Table 2. Performance of LLM models across different configurations.

Model	**Prompt Template**	**Techniques**			**Tactics**		
		Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Gemini	ICL_0	0.20	0.14	0.16	0.24	0.26	0.25
	ICL_1	0.34	0.24	0.28	0.34	0.39	0.36
	ICL_2	0.30	0.18	0.23	0.29	0.28	0.29
	T-ICL_0	0.41	0.28	0.33	0.38	0.45	0.41
	T-ICL_1	0.60	0.48	0.53	0.50	0.66	0.57
	T-ICL_2	0.55	0.48	0.48	0.46	0.60	0.52
Claude	ICL_0	0.29	0.40	0.34	0.39	0.57	0.46
	ICL_1	0.28	0.41	0.33	0.39	0.60	0.47
	ICL_2	0.31	0.42	0.36	0.42	0.64	0.51
	T-ICL_0	0.51	0.59	0.55	0.57	0.79	0.66
	T-ICL_1	0.57	0.60	0.59	0.59	**0.80**	0.68
	T-ICL_2	0.63	0.61	0.62	0.61	0.80	0.69
ChatGPT	ICL_0	0.45	0.27	0.34	0.50	0.45	0.47
	ICL_1	0.46	0.29	0.36	0.49	0.47	0.48
	ICL_2	0.35	0.30	0.32	0.44	0.50	0.47
	T-ICL_0	0.56	0.55	0.56	0.62	0.69	0.65
	T-ICL_1	0.66	0.58	0.62	0.62	0.69	0.65
	T-ICL_2	0.70	0.55	0.62	0.66	0.72	0.69
	Approach						
Baseline	Top-1	0.2	0.2	0.2	0.25	0.34	0.29
	Top-2	0.17	0.34	0.22	0.24	0.5	0.33
	RT-1	0.06	0.05	0.05	N/A	N/A	N/A
	RT-2	0.12	0.1	0.1	N/A	N/A	N/A

The presented methodologies significantly outperform the baselines across all prompt template configurations besides Gemini ICL_0. This shows the general feasibility of applying LLMs within the task of labeling NIDS rules with MITRE ATT&CK techniques and tactics.

The results demonstrate a clear hierarchy in model performance, with Claude and ChatGPT consistently outperforming Gemini across all tested configurations. Claude and ChatGPT exhibit superior performance, achieving higher precision and recall in their responses, regardless of the prompt template used. The highest F1-score for technique (0.62) and tactic labeling (0.69) is reached under the T-ICL_2 configuration, which provides example-driven contextualization and methodological guidance. While ChatGPT provides superior results with regard to precision, Claude has its strength in providing higher levels of recall.

Unlike Claude and ChatGPT, Gemini demonstrates the highest performance within the T-ICL_1 setup (with an F1-score of 0.53 for technique labeling and 0.57 for tactics labeling). Furthermore, Gemini shows the highest performance increase when combining contextual and methodological guidance.

Within our experiments, we reach significantly higher performance than in previous works, reaching a maximal F1-score of 0.32 using the same model of ChatGPT employed in this work as well as with ChatGPT-3.5, where an F1-score of 0.49 is reached [40]. While Daniel et al. [40] combine labels of different results, our workflow allows to reach more precise results through improved prompt design.

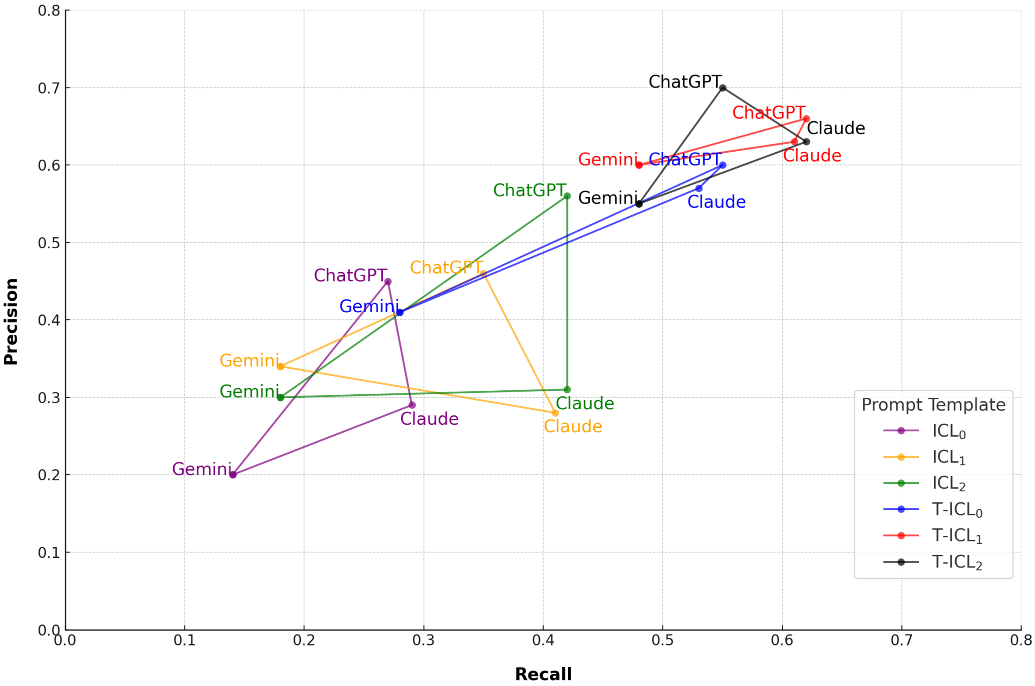


Figure 5. Precision and recall of LLM-based technique labeling across different configurations.

When comparing the different configurations of the prompt template, the contextual guidance shows a greater effect for the probability that the LLM selects the right label (the T-ICL_0 prompt template outperforms ICL_2). The superior performance with the T-ICL_2 prompt template demonstrates the value of providing both contextual information and methodological guidance (specifically ICL) within the prompts.

Table 4 shows the results attained by the LLMs (with their best-performing prompt template configuration as demonstrated using the test set; see Table 2) when labeling is performed on the rare techniques set. Gemini, Claude, and ChatGPT show comparable performance within their best performing configurations. Overall, the reached performance of the LLMs in labeling rare techniques is low. However, compared to the baselines computed, the LLMs show superior performance, proving their value for labeling rarely seen techniques (e.g., if labeled data are scarce). Gemini performs especially well in labeling tactics, suggesting that closely related techniques (techniques of the same tactic) are frequently proposed in the technique labeling task.

Within our tests, all ML models outperform the LLMs across all metrics. The SVM model trained by Gemini on the labeled dataset achieves the highest overall performance in technique labeling, with a precision of 0.88, recall of 0.87, and an F1-score of 0.87. The SVM model trained by Claude achieves an F1-score of 0.85, supported by a precision of 0.86 and recall of 0.85. Finally, SVM trained by ChatGPT performs worse than the other models, achieving an F1-score of 0.79, with the highest precision of 0.92 and the lowest recall of 0.70.

For tactic labeling, the models trained by Claude (F1-score of 0.91) and Gemini (F1-score of 0.92) show competitive results. The SVM trained by ChatGPT performs slightly worse, with an F1-score of 0.85.

Table 3. Performance of ML models developed by the different LLMs.

Developing Model	ML Model	Techniques			Tactics		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score
Gemini	SVM	0.88	0.87	0.87	0.91	0.92	0.92
Claude	SVM	0.86	0.85	0.85	0.91	0.92	0.91
ChatGPT	SVM	0.92	0.70	0.79	0.96	0.77	0.85

Table 4. Performance of LLMs in rare technique labeling.

Model	Prompt Template	Techniques			Tactics		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score
Gemini	T-ICL_1	0.22	0.21	0.22	0.29	0.38	0.33
Claude	T-ICL_2	0.23	0.18	0.20	0.26	0.29	0.27
ChatGPT	T-ICL_2	0.29	0.19	0.23	0.28	0.26	0.27
Baseline	Approach						
	Top-1	0.05	0.05	0.05	0.09	0.09	0.09
	Top-2	0.05	0.11	0.08	0.13	0.22	0.17
	RT-1	0.04	0.04	0.04	N/A	N/A	N/A
	RT-2	0.09	0.09	0.09	N/A	N/A	N/A

Overall, the ML models exhibit greater consistency and robustness compared to the LLMs. For labeling tactics, our approach achieves an F1-score of 0.92 (ML model trained and developed by Gemini), demonstrating comparable performance to previously published tools in labeling attack tactics [3]. The results indicate strong precision, recall, and F1-scores for labeling NIDS rules with tactics, providing comparable performance to existing ML-based methods for labeling NIDS rules with tactics [3]. Furthermore, the presented models provide a good basis for decreasing the alert explainability gap by providing highly precise tactic labels for NIDS rules.

6. Discussion

The results highlight clear differences between the LLM-based and ML-based approaches for mapping NIDS rules to MITRE ATT&CK, both in terms of performance metrics and potential practical applications. We reach comparable results as presented by the related work with regard to tactic labeling [3] while designing the workflow for technique labeling. Furthermore, we introduce the first work elaborating on the labeling of NIDS rules with MITRE ATT&Ck techniques and contribute a methodology for labeling based on LLMs and ML models.

6.1. LLM Performance and Insights

The LLMs demonstrate variable performance depending on the configuration and prompt design. Highlighting the need for efficient prompt engineering.

The results of our comparative evaluation of Claude, ChatGPT, and Gemini with regard to their capabilities in the context of NIDS rule labeling across multiple prompt configurations reveal significant performance discrepancies. Claude and ChatGPT consistently outperform Gemini in terms of recall and precision. Gemini exhibits the poorest performance in all configurations tested. These findings warrant further exploration of the factors influencing the observed performance variations. Furthermore, in future research, additional prompt design techniques, especially formal reasoning techniques, and additional LLMs should be tested to further boost the performance of LLMs for the proposed labeling task, allowing practical application within SOCs.

The performance results of Claude and ChatGPT reaching F1-scores greater than 0.6 underscores the utility of incorporating the use of LLMs for labeling techniques. Furthermore, with regard to the prompt templates, the superiority of the T-ICL_2 templates proves the benefits of example-driven prompts that guide the model toward task-specific reasoning. Yet, while ICL allows LLMs to leverage examples provided directly within the prompt to improve performance for a given task, it may not always be the most efficient strategy, particularly in resource-constrained environments. One key limitation of ICL is that it increases computational costs and token usage, especially when large examples or numerous prompt tokens are required. This results in higher compute costs and longer response times, which can be prohibitive in large-scale or real-time applications.

6.2. Ml Model Superiority

The ML models demonstrate superior performance across all datasets, achieving the highest F1-scores (0.87 for technique labeling and 0.92 for tactic labeling) when trained by Gemini. This result emphasizes the strength of supervised learning, particularly when trained on well-annotated datasets. The ML models' precision scores, such as 0.92 for the SVM model trained on ChatGPT in technique labeling and 0.96 in tactic labeling, are notably higher than those of the LLMs, reflecting their ability to minimize false positives effectively.

The strong performance of ML models can be attributed to their capacity to exploit structured feature representations, such as TF-IDF vectors, to identify patterns within the dataset. However, it is worth noting that the success of the ML approach heavily depends on the quality of the training data.

6.3. Comparative Implications

While LLMs offer flexibility and the ability to generate explanations for their predictions, their lower performance metrics suggest that they are at least currently better suited for tasks where interpretability and generalization are more important than raw accuracy. On the other hand, ML models excel in precision and recall but rely on high-quality labeled datasets and cannot inherently provide reasoning for their decisions. Yet, if no labeled datasets are available that the ML models can be trained on, LLMs can provide significant value in the early stages of generating labeled datasets, especially if the LLMs are used to assist human analysts by generating candidate labels (human-in-the-loop processes). LLMs can thereby take advantage of their ability to understand NIDS rules and generate labels especially in setups with partial information.

A hybrid approach could harness the strengths of both methods: LLMs could be employed for initial labeling and to provide contextual explanations, while ML models could refine these labels for deployment in high-stakes applications requiring high precision and recall. Future work could explore ensemble strategies or the integration of LLM-generated insights into ML feature engineering to further enhance performance.

6.4. LLM Driven Code Generation

A key aspect of this study is the use of LLMs to autonomously develop ML pipelines, a task traditionally requiring significant human expertise. OpenAI's Codex, as highlighted by Chen et al. [45], demonstrates the potential of LLMs trained on code to generate robust, domain-specific workflows. In our work, LLMs choose ML models, feature engineering, and hyperparameter tuning that give a 0.87 F1-score in technique labeling, further underlining their capability to handle complex cyber security tasks.

Code generation driven by LLMs has its own set of limitations. Wong et al. [46] discuss the challenges of generating secure and reliable code in AI-assisted programming, especially in scenarios involving sensitive domains like cyber security. Proper prompt engineering still plays a very important role in making sure the generated code fits the domain-specific requirements.

In general, LLM embeddings in cyber security workflows is a promising future research direction with respect to the continuous advancements of LLMs, for both automation and infrastructure development.

7. Conclusions

Within this study, we present a dataset comprising 973 labeled NIDS rules. The collected dataset serves as a useful resource for the cyber security research community, providing a ground truth for developing and testing new methods for mapping NIDS rules to MITRE ATT&CK techniques. This dataset can be utilized in future studies to benchmark model performance and explore new approaches to enhancing NIDS capabilities.

The experiments we conduct on the basis of this dataset underscore the potential of leveraging both LLMs and ML techniques to automate the labeling of NIDS rules with MITRE ATT&CK techniques and tactics. LLMs such as ChatGPT, Claude, and Gemini demonstrate their ability to provide contextual and explainable mappings, particularly when optimized through prompt engineering and contextual guidance. Among these, ChatGPT achieves the best overall performance, highlighting its potential to assist security analysts in tasks requiring contextual reasoning. However, their precision and recall scores, while promising, are consistently outperformed by supervised ML models trained on the same dataset.

The ML models, particularly SVM, showcase superior accuracy, benefiting from well-defined feature extraction and robust training datasets generated by the LLMs. The high precision and recall achieved by the ML models affirm their applicability for high-stakes scenarios where accuracy is critical. However, these models rely heavily on high-quality, pre-labeled datasets, a limitation that could restrict scalability to broader domains without significant initial manual effort.

A key contribution of this study is the collection of a large dataset of 973 Snort NIDS rules mapped to MITRE ATT&CK techniques and tactics. This dataset not only serves as a foundation for the evaluations conducted in this work but also provides a valuable resource for the cyber security research community. It can be leveraged for training, benchmarking, and further improving both LLM- and ML-based methods for CTI enrichment.

A key finding of this study is the complementary nature of LLMs and ML models. While LLMs provide a flexible and scalable framework for generating initial labels and explanations, ML models excel in refining these labels to achieve superior performance metrics. This synergy suggests the feasibility of a hybrid approach, where LLMs are used for initial data enrichment and add explainability, while ML models are used for final labeling.

Building on these findings, several directions can be pursued:

- **Prompt engineering:** Further prompt engineering techniques could be used to increase performance. Furthermore, the effect of improving contextual guidance by restricting the technique set to relevant techniques detectable with NIDS (e.g., selected based on DETT&CT) should be investigated.
- **Hybrid Approaches:** Future research could explore hybrid frameworks that integrate LLMs for initial rule labeling and ML models for fine-tuning. This approach may combine the scalability of LLMs with the precision of ML models.
- **Domain-Specific Fine-Tuning:** Fine-tuning LLMs with domain-specific datasets could improve their accuracy and reduce the need for extensive prompt engineering while increasing applicability. This would be particularly beneficial for complex domains

such as ICS and real-time SOC environments with computational constraints and critical response times.

- **Enhanced Feature Engineering:** ML models could benefit from incorporating additional features, such as contextual relationships within rules or temporal correlations, to further boost their labeling accuracy.
- **Cloud Versus Local:** Given the sensitivity of collected data within SOCs and the computational constraints and critical response time, it would be of especial interest to deploy high-quality local models to analyze the rules.
- **Evaluation on Diverse Datasets:** Expanding the evaluation to include datasets from different domains and attack scenarios will ensure the generalizability and robustness of the proposed methods.
- **Explainable AI:** While LLMs inherently provide explainability through natural language outputs, enhancing ML models with interpretable explanations could improve trust and adoption in operational environments.

In conclusion, this study demonstrates that the automated labeling of NIDS rules is both feasible and effective using a combination of LLMs and ML techniques. By addressing the challenges of scalability, accuracy, and domain adaptation, the proposed approaches can significantly enhance the capabilities of cyber security analysts in mitigating evolving threats.

Author Contributions: Conceptualization, N.D. and F.K.K.; methodology, N.D., F.K.K., S.G., S.S. (Sapir Sharabi), R.M. and S.S. (Shalev Shpolyansky); software, N.D., S.G., S.S. (Sapir Sharabi), R.M. and S.S. (Shalev Shpolyansky); validation, N.D. and F.K.K.; formal analysis, N.D., F.K.K., S.G., S.S. (Sapir Sharabi), R.M. and S.S. (Shalev Shpolyansky); investigation, N.D., F.K.K., S.G., S.S. (Sapir Sharabi), R.M. and S.S. (Shalev Shpolyansky); resources, N.D., F.K.K., R.P. and A.E.; data curation, N.D., S.G., S.S. (Sapir Sharabi), R.M. and S.S. (Shalev Shpolyansky); writing—original draft preparation, N.D. and F.K.K.; writing—review and editing, R.P., A.E. and A.M.; visualization, N.D. and F.K.K.; supervision, R.P. and A.E.; project administration, N.D., R.P. and A.E.; funding acquisition, R.P. and A.E. All authors have read and agreed to the published version of the manuscript.

Funding: A part of this research was funded by the U.S.-Israel Energy Center managed by the Israel-U.S. Binational Industrial Research and Development (BIRD) Foundation. A part of this research was funded by Fujitsu.

Data Availability Statement: The original data and code presented in the study are openly available in GitHub at https://github.com/NirDaniel/Labeling-NIDS-Rules-with-MITRE-ATT-CK-TTPs.

Conflicts of Interest: Author Florian Klaus Kaiser was employed by the company Agnostic Intelligence AG. Author Andres Murillo was employed by the company Fujitsu Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AT	Attack Type
AI	Artificial Intelligence
CTI	Cyber Threat Intelligence
DT	Decision Tree
GBM	Gradient Boosting Machine
GPTs	Generative Pre-trained Transformers
HIDS	Host-based Intrusion Detection Systems
ICL	In-Context-Learning

ICS	Industrial Control Systems
IDS	Intrusion Detection System
IoC	Indicator of Compromise
E	Explanation
EDR	Endpoint Detection and Response
KNN	K-Nearest Neighbor
LCA	Lowest Common Ancestor
LLM	Large Language Model
LLMs	Large Language Models
MCML-C	Multi-Class Multi-Label Classifier
ML	Machine Learning
ML-C	Multi-Label Classification
NER	Named Entity Recognition
NIDS	Network Intrusion Detection Systems
RF	Random Forest
SOCs	Security Operations Centers
SVM	Support Vector Machines
T	Technique
TA	Tactic
TH	Threat Hunting
TF	Term Frequence
TF-IDF	Term Frequence Inverse Document Frequency
TTPs	Tactics, Techniques, and Procedures

References

1. Chakrabarti, S.; Chakraborty, M.; Mukhopadhyay, I. Study of snort-based IDS. In Proceedings of the International Conference and Workshop on Emerging Trends in Technology, Almaty, Kazakhstan, 23 January 2010; pp. 43–47.
2. Elitzur, A.; Puzis, R.; Zilberman, P. Attack hypothesis generation. In Proceedings of the 2019 European Intelligence and Security Informatics Conference (EISIC), Warsaw, Poland, 15 September 2019; pp. 40–47.
3. Lin, S.X.; Li, Z.J.; Chen, T.Y.; Wu, D.J. Attack tactic labeling for cyber threat hunting. In Proceedings of the 2022 24th International Conference on Advanced Communication Technology (ICACT), Seoul, South Korea, 20 February 2022; pp. 34–39.
4. Bagui, S.S.; Mink, D.; Bagui, S.C.; Ghosh, T.; Plenkers, R.; McElroy, T.; Dulaney, S.; Shabanali, S. Introducing UWF-ZeekData22: A Comprehensive Network Traffic Dataset Based on the MITRE ATT&CK Framework. *Data* **2023**, *8*, 18. [CrossRef]
5. Sentonas, M. Crowdstrike Introduces Charlotte AI, Generative AI Security Analyst—Crowdstrike, 2023. Available online: https://www.crowdstrike.com/en-us/blog/crowdstrike-introduces-charlotte-ai-to-deliver-generative-ai-powered-cybersecurity/ (accessed 10 August 2024).
6. Törnberg, P. Chatgpt-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning. *arXiv* **2023**, arXiv:2304.06588.
7. Long, C.; Lowe, K.; dos Santos, A.; Zhang, J.; Alanazi, A.; O'Brien, D.; Wright, E.; Cote, D. Evaluating ChatGPT4 in Canadian Otolaryngology-Head and Neck Surgery Board Examination using the CVSA Model. *medRxiv* **2023**, 2023–05 .
8. Guerra, J.L.; Catania, C.; Veas, E. Datasets are not enough: Challenges in labeling network traffic. *Comput. Secur.* **2022**, *120*, 102810. [CrossRef]
9. Gjerstad, J.L. Generating Labelled Network Datasets of APT with the MITRE CALDERA Framework. Master's Thesis, University of Oslo, Oslo, Norway, 2022 .
10. Palacin, V. *Practical Threat Intelligence and Data-Driven Threat Hunting*; Packt Publishing: Birmingham, UK, 2021.
11. Chismon, D.; Ruks, M. *Threat Intelligence: Collecting, Analysing, Evaluating*; MWR Infosecurity Ltd.: Basingstoke, UK, 2015; Volume 3, pp. 36–42.
12. Haddad, A.; Aaraj, N.; Nakov, P.; Mare, S.F. Automated Mapping of CVE Vulnerability Records to MITRE CWE Weaknesses. *arXiv* **2023**, arXiv:2304.11130.
13. Daszczyszak, R.; Ellis, D.; Luke, S.; Whitley, S. *Ttp-Based Hunting*; Technical Report; Mitre Corp Mclean: McLean, VA, USA, 2019.
14. Kaiser, F.K.; Dardik, U.; Elitzur, A.; Zilberman, P.; Daniel, N.; Wiens, M.; Schultmann, F.; Elovici, Y.; Puzis, R. Attack Hypotheses Generation Based on Threat Intelligence Knowledge Graph. *IEEE Trans. Dependable Secur. Comput.* **2023**, *20*, 4793–4809. [CrossRef]

15. Liao, X.; Yuan, K.; Wang, X.; Li, Z.; Xing, L.; Beyah, R. Acing the IOC game: Toward automatic discovery and analysis of open-source cyber threat intelligence. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 755–766.
16. Strom, B.E.; Applebaum, A.; Miller, D.P.; Nickels, K.C.; Pennington, A.G.; Thomas, C.B. *MITRE ATT&CK®: Design and Philosophy*; The MITRE Corporation: Mc Lean, VA, USA, 2020.
17. Landauer, M.; Frank, M.; Skopik, F.; Hotwagner, W.; Wurzenberger, M.; Rauber, A. A framework for automatic labeling of log datasets from model-driven testbeds for HIDS evaluation. In Proceedings of the 2022 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems, Baltimore, MD, USA, 27 April 2022; pp. 77–86.
18. Othman, S.M.; Alsohybe, N.T.; Ba-Alwi, F.M.; Zahary, A.T. Survey on intrusion detection system types. *Int. J. Cyber-Secur. Digit. Forensics* **2018**, *7*, 444–463.
19. Peng, Y.; Wang, H. Design and implementation of network instruction detection system based on snort and NTOP. In Proceedings of the 2012 International Conference on Systems and Informatics (ICSAI2012), Beijing, China, 17–19 April 2012; pp. 116–120.
20. Khamphakdee, N.; Benjamas, N.; Saiyod, S. Improving intrusion detection system based on Snort rules for network probe attack detection. In Proceedings of the 2014 2nd International Conference on Information and Communication Technology (ICoICT), Berlin, Germany, 14–17 April 2014; pp. 69–74.
21. Motlagh, F.N.; Hajizadeh, M.; Majd, M.; Najafi, P.; Cheng, F.; Meinel, C. Large language models in cybersecurity: State-of-the-art. *arXiv* **2024**, arXiv:2402.00891.
22. Tod-Răileanu, G.; Axinte, S.D. ChatGPT-Information Security Overview. In Proceedings of the International Conference on Cybersecurity and Cybercrime, Porto, Portugal, 25–29 September 2023; Volume 10.
23. Liu, Y.; Tao, S.; Meng, W.; Yao, F.; Zhao, X.; Yang, H. Logprompt: Prompt engineering towards zero-shot and interpretable log analysis. In Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings, Hong Kong, China, 13–16 May 2024; pp. 364–365.
24. Ishibashi, R.; Miyamoto, K.; Han, C.; Ban, T.; Takahashi, T.; Takeuchi, J. Generating labeled training datasets towards unified network intrusion detection systems. *IEEE Access* **2022**, *10*, 53972–53986. [CrossRef]
25. Husari, G.; Al-Shaer, E.; Ahmed, M.; Chu, B.; Niu, X. Ttpdrill: Automatic and accurate extraction of threat actions from unstructured text of CTI sources. In Proceedings of the 33rd Annual Computer Security Applications, Orlando, FL, USA, 4–8 December 2017; pp. 103–115.
26. Legoy, V.; Caselli, M.; Seifert, C.; Peter, A. Automated retrieval of ATT&CK tactics and techniques for cyber threat reports. *arXiv* **2020**, arXiv:2004.14322.
27. Mendsaikhan, O.; Hasegawa, H.; Yamaguchi, Y.; Shimada, H. Automatic mapping of vulnerability information to adversary techniques. In Proceedings of the The Fourteenth International Conference on Emerging Security Information, Systems and Technologies SECUREWARE2020, Valencia, Spain, 21–25 November 2020.
28. Satvat, K.; Gjomemo, R.; Venkatakrishnan, V. Extractor: Extracting attack behavior from threat reports. In Proceedings of the 2021 IEEE European Symposium on Security and Privacy (EuroS&P), Venice, Italy, 30 June–4 July 2021; pp. 598–615.
29. You, Y.; Jiang, J.; Jiang, Z.; Yang, P.; Liu, B.; Feng, H.; Wang, X.; Li, N. TIM: Threat context-enhanced TTP intelligence mining on unstructured threat data. *Cybersecurity* **2022**, *5*, 3. [CrossRef]
30. Li, Z.; Zeng, J.; Chen, Y.; Liang, Z. AttacKG: Constructing technique knowledge graph from cyber threat intelligence reports. In Proceedings of the Computer Security–ESORICS 2022: 27th European Symposium on Research in Computer Security, Copenhagen, Denmark, September 26–30. 2022; Proceedings, Part I. Springer: Berlin/Heidelberg, Germany, 2022; pp. 589–609.
31. Rani, N.; Saha, B.; Maurya, V.; Shukla, S.K. TTPHunter: Automated Extraction of Actionable Intelligence as TTPs from Narrative Threat Reports. In Proceedings of the 2023 Australasian Computer Science Week, Melbourne, VIC, Australia, 30 January–3 February 2023; pp. 126–134.
32. Aghaei, E.; Niu, X.; Shadid, W.; Al-Shaer, E. Securebert: A domain-specific language model for cybersecurity. In Proceedings of the International Conference on Security and Privacy in Communication Systems, Virtual, 17–19 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 39–56.
33. Gabrys, R.; Bilinski, M.; Fugate, S.; Silva, D. Using Natural Language Processing Tools to Infer Adversary Techniques and Tactics Under the Mitre ATT&CK Framework. In Proceedings of the 2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 8–10 January 2024; pp. 0541–0547.
34. McPhee, M. *Methods to Employ Zeek in Detecting MITRE ATT&CK Techniques*; The MITRE Corporation: Mc Lean, VA, USA, 2020.
35. Arafune, M.; Rajalakshmi, S.; Jaldon, L.; Jadidi, Z.; Pal, S.; Foo, E.; Venkatachalam, N. Design and development of automated threat hunting in industrial control systems. In Proceedings of the 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), Las Vegas, NV, USA, 26–29 January 2022; pp. 618–623.
36. Garcia, S.; Valeros, V. Towards a better labeling process for network security datasets. *arXiv* **2023**, arXiv:2305.01337.

37. Masumi, K.; Han, C.; Ban, T.; Takahashi, T. Towards efficient labeling of network incident datasets using tcpreplay and snort. In Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy, Virtual, 26–28 April 2021; pp. 329–331.
38. Sharma, Y.; Birnbach, S.; Martinovic, I. RADAR: Effective Network-based Malware Detection based on the MITRE ATT&CK Framework. *arXiv* **2022**, arXiv:2212.03793.
39. Jüttner, V.; Grimmer, M.; Buchmann, E. ChatIDS: Advancing Explainable Cybersecurity Using Generative AI. *Int. J. Adv. Secur.* **2024**, *17*, 2.
40. Daniel, N.; Kaiser, F.K.; Dzega, A.; Elyashar, A.; Puzis, R. Labeling NIDS Rules with MITRE ATT &CK Techniques Using ChatGPT. In Proceedings of the European Symposium on Research in Computer Security, The Hague, The Netherlands, 25–29 September 2023; Springer: Berlin/Heidelberg, Germany; pp. 76–91.
41. Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; et al. Self-refine: Iterative refinement with self-feedback. *Adv. Neural Inf. Process. Syst.* **2024**, *36*.
42. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *Openai Blog* **2019**, *1*, 9.
43. Brown, T.B. Language models are few-shot learners. *arXiv* **2020**, arXiv:2005.14165.
44. Liu, J.; Shen, D.; Zhang, Y.; Dolan, B.; Carin, L.; Chen, W. What Makes Good In-Context Examples for GPT-3? *arXiv* **2021**, arXiv:2101.06804.
45. Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H.P.D.O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. Evaluating large language models trained on code. *arXiv* **2021**, arXiv:2107.03374.
46. Wong, M.F.; Guo, S.; Hang, C.N.; Ho, S.W.; Tan, C.W. Natural language generation and understanding of big code for AI-assisted programming: A review. *Entropy* **2023**, *25*, 888. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.