future internet

MDPI

Article

Classification of Hacker's Posts Based on Zero-Shot, Few-Shot, and Fine-Tuned LLMs in Environments with Constrained Resources

Theodoros Giannilias, Andreas Papadakis *, Nikolaos Nikolaou and Theodore Zahariadis

Synelixis Solutions S.A., Farmakidou 10, 34100 Chalkida, Greece; theodore.giannilias1994@gmail.com (T.G.); nikolaou@power-ops.com (N.N.); zahariad@synelixis.com (T.Z.)
* Correspondence: papadakis@synelixis.com

Abstract: This paper investigates, applies, and evaluates state-of-the-art Large Language Models (LLMs) for the classification of posts from a dark web hackers' forum into four cyber-security categories. The LLMs applied included Mistral-7B-Instruct-v0.2, Gemma-1.1-7B, Llama-3-8B-Instruct, and Llama-2-7B, with zero-shot learning, few-shot learning, and fine-tuning. The four cyber-security categories consisted of "Access Control and Management", "Availability Protection and Security by Design Mechanisms", "Software and Firmware Flaws", and "not relevant". The hackers' posts were also classified and labelled by a human cyber-security expert, allowing a detailed evaluation of the classification accuracy per each LLM and customization/learning method. We verified LLM fine-tuning as the most effective mechanism to enhance the accuracy and reliability of the classifications. The results include the methodology applied and the labelled hackers' posts dataset.

Keywords: large language model; mistral; gemma; llama; cybersecurity; dark web; text classification; few-shot learning; fine-tuning

check for updates

Academic Editors: Raha Moraffah, Li Yang and Paolo Bellavista

Received: 4 March 2025
Revised: 25 April 2025
Accepted: 29 April 2025
Published: 5 May 2025

Citation: Giannilias, T.; Papadakis, A.; Nikolaou, N.; Zahariadis, T. Classification of Hacker's Posts Based on Zero-Shot, Few-Shot, and Fine-Tuned LLMs in Environments with Constrained Resources. *Future Internet* **2025**, *17*, 207. https://doi.org/10.3390/fi17050207

1. Introduction

In the current interconnected digital landscape, cybersecurity threats have grown in volume and sophistication. Unregulated data sharing and informal communication channels empower malicious actors to rapidly spread harmful techniques that threaten system integrity. Identifying vulnerabilities and correlating them with attack patterns is critical for proactive cybersecurity management and defence. Artificial Intelligence (AI) and, more recently, Large Language Models (LLMs) have emerged as powerful tools for automating cybersecurity tasks, including phishing detection, malware analysis, and threat intelligence extraction [1,2]. These models are very effective in processing complex and unstructured textual data—like those found in hacker forums. The dark web is a valuable source of threat intelligence, offering insights into exploits and emerging attacks. However, its informal, coded, and unstructured nature presents unique challenges for automatic classification and analysis. Manual processing is impractical, and existing rule-based or traditional machine learning (ML) approaches often miss the nuance of texts coming from the dark web.

Deep net and dark web data, although useful to extract hints on emerging or even imminent threats, have two-fold challenges: (a) access to the content involving scraping, decryption and preprocessing [3] and (b) its analysis, which is to a large extent, automated, so that useful information, such as categorized threat indications, can be extracted. The analysis of the content, which is unstructured, informal (slang-based), and even codified, using AI and specifically LLM-based techniques, defines the aim of this work. Currently, efforts on proactive security, involving sources from the dark web and employing state-of-the-art AI tools, have been intensified, both from an academic and commercial perspective, involving white hats, government agencies, and independent labs. Specifically, AI and LLMs are increasingly used in cybercrime forums [4] to extract cyber threat intelligence and support detection [5], and there have been initial efforts to understand content from opaque and not easily accessible sources [6]. Additionally, there have been efforts based on the collection and consolidation of CTIs [7]. These approaches are aligned with legislative efforts, including NIS2 [8] and the resilience of critical entity directives [9] in the EU. Furthermore, the market currently offers real-time CTI info from the dark web, including the monitoring of attack surfaces, threat intelligence feeds, and insights into dark web activities.

The present research aims to enhance existing analyst-driven intelligence strategies with an open, LLM-based methodology. Specifically, the use of open-source LLMs is employed to categorize posts from hacker forums into four distinct cybersecurity vulnerability groups. The posts are collected by the University of Arizona [10], and they are of an informal and unstructured format, involving security and other thematic areas. In this context, zero- and few-shot, as well as parameter-efficient fine-tuning strategies (LoRA), are evaluated in scenarios with limited resources and labelled data. The LLM behaviour is investigated in small, high-quality datasets where data collection is constrained by domain specificity, sensitivity, or access. This integration enhances our ability to analyze and categorize complex informal communications more accurately.

Our methodology enables the scalable classification of such content into cybersecurity-relevant categories, supporting tasks like vulnerability tracking, trend mapping, or risk detection, contributing to the "Identify" function within the NIST cybersecurity framework [11], enhancing threat intelligence and improving the identification and categorization of cybersecurity threats. These advancements can yield actionable insights for future threat detection and prevention. Additionally, our work bridges cybersecurity and AI by advancing methodologies, which typically rely on pre-trained embedding models and cosine similarity metrics [12].

The main contributions of this work include the following:

- **The novel classification of dark web resources for proactive security**: We introduce a novel, automatic classification mechanism involving posts from dark web hackers into specific cybersecurity categories identified by a domain expert.
- **The evaluation of state-of-the-art models and fine-tuning methods:** This classification is performed using four state-of-the-art open LLMs, namely Mistral, Gemma, Llama2, and Llama3. We perform comparative and experimental assessments of the zero-shot, one-shot, three-shot, and fine-tuned versions of each LLM. The results demonstrate that fine-tuned models consistently outperform prompt-based approaches, even with minimal data.
- **The expert-annotated dataset:** Using cosine similarity to match posts with cybersecurity vulnerability descriptions, we curated a high-quality, expert-annotated dataset comprising 670 of the hacker's filtered posts.

The paper is organized as follows: Section 2 reviews the current state of the art, focusing on the application of LLMs in cybersecurity, prompt engineering techniques, and methods for customization and fine-tuning. Section 3 discusses the selection of LLMs and learning approaches. Section 4 presents the experimental results for four different LLM deployment strategies, including zero-, one-, three-shot and fine-tuning strategies, using a selection of carefully chosen models. Section 5 provides an aggregated discussion and comparative analysis of these findings, and Section 6 concludes with key takeaways and suggestions for future work.

2. Related Work

The increasing complexity of cyber threats and the fact that traditional rule-based and signature-based systems often fall short in detecting new threats have necessitated the adoption of AI-driven security solutions. The use of ML and Natural Language Processing (NLP) techniques has improved threat intelligence processing, malware analysis, and intrusion detection. However, challenges persist, particularly in managing the unstructured cybersecurity-related content from underground communities and hacker forums. LLMs are paving the way for innovative cybersecurity automation, particularly for in-text classification, threat intelligence, and vulnerability detection. Currently, their deployment in cybersecurity faces challenges, such as the limited availability of domain-specific labelled data and bounded or even constrained computational resources.

2.1. AI and Machine Learning in Cybersecurity

Traditional rule-based systems and signature-based detection techniques struggle to identify zero-day attacks, polymorphic malware, and changing cyberattack tactics [13–15]. AI-powered security solutions, on the other hand, showcase advancements in malware analysis, intrusion detection systems (IDSs), threat intelligence, and network anomaly detection [16]. ML models, like Support Vector Machines (SVMs), Random Forests, and Artificial Neural Networks (ANNs), have been widely used in cybersecurity for event classification, anomaly detection, and malicious activity identification [17]. Additionally, genetic algorithms have been explored to develop adaptive cybersecurity defences, which adjust dynamically to changing attack patterns. Despite these developments, traditional ML approaches often rely on manually engineered features and cannot be readily generalized in novel and unidentified attack types. Polymorphic and invisible malware can elude conventional antivirus signatures. The work in [18] proposes an automated signature extraction method that leverages ML to effectively detect malicious code variants, including those generated by mutation engines. Similarly, ref. [19] introduces Andromaly, an ML-based malware detection framework for Android smartphones that continuously monitors system data to identify unusual activity, which could indicate potential infection.

While AI-driven IDS solutions improve detection accuracy and reduce computing complexity, they still rely on structured datasets and often produce high false positive rates, especially when faced with adversarial or noisy inputs [17]. Traditional machine learning models—such as Decision Trees, Bayesian Networks, and Convolutional Neural Networks (CNNs)—have been widely applied in cybersecurity tasks, like network traffic monitoring, authentication, and anomaly detection [20]. However, these models typically require large, labelled datasets and may not effectively cope with unstructured and dynamic cyber threats, such as conversations on underground web groups.

Recent advances in deep learning have led to the exploration of Generative Adversarial Networks (GANs) and Recurrent Neural Networks (RNNs) for dynamic malware detection and attack simulation [21]. Additionally, adversarial ML, in which AI models are trained to detect subtle alterations in cyber threats, has gained attention as a defence mechanism against evasion attacks. However, these approaches still face challenges related to interpretability, scalability, and computational efficiency, which limit their adoption in resource-constrained environments.

AI-driven text classification extends beyond traditional applications like malware detection and intrusion prevention. By analyzing threat intelligence reports, system logs, and discussions on hacker forums, security experts can forecast emerging risks. Traditional text classification methods, such as TF-IDF, Naïve Bayes, and even NLP models like BERT [19], have been employed to analyze large-scale cybersecurity data. However, these approaches have challenges confronting informal, aggressive, and unstructured language.

The work in [22] proposes an AutoML-based framework that can automatically build, optimize, and deploy IDS without human intervention. It combines intelligent search, model selection, and tuning to create self-adaptive cybersecurity defences, aiming toward fully autonomous security systems that can respond to evolving threats.

2.2. Large Language Models (LLMs) in Cybersecurity

In contrast to traditional AI methods that frequently rely on structured threat intelligence and predefined attack signatures [23], LLMs are engineered to process unstructured textual data, such as threat reports, security advisories, and informal communications [2]. Identifying unusual patterns within vast amounts of network traffic and security records, LLMs play a crucial role in cyber threat detection [5,24]. They also support identifying zero-day threats by automating vulnerability assessments and exploit detection during malware analysis [1]. Furthermore, LLM-driven email classification models can effectively distinguish phishing attempts from legitimate emails [25]. Beyond detection, LLMs enhance cybersecurity policy compliance by automating risk evaluations and generating security protocols tailored to corporate environments [24].

Researchers have introduced a range of benchmarking tools to assess LLM performance in cybersecurity applications. Domain-specific benchmarks, such as CyberBench and CyberInstruct, evaluate LLMs' capabilities in automated code analysis, threat intelligence processing, and Named Entity Recognition (NER) [2]. Notably, CyberInstruct integrates instruction-based fine-tuning and has shown performance comparable to closed models like GPT-4. In addition, the CyberMetric dataset—developed using Retrieval-Augmented Generation (RAG) and consisting of over 10,000 cybersecurity-related questions—evaluates the broader cybersecurity expertise of LLMs [1]. A review of 25 cutting-edge LLMs shows that larger models, like GPT-4 and Mixtral, consistently outperform smaller ones in domain-specific cybersecurity tasks [26]. These benchmarks emphasize both the potential and limitations of LLMs in cybersecurity, providing standardized methods to evaluate their effectiveness. When it comes to cybersecurity, general-purpose LLMs have limitations stemming from the fact that they are pre-trained on broad internet data rather than specialized cybersecurity texts, resulting in a low comprehension of domain-specific terminology. Consequently, fine-tuning with cybersecurity-focused datasets is essential to mitigate issues such as hallucinations or insufficient threat assessments.

While LLMs can automate tasks, such as social engineering attacks, penetration testing, and vulnerability detection, they are not without risks. Attackers can manipulate LLM-generated outputs to bypass security measures, raising concerns about their reliability in high-stakes cybersecurity applications. Additionally, scalability and computational costs remain significant obstacles to broader adoption. Many state-of-the-art models, like GPT-4 and Mixtral, demand extensive GPU resources, making them less accessible for organizations with limited computational resources. This situation underscores the need for open-source, streamlined models that are specifically optimized for cybersecurity tasks [1].

The work in [27] investigates using LLMs to generate counterfactual examples in a zero-shot manner to evaluate NLP models more thoroughly. By prompting LLMs without fine-tuning, they create diverse, label-consistent variations that expose model weaknesses. The obtained results show that LLM-generated counterfactuals are effective for stress testing and improving model robustness.

2.3. LLMs for Cybersecurity Text Classification and Threat Intelligence

Threat intelligence, malware descriptions, phishing content, and hacker forum discussions are types of text that must be categorized/classified as an initial step in risk assessment. Traditional approaches, like rule-based classifiers and SVMs, are less effective at processing the vast, unstructured, and dynamic nature of cybersecurity texts because they require extensive manual feature engineering and domain-specific tuning [28].

LLMs have proven effective in addressing various cybersecurity text categorization challenges. For instance, they enhance phishing detection by identifying fraudulent URLs and misleading emails [2]. They also facilitate the processing of diverse security-related content, including threat intelligence, vulnerability reports, security warnings, and dark web communications [28,29]. Additionally, LLMs provide support in classifying malware and exploits by analyzing malware signatures and cyberattack methods [1]. They also contribute to anomaly detection by uncovering unusual patterns in network logs and HTTP requests [2].

According to the CyberBench benchmark, optimized LLMs significantly outperform traditional methods in cybersecurity-specific classification tasks [2]. Furthermore, incorporating RAG enhances LLM accuracy by enabling more context-aware threat insights. To further address existing challenges, recent studies have explored domain-specific fine-tuning, model compression techniques, and hybrid AI systems that integrate LLMs with conventional cybersecurity frameworks [30].

Fine-tuned lightweight models have demonstrated high efficiency with minimal computational overhead, making them practical for real-world applications. As large-scale LLMs demand substantial computational resources, smaller, optimized models that maintain high accuracy are becoming more popular. Techniques such as parameter-efficient fine-tuning (PEFT) and low-rank adaptation (LoRA) further reduce hardware requirements by eliminating the need for full-scale model retraining [30].

The work in [31] proposes a method where LLMs are augmented with external tools (like web searches, APIs, and databases) to improve open-source intelligence (OSINT) gathering. Instead of relying only on the LLM's internal knowledge, they let the model decide when and how to use tools to retrieve fresh, accurate, and verifiable information. Their method shows that tool-augmented LLMs can outperform standard LLMs in complex, real-world intelligence tasks.

3. Materials and Methods

3.1. Selection of LLMs

The selected LLMs are up to date, of an adequate level of sophistication, and have comparable architectural complexity (as measured by the number of parameters) to ensure meaningful comparisons. Additionally, models are open source, with weights available under an open licence to promote repeatability and transparency. The selection is also guided by computational resource constraints and insights from previous studies [26].

Selected LLMs include Llama3-8b-Instruct, Mistral-7b-Instruct-v0.2, Gemma-1.1-7b, and Llama-2-7b-Instruct. These open-source models are largely comparable in terms of complexity (with Llama3 having the highest parameter count), as depicted in Table 1. The inclusion of Llama2 and Llama3 enables a direct comparison of consecutive versions of the same core design, showcasing their evolutionary improvements. Although the Mixtral-8x7B-Instruct was initially considered as a baseline model [26], it was excluded due to hardware limitations; its 56 billion parameters require roughly 128 GB of GPU memory, which exceeds the capacity of our NVIDIA A4500 GPU (20 GB VRAM).

Table 1. Characteristics of the selected LLMs.

Name (Origin)	Min. Gpu RAM	Num of Parameters	Vocab Size
Mistral-7b-v0.2	16 GB	7.3B	32K tokens
Gemma	14 GB	6.8B	64K tokens
Llama2-7b	14 GB	7B	32K tokens
Llama3-8b	16 GB	8B	128K tokens

To address the GPU VRAM constraints, the models were deployed in a quantized format using a 4-bit quantization technique. Specifically, we applied the NF4 quantization method alongside the 'BitsAndBytesConfig' library set to 4-bit precision, ensuring both scalability and efficiency. Additionally, to reduce the computational cost, we disabled the double quantization option and carried out computations using the half-precision floating point format.

3.2. Zero-Shot, Few-Shot, and Fine-Tuning

Although LLMs are primarily trained on general web content, they can adapt to specific tasks by leveraging their broad knowledge in new contexts [29]. This practice, known as zero-shot learning, relies on carefully crafted prompts that guide the model to perform tasks without any additional labelled training data. In this approach, the model generates output based solely on its understanding of the prompt and its pre-existing knowledge.

To further enhance their performance, especially when adapting to new contexts or tasks, these models can be customized. However, even in their quantized form, LLMs consist of billions of parameters, making full retraining highly resource-intensive. Consequently, techniques like prompt engineering and fine-tuning are often used as more efficient alternatives.

According to [32], a range of prompt engineering techniques has been developed to address diverse tasks and applications. These methods aim to enable models to handle new tasks without extensive retraining, improve reasoning and logical capabilities, reduce hallucination, enhance user interactions, support knowledge-based reasoning, better understand user intent, and foster metacognitive processes and self-reflection. For example, few-shot learning—a simple yet effective prompt engineering approach—provides context through exemplary questions and responses, guiding the model in generating appropriate outputs.

In addition to prompt engineering, model fine-tuning can refine performance by adjusting only a subset of the LLM's parameters, using pre-existing (possibly human-generated) input. Techniques such as PEFT optimize this process through selective, additive, re-parameterized, and hybrid fine-tuning strategies [33].

3.3. Dataset and Categories

3.3.1. Hacker's Posts Dataset

The Arizona State University dataset includes an extensive collection of data gathered from various dark web sources, including forums, IRC chats, marketplaces, and phishing websites [1]. The dataset contains approximately 500K posts, covering thematic areas related to illegal activities, including drug trafficking, weapons sales, stolen data, and hacking tools. The posts vary significantly in length and format, but they share a common characteristic: an informal style of communication.

A subset has been selected for further classification based on each post's thematic relevance to the identified vulnerabilities. Specifically, Google's Universal Sentence Encoder (USE) [34] was used to calculate the cosine similarity between each post and the vulnerability descriptions. Posts exceeding a similarity score of 0.75 were included, resulting in a refined subset of 670 posts.

As the original form of the dataset consisted of raw IRC-like chat logs, including timestamps, user handles, connection metadata and interleaved system messages, preprocessing was needed. Specifically, a custom cleaning pipeline was developed to extract only substantive, human-authored messages from these logs. Specifically, regular expressions have been utilized to perform the following:

- Remove timestamps and nicknames (e.g., 15:12 <usename>).
- Eliminate system noise and like join/leave messages (e.g., !—user joined/quit the chat).
- Remove duplicated messages, chat clutter, or channel announcements.
- Concatenate multi-line posts, where appropriate, to reconstruct complete sentences/discussions.

The interventions have been as minimal as possible so that the format of each post is not altered and, more importantly, its meaning is not distorted.

3.3.2. Selection of Categories and Post Categorization

The cybersecurity vulnerabilities considered, as detailed in [35], focus on critical infrastructures and draw upon the following resources: (a) the NISTIR [36] hierarchical structure enriched with vulnerabilities from NESCOR electric sector failure scenarios, (b) the Common Vulnerabilities and Exposures (CVEs) Dictionary of publicly known vulnerabilities, (c) the National Vulnerability Database, and the (d) ISO/IEC 27005 risk management standard [37]. These vulnerabilities have different levels of granularity, informational depth, and levels of specificity. Vulnerability description examples include the "System relies on communications that are easy to jam", "insufficient maintenance", "Weaknesses in Authentication Process or Authentication Keys", and "Use of weak SSL cipher and other cryptographic design flaws".

To establish a practical and manageable set of categories, we initially defined a broader set of ten categories, as depicted in Table 2. The initial expert-based classification of posts produced uneven results, and early experiments proved challenging for LLMs to handle this level of granularity (as presented in Table 2).

Table 2. The initial set of vulnerability categories.

Vulnerability Categories	Num of Distinct Posts	% of Distinct Posts
SW and firmware flaws	151	21.51%
Availability protection and security by design mechanisms	228	32.48%
Access control and management	113	16.10%
Not relevant	123	17.52%
Data integrity mechanisms	18	2.56%
Patch management	12	1.71%
Security awareness and training	13	1.85%
Data encryption at rest, in motion, and in use	14	1.99%
Misconfiguration	14	1.99%
Governance, ICT security policies, and procedures	11	1.57%
Security information/log management and monitoring	10	1.42%
Total	**702**	**100%**

Considering the above, we focused on the first three vulnerability types, namely "Software and firmware flaws", "Availability protection and security by design mechanisms", and "Access control and management", which together account for most posts (approximately 70%). Including the "Not relevant" category brings the total coverage to approximately 88%. Consequently, our classification relies on these four categories.

As presented in Figure 1, each selected post is categorized by the LLMs employing zero-shot, one-shot and three-shot prompt engineering, as well as fine-tuned LLM variants. The resulting inferences are then compared to those provided by a human expert.

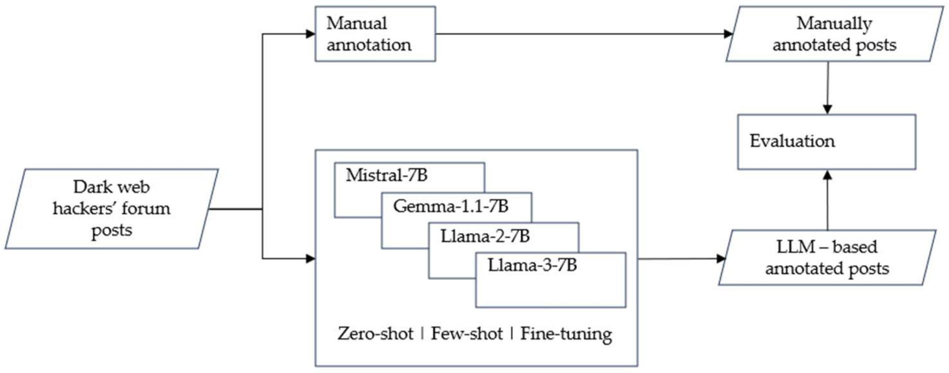


Figure 1. The steps performed.

The manual assignment of posts to the categories (labelling) was performed empirically through the gradual filtering of possible categories. Mapping posts to the categories has not always been straightforward. Criteria included the content, the keywords included (as mapped according to dark web slang) and most importantly, the identification of the intent of the writer. The concept of confidence level was included in our experimentation, but it proved that it did not improve the accuracy of the LLMs. Proof by contradiction logic was also employed by an expert if assigning a post to other categories led to contradictions.

The experiments, including both training and inference, were conducted on a predefined computational setup with limited resources, specifically using an NVIDIA A4500 GPU (manufactured by TSMC, Hsinchu, Taiwan) of 20 GB VRAM.

4. Results

In the following subsections, we present the accuracy, precision, recall, and f1-scores for each of the four LLMs (Mistral-7B-Instruct-v0.2, Meta-Llama3-8b-Instruct, Google-Gemma-1.1-7b, and Meta-Llama2-7b), across four approaches of prompt engineering: zero-shot, one-shot, three-shot, and fine-tuning strategies.

4.1. Zero-Shot Approach

In the zero-shot approach, each LLM was prompted to classify the selected posts in the four cybersecurity-related categories without any prior examples or additional contextual information. To ensure consistent and reliable classification, we employed a precise and specific prompt template. In line with core prompt engineering principles—specificity, clarity, and structured outputs—the template was carefully designed to guide the model effectively [38].

The final prompt template emerged from an iterative refinement process driven by systematic experimentation and trial and error:

You are a sentence classifier. You will receive posts from hacker forums and categorize them into one of the following categories: 'Access control and management', 'Availability *protection and Security by design mechanisms', 'Software and Firmware Flaws', 'Not relevant'. Just generate the category of the following sentence, without anything else.*

The prompt was crafted according to established prompt engineering best practices, including structured input–output formatting [39], clearly defined goals, and the removal of extraneous information. To reduce ambiguity, the prompt explicitly states the following:

- The classification objective.
- The exact category labels.
- The expected response format ("just generate the category without anything else").

Additionally, the classification task was decomposed into a single, well-defined operation to optimize the model's reasoning efficiency. By clearly defining the scope of the task (i.e., classification and not text generation), this approach minimized cognitive load and simplified the generation of structured and deterministic outputs.

The accuracy, precision, recall and f1-score per LLM, employing zero-shot learning, are presented in Figures 2–5.

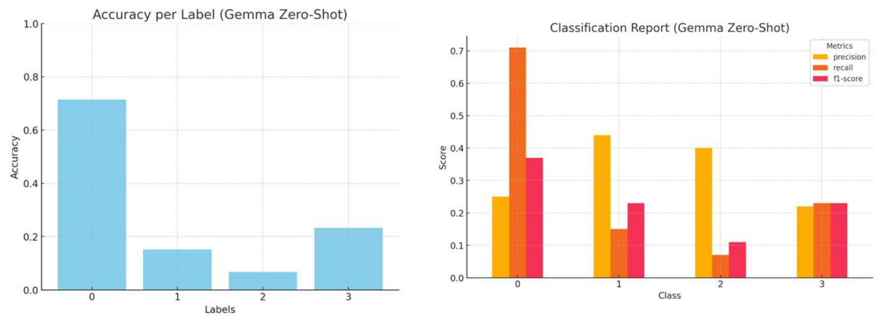


Figure 2. Accuracy by label and a comprehensive classification report (precision, recall, and f1-score) for Gemma1.1 zero-shot inference.

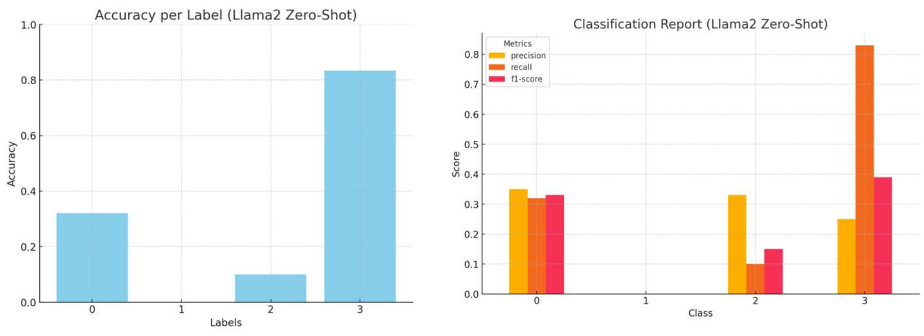


Figure 3. Accuracy by label and a comprehensive classification report (precision, recall, and f1-score) for Llama2 zero-shot inference.

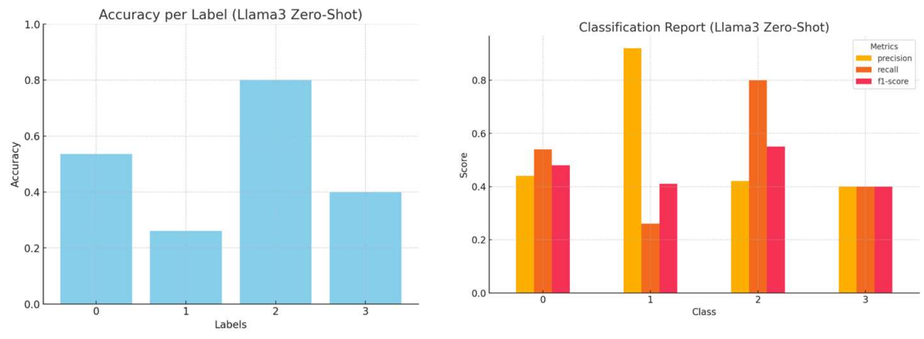


Figure 4. Accuracy by label and a comprehensive classification report (precision, recall, and f1-score) for Llama3 zero-shot inference.

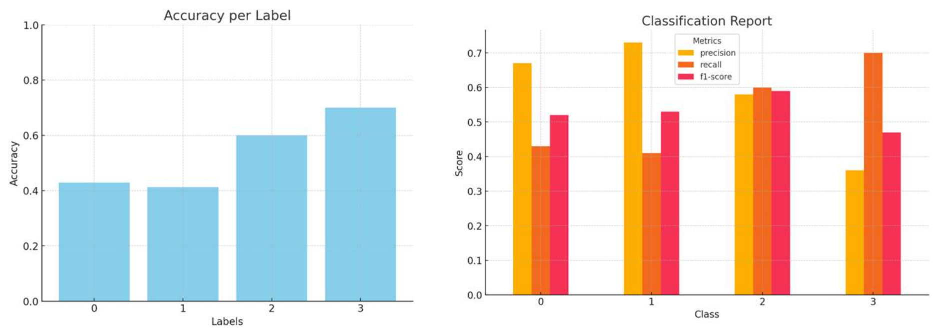


Figure 5. Accuracy by label and a comprehensive classification report (precision, recall, and f1-score) for Mistral zero-shot inference.

Although Mistral demonstrates higher overall classification accuracy, there is an asymmetry in the accuracy achieved per category in all models when starting from their initial state.

4.2. One Shot

In the one-shot classification approach, we augmented the prompt by providing a single representative example for each of the cybersecurity categories. A cybersecurity expert carefully selected these examples to ensure that they were effectively capturing the essence of each category while maintaining clarity, specificity, and relevance. Each example consists of a concise and unambiguous hacker forum post alongside its correct classification label. By incorporating such labelled examples directly into the prompt, the model gains a clearer grasp of how inputs map to outputs, enhancing its ability to generalize from minimal data. Unlike zero-shot classification, which relies solely on pre-trained knowledge, the one-shot approach utilizes in-context learning, providing a concrete reference point for decision-making.

To maintain consistency in data formatting, each selected example was appended to the prompt using a structured pipeline. The core instruction template remained largely the same as the zero-shot scenario, with the addition of examples to guide the classification process. The examples reflect the diverse and often unconventional nature of hacker forum posts, offering concrete references for the model's decision-making. They include the following points:

- **Access Control and Management:** "& don't forget to google [password recovered] to find passwords captured by keyloggers".
- **Availability Protection and Security by Design Mechanisms:** "A ddos required more than one person/computer for attacking a website".
- **Software and Firmware Flaws:** "For example, cracking drm of popular window games, especially new releases, then packaging your malware inside".
- **Not Relevant:** "I believe they are using a method related to traffic shaping on a world scale".

By integrating these labelled examples within the prompt, the model can derive classification patterns from a single reference, creating a more structured report and minimizing ambiguity during decision-making. The accuracy, precision, recall, and f1-score per LLM, employing the one-shot strategy, are presented in Figures 6–9.

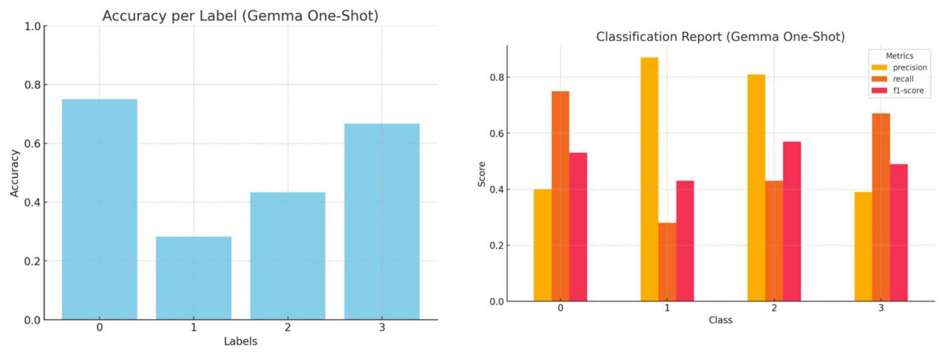


Figure 6. Accuracy by label and a comprehensive classification report (precision, recall, and f1-score) for Gemma1.1 one-shot inference.

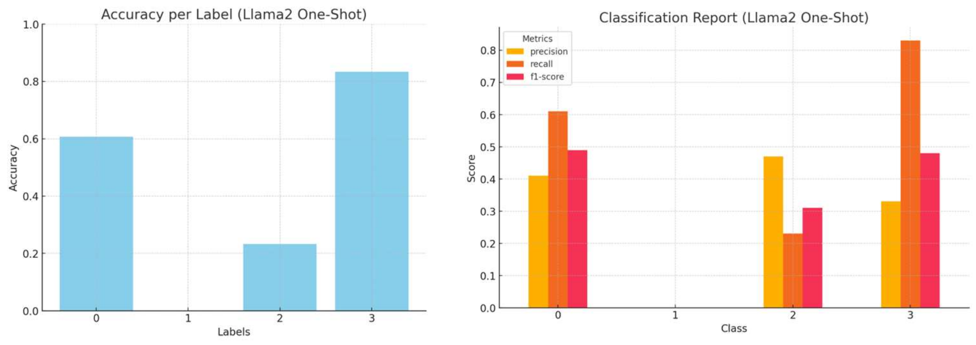


Figure 7. Accuracy by label and a comprehensive classification report (precision, recall, and f1-score) for Llama2 one-shot inference.

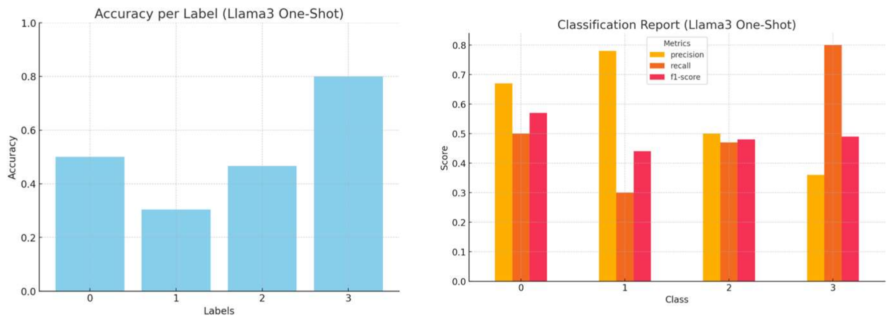


Figure 8. Accuracy by label and a comprehensive classification report (precision, recall, and f1-score) for Llama3 one-shot inference.

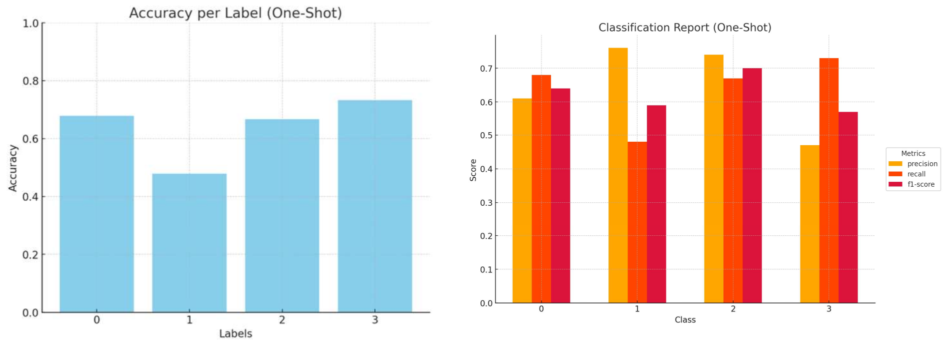


Figure 9. Accuracy by label and a comprehensive classification report (precision, recall, and f1-score) for Mistral one-shot inference.

4.3. Three Shot

The three-shot approach extends in-context learning by including three representative examples per category in the prompt. This is particularly beneficial when dealing with ambiguous phrasing or specialized jargon in hacker forum posts. As in the one-shot approach, a cybersecurity expert selected the examples to ensure that they accurately represented each category while maintaining specificity and clarity [40,41]. By expanding the number of examples, the model develops a stronger contextual understanding and achieves greater classification accuracy without relying on external retrieval or fine-tuning. The accuracy, precision, recall, and f1-score per LLM, employing the three-shot approach, are presented in Figures 10–13.

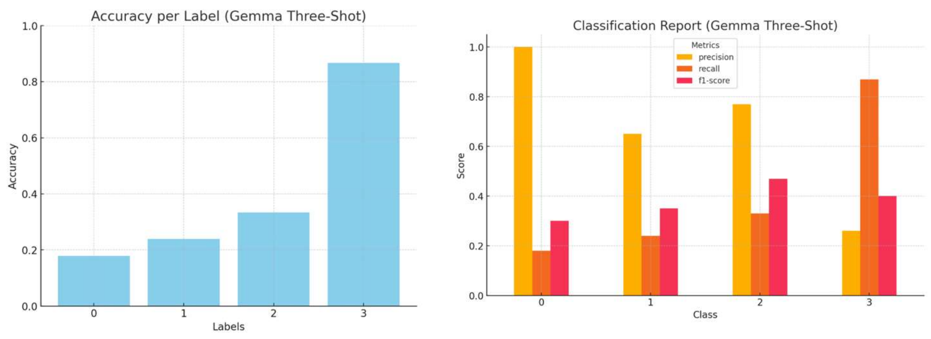


Figure 10. Accuracy by label and a comprehensive classification report (precision, recall, and f1-score) for Gemma1.1 three-shot inference.

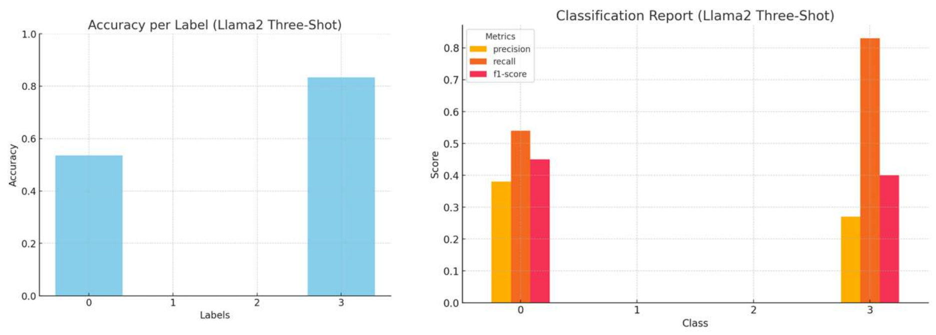


Figure 11. Accuracy by label and a comprehensive classification report (precision, recall, and f1-score) for Llama2 three-shot inference.

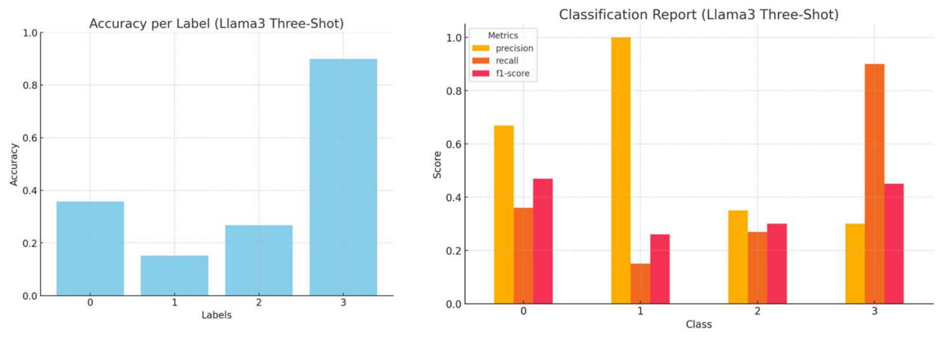


Figure 12. Accuracy by label and a comprehensive classification report (precision, recall, and f1-score) for Llama3 three-shot inference.

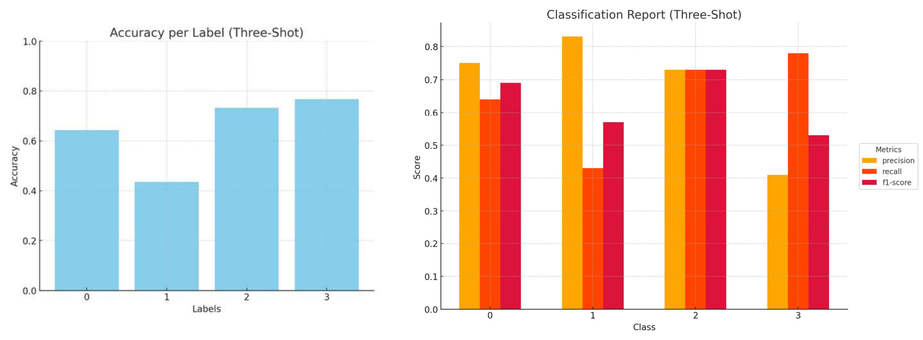


Figure 13. Accuracy by label and a comprehensive classification report (precision, recall, and f1-score) for Mistral three-shot inference.

4.4. Fine-Tuning with LoRA

The LLMs were fine-tuned using LoRA [42] optimizing task-specific parameters while preserving the pre-trained model weights. All models utilized 4-bit NF4 quantization for memory efficiency and exhibited optimal performance under the following LoRA settings:

- lora_alpha = 64 (optimal scaling factor);
- lora_dropout = 0.1 (preventing overfitting);
- r = 8 (best rank of updated matrices);
- Bias = "none";
- task_type = "CAUSAL_LM".

Each model was further tailored for its architecture, including specific target LoRA modules:

- **Mistral-7B-Instruct:** Applied standard LoRA configurations for efficient adaptation.
- **Llama-2-7B:** Utilized "all-linear" target modules for broader weight adaptation.
- **Llama-3-8B:** Standard LoRA adaptation was applied for causal language modelling.
- **Gemma-1.1-7B:** Fine-tuned with target modules, ["q_proj", "k_proj", "v_proj", "o_proj", "gate_proj", "up_proj", "down_proj"], ensuring more precise control over weight updates.

Fine-tuning was performed using the Supervised Fine-Tuning Trainer (SFTTrainer) with dynamically generated, task-specific prompts. This approach ensured a fair comparison across the models by preserving consistent domain-specific classification performance and leveraging each model's architectural strengths.

The accuracy, precision, recall and f1-score per LLM, employing the three-shot approach, are presented in Figures 14–17.

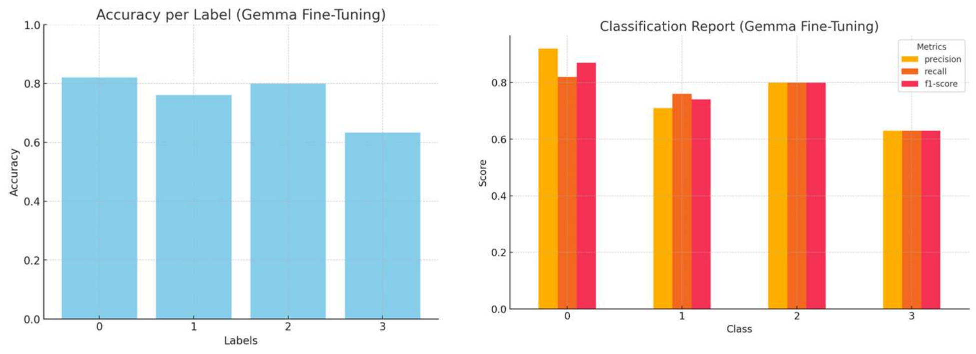


Figure 14. Accuracy by label and a comprehensive classification report (precision, recall, and f1-score) for fine-tuned Gemma1.1 inference.

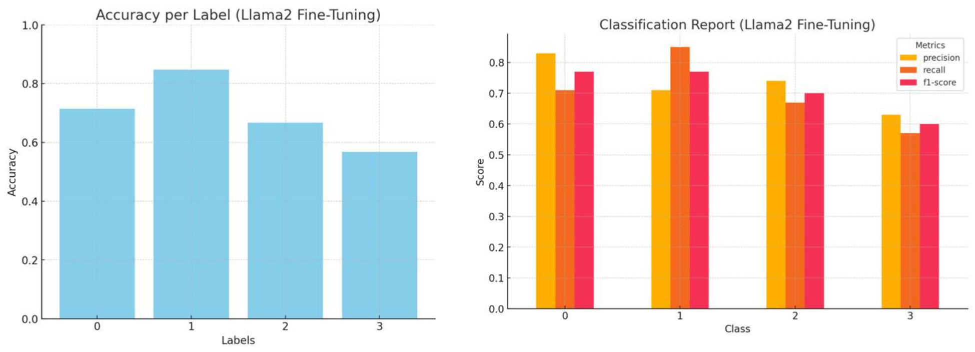


Figure 15. Accuracy by label and a comprehensive classification report (precision, recall, and f1-score) for fine-tuned Llama2 inference.

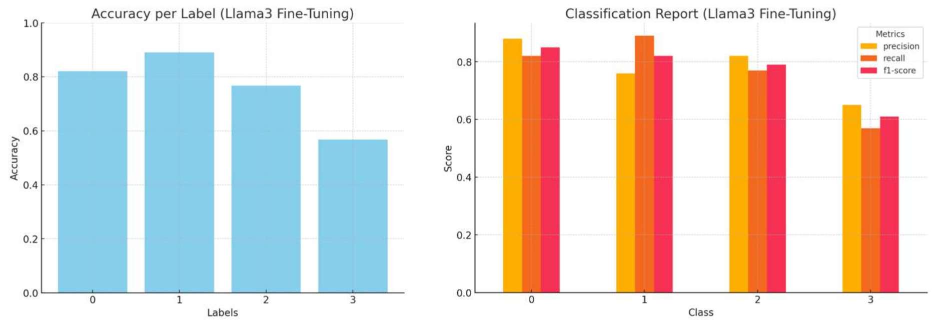


Figure 16. Accuracy by label and a comprehensive classification report (precision, recall, and f1-score) for fine-tuned Llama3 inference.

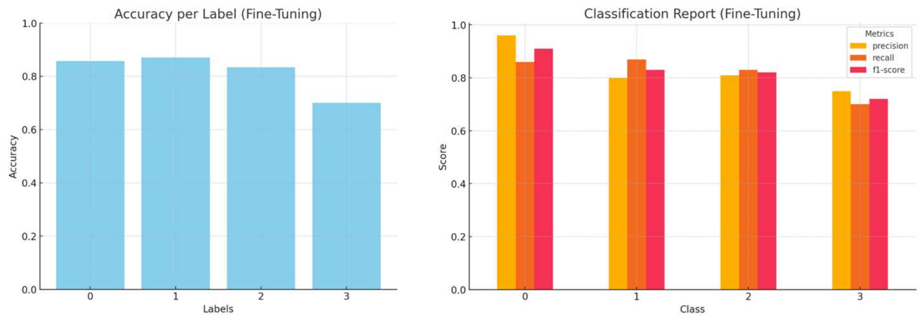


Figure 17. Accuracy by label and a comprehensive classification report (precision, recall, and f1-score) for fine-tuned Mistral inference.

For all vulnerability categories, the results are smoother in the case of fine-tuning and accuracy fluctuations are flattened with continued training. Furthermore, Table 3 includes an estimate of the training time for fine-tuning using LoRA on our 20 GB VRAM setup. These figures confirm the lightweight nature of the tuning process. We also noted the typical inference times per post for the zero-shot and few-shot runs.

Table 3. Time estimates for inference (zero/few-shot) and training (LoRA).

Model Name	Zero-Shot (Inf)	One-Shot (Inf)	Three-Shot (Inf)	Fine-Tunning (Train/Test)
Mistral-7B	0.41 s	0.59 s	0.77 s	~7.97 min/0.57 s
LLaMA-2-7B	0.48 s	0.66 s	0.85 s	~8.75 min/0.64 s
LLaMA-3-8B	0.53 s	0.74 s	0.96 s	~9.96 min/0.72 s
Gemma-1.1-7b	0.50 s	0.70 s	0.91 s	~9.17 min/0.66 s

4.5. Generalization

To assess the applicability and generalizability of the proposed approach across different domains, we applied our methodology to the GoEmotions dataset [43,44]. This dataset comprises Reddit comments, which are manually annotated with 27 distinct emotion labels. Similarly to the hacker's posts dataset, GoEmotions contains noisy, user-generated content with heterogeneous lengths and formats. To simulate comparable conditions, we selected a subset of the GoEmotions dataset [44] comprising four emotion categories—joy, anger, sadness, and neutral—and ensured a balanced distribution of samples across these classes. For our generalization experiments, we employed the Mistral LLM, which previously demonstrated the highest accuracy in the hacker's posts classification task.

We investigated two experimental settings: (a) prompts constructed using examples that are representative of each target category and (b) prompts constructed using randomly selected examples. Model performance in both settings was evaluated based on the accuracy metrics reported in Table 4.

Table 4. Comparison of accuracy between GoEmotions and hacker's posts using Mistral.

Model Name	GoEmotions Dataset	Hacker's Posts Dataset
Zero-Shot	0.65 s	0.52 s
One-Shot	0.70 s (0.61 s if randomly selected)	0.62 s (0.57 s if randomly selected)
Three-shot	0.72 s (0.63 s if randomly selected)	0.62 s (0.58 s if randomly selected)
Fine-Tunning	0.79 s	0.82 s

As shown in Table 4, the accuracy achieved by the proposed method in the emotion recognition domain exhibits a pattern consistent with that observed in the hacker's posts experiment. More specifically, fine-tuning yielded the highest performance, followed by three-shot prompting, which slightly outperformed the one-shot approach, both of which surpassed zero-shot performance. When prompt examples are selected randomly, accuracy declines across all configurations, falling below the few-shot and zero-shot baseline in both domains. These observations provide a strong indication of the applicability and robustness of the proposed methodology across diverse domains and varying levels of data noise.

5. Discussion

The heterogeneity of the posts and their informal language, coupled with the specialized nature of the categories, makes classification especially challenging. This difficulty is amplified by the need to discern the author's intent behind each post, i.e., whether the post indicates preparation for an attack or merely discusses a malicious threat.

Figure 18 illustrates how the LLM customization and training methods affect classification accuracy, consolidating the findings from Section 3. In general, few-shot learning demonstrates superior performance compared to zero-shot approaches, and fine-tuning outperforms all prompt engineering techniques.

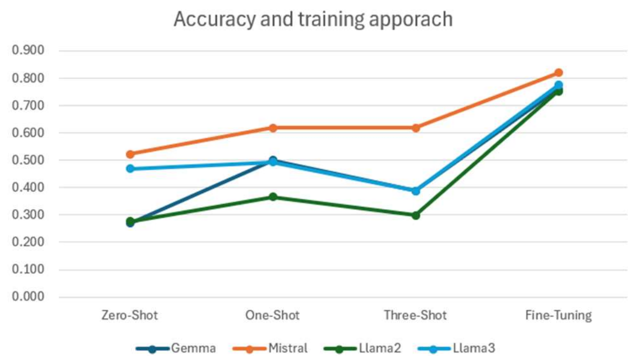
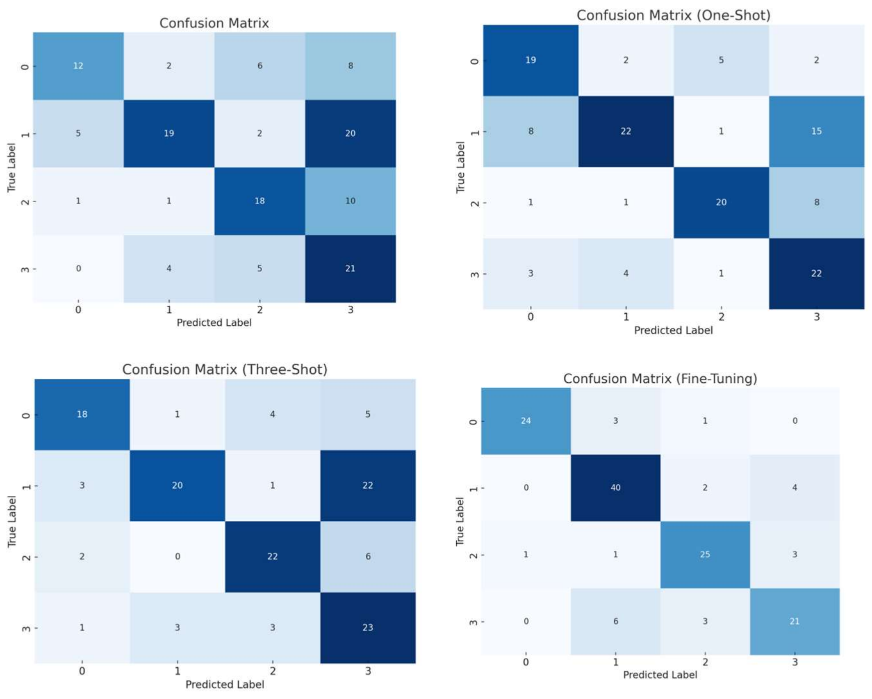


Figure 18. Accuracy per customization technique (considering different LLMs).

The improvement in accuracy is further evidenced in the confusion matrices, as depicted in Figure 19, for the Mistral LLM, where the darker the colour, the better the accuracy.



True Label \ Predicted Label (Confusion Matrix)	0	1	2	3
0	12	2	6	8
1	5	19	2	20
2	1	1	18	10
3	0	4	5	21

True Label \ Predicted Label (Confusion Matrix (One-Shot))	0	1	2	3
0	19	2	5	2
1	8	22	1	15
2	1	1	20	8
3	3	4	1	22

True Label \ Predicted Label (Confusion Matrix (Three-Shot))	0	1	2	3
0	18	1	4	5
1	3	20	1	22
2	2	0	22	6
3	1	3	3	23

True Label \ Predicted Label (Confusion Matrix (Fine-Tuning))	0	1	2	3
0	24	3	1	0
1	0	40	2	4
2	1	1	25	3
3	0	6	3	21

Figure 19. Accuracy for zero-shot, one-shot, three-shot, and fine-tuning approaches for Mistral LLM.

By providing one example per category, the one-shot prompt offers a minimal yet effective demonstration of the classification task. This approach enhances the LLM's ability to produce accurate classifications by fostering a nuanced understanding of the context. Although having more than one sample per category might seem beneficial for the context, the results sometimes indicate decreased accuracy. This can occur because LLMs become overwhelmed with too many instances, leading to confusion or misalignment as they attempt to identify the most relevant patterns. As the number of samples increases, the models have difficulties in generalizing effectively, especially if the additional examples introduce noise or semantic overlap.

In hackers' forum posts, adding more examples does not necessarily improve the model's understanding of the content. As shown in Figure 18, the three-shot prompting accuracy is equivalent or worse (in three out of four LLMs) than a single representative example for each class. For Gemma, in particular, additional examples seem to bias the model toward "Not relevant" (Class 3). This is because Gemma-1.1-7b, unlike the other models, is pre-trained predominantly on mathematical texts, datasets and extensive programming codes from various sources. This decrease in performance has been even more apparent in 5-shot prompting, a method we tested but decided against using due to its inefficacy.

Figure 20 illustrates the performance of LLMs across different types of prompting. While Mistral consistently shows the highest accuracy across all customization methods, there is also a noticeable improvement in the performance metrics of Llama3 compared to Llama2. This improvement highlights the advancements performed in the updated version of the model.

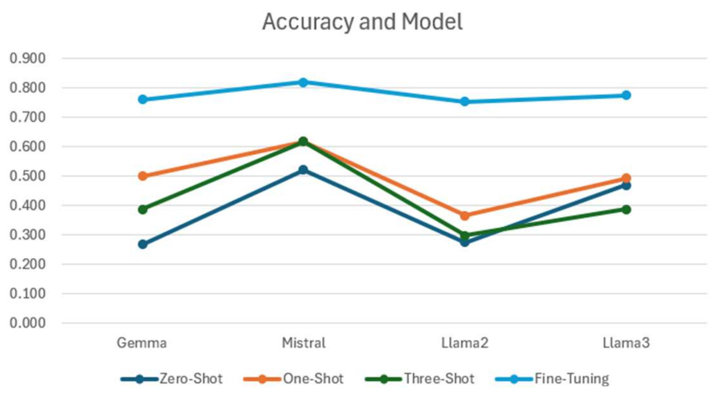


Figure 20. Accuracy per LLM (considering different customization techniques).

Mistral-7b was initially trained using online data, encompassing a broad spectrum of general internet text and common knowledge, rather than a diverse mix of web documents, scientific or academic papers, or specialized datasets in mathematics and coding. This training approach may explain why Mistral consistently outperforms other models across various prompting techniques.

Figure 21 depicts the accuracy achieved per LLM and prompting technique.

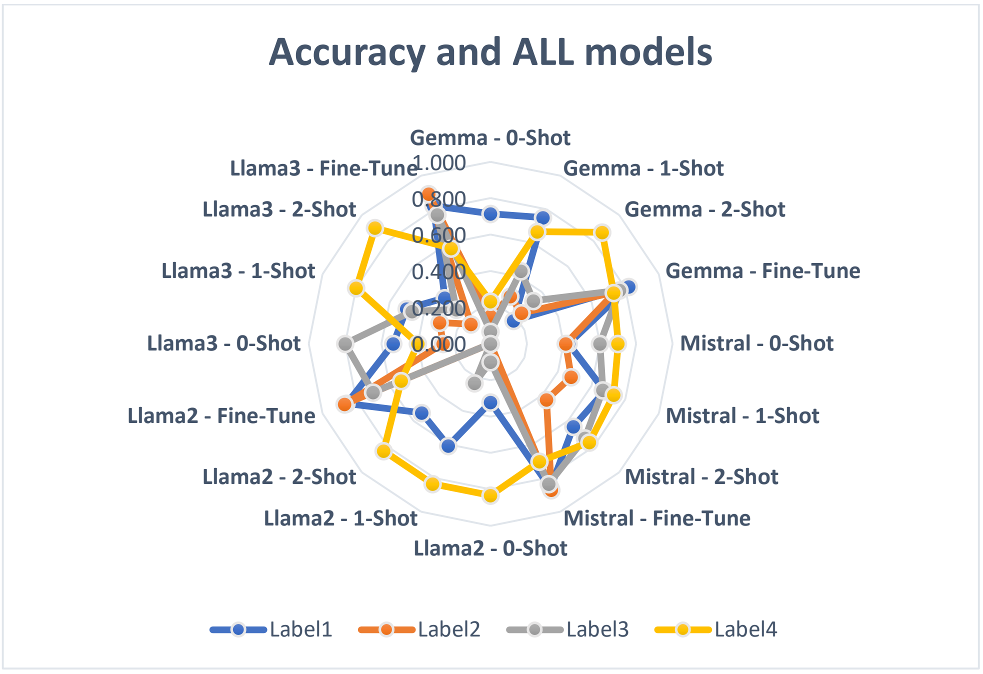


Figure 21. Accuracy per model and customization technique.

Figure 22 depicts the accuracy achieved per class for the different models employed. We observe that Mistral's performance in Class 3 ("Not Relevant") improved progressively from the zero-shot to one-shot to three-shot approaches, thanks to its general-purpose pre-training corpus. Mistral has been extensively trained on diverse general knowledge datasets, so it excels at identifying unclear or irrelevant text. This capability allows the refinement of classification boundaries in few-shot scenarios by incorporating more instances. The improvement in Class 3 demonstrates how contextual examples can diminish ambiguity and help general-purpose models, such as Mistral, align their predictions more closely with the desired label. A possible over-reliance on broad corpus patterns, however, may lead to incorrect classifications for more complex technical categories like Class 1 or Class 2.

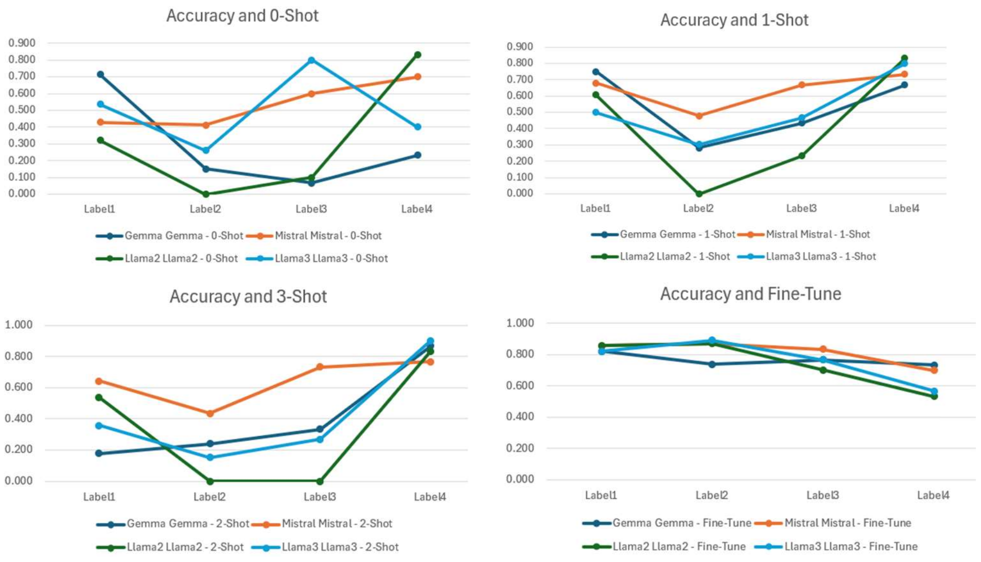


Figure 22. Accuracy per class of model and customization technique.

Additionally, Classes 1 and 2 are prone to misclassification due to possible contextual ambiguity and overlapping terminology. For instance, posts about "software patching" or "redundant backups" may be interpreted as either software defects or availability protection. These fine-tuned models have shown moderate improvement in disambiguating such cases, demonstrating their enhanced ability to differentiate between closely related terms and contexts.

LoRA selectively fine-tunes specific model layers, such as q_proj and k_proj, to enhance the model's ability to capture token relationships effectively. This targeted approach is particularly beneficial for models like Gemma and Mistral in technical categories like Class 2 ("Software and Firmware Flaws"), although it may be less effective in broader categories like Class 1 ("Availability Protection"). Similarly, both Llama2 and Llama3 are configured to use this method, with Llama3's larger scale (8 billion parameters) providing more robust contextual embeddings. This advantage makes Llama3 especially effective in handling categories with a high ambiguity, such as Class 0 ("Access Control and Management").

The accuracy of classifying technical posts is influenced by features such as the number of words, syntactic correctness and complexity, structure, and the usage of coded language. Specifically, posts that contain shorter sentences or informal language, including slang, abbreviations, and technical jargon, have been more difficult to classify. This challenge is particularly observed in zero-shot and few-shot prompting scenarios, where limited contextual information hampers the models' ability to accurately interpret and classify the content.

In terms of limitations and defining the aim of our work, we draw attention to the following points:

- Computational resources: The experiments were performed within an infrastructure of dedicated but bounded resources, employing quantized versions of LLMs.
- Restricted dataset: The absence, to our knowledge, of labelled datasets on the dark web, combined with their informal nature and the format of the posts, necessitated the generation of a labelled dataset annotated by the domain expert, which included a relatively small population (as we expect this to be typical in similar realistic cases), high quality, and limited noise.
- The subjectivity of human-based labelling: The understanding and manual annotation of the expert may also introduce variance in the classification.
- Carefully selected examples: the randomness of example selection for prompt engineering is proven to decrease the overall accuracy.

The limitations of the current work reflect the boundaries of realistic situations and deployments and do not influence the methodological approach of this study and its application within the perimeter of the system. In this respect, a practical implementation could involve the (near) real-time classification of dark web posts, generating alerts for potential cyber-security threats. This requires the continuous feeding of dark web data into the classification mechanism of our solution.

6. Conclusions and Future Work

In this work, we have explored the use of AI for the effective recognition and classification of informal text (hackers' posts). Hackers' posts were classified into four cybersecurity vulnerability categories, employing four different LLM customization techniques: zero-shot, one-shot, three-shot and fine-tuning approaches. We have contributed to advancing traditional techniques, such as cosine similarity and keyword-based analysis, by employing state-of-the-art LLMs, namely Gemma-1.1-7b, Llama3-8b-Instruct, Mistral-7b-Instruct-v0.2, and Llama-2-7b-Instruct, in their quantized versions. The classification results were evaluated according to their accuracy, precision, recall, and f1-score, comparing them against the assessments made by a human cybersecurity expert. The classification accuracy varied with the customization technique, with results reaching up to 80%.

The classification task has proven challenging due to the complex nature of the text, language and terminology coming from the dark web. This task demands a deep comprehension of language, a nuanced understanding of the context, and the ability to learn in context. To overcome these challenges, both fundamental and advanced capabilities of LLMs are required [24].

The analysis highlighted the importance of balancing and optimizing example selection to enhance the language model's performance without adding unnecessary complexity. Our work has produced a labelled dataset of hacker posts, which aligns hacker texts with categorized cybersecurity vulnerabilities, thus enriching the resources available to the cybersecurity community.

Moving forward, a key area for future research is examining the performance of larger LLMs, such as Mixtral, GPT-4, and other models with more than 8 billion parameters, and assessing the impact of using models without quantization (full precision) in terms of classification accuracy. This study has also underscored the importance of fine-tuning for cybersecurity text classification. In this view, further efforts should explore advanced fine-tuning methods, such as domain-specific continual learning, parameter-efficient fine-tuning (PEFT), and low-rank adaptation (LoRA), to enhance model accuracy while maintaining computational efficiency. Additionally, self-supervised learning techniques may improve the flexibility of LLMs without relying on extensive human-labelled datasets.

While this work is primarily focused on evaluating prompt engineering and fine-tuning techniques for LLMs, the expert-annotated dataset of cybersecurity vulnerabilities and hacker forum posts could serve as a valuable benchmark for various deep learning architectures, including CNNs, RNNs, and Transformer-based models, as well as traditional ML models, like SVMs, Random Forests, and Decision Trees. Future research should assess these models in terms of computing costs, interpretability, and efficiency in comparison with LLMs. Additionally, the dataset could be enriched by incorporating more posts, threat intelligence reports, and metadata, including geographical threat landscapes, multilingual content aspects, and the confrontation/mitigation of identified vulnerabilities [45].

Author Contributions: Conceptualization, T.G. and A.P.; methodology, A.P.; software, T.G.; formal analysis, N.N.; resources, T.G.; writing—review and editing, A.P. and T.G.; visualization, N.N.; supervision, T.Z.; project administration, T.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The hacker forum posts are publicly available. The selected and labelled posts and the trained LLMs are available upon request.

Conflicts of Interest: All authors were employed by the company Synelixis S.A. The 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.

References

1. Zhang, J.; Bu, H.; Wen, H.; Liu, Y.; Fei, H.; Xi, R.; Li, L.; Yang, Y.; Zhu, H.; Meng, D. When LLMs Meet Cybersecurity: A Systematic Literature Review. *Cybersecurity* **2025**, *8*, 55. [CrossRef]
2. Liu, Z.; Shi, J.; Buford, J.F. CyberBench: A Multi-Task Benchmark for Evaluating Large Language Models in Cybersecurity. In Proceedings of the AAAI-24 Workshop on Artificial Intelligence for Cyber Security (AICS), Vancouver, BC, Canada, 26–27 February 2024.
3. Open-Source Intelligence (OSINT), OSINT Framework. Available online: https://osintframework.com/ (accessed on 3 March 2025).
4. Clairoux-Trepanier, V.; Beauchamp, I.M.; Ruellan, E.; Paquet-Clouston, M.; Paquette, S.O.; Clay, E. The Use of Large Language Models (LLM) for Cyber Threat Intelligence (CTI) in Cybercrime Forums. *arXiv* **2024**. [CrossRef]
5. Chen, Y.; Cui, M.; Wang, D.; Cao, Y.; Yang, P.; Jiang, B.; Lu, Z.; Liu, B. A survey of large language models for cyber threat detection. *Comput. Secur.* **2024**, *145*, 104016. [CrossRef]
6. Chen, H.; Diao, Y.; Xiang, H.; Huo, Y.; Xie, X.; Zhao, J.; Wang, X.; Sun, Y.; Shi, J. Decode the Dark Side of the Language: Applications of LLMs in the Dark Web. In Proceedings of the 2024 IEEE 9th International Conference on Data Science in Cyberspace (DSC), Jinan, China, 23–26 August 2024; pp. 224–231. [CrossRef]
7. Balasubramanian, P.; Nazari, S.; Kholgh, D.K.; Mahmoodi, A.; Seby, J.; Kostakos, P. A cognitive platform for collecting cyber threat intelligence and real-time detection using cloud computing. *Decis. Anal. J.* **2025**, *14*, 100545. [CrossRef]
8. European Parliament, Directive (EU) 2022/2555 on Measures for a High Common Level of Cybersecurity Across the Union (NIS 2). Available online: https://eur-lex.europa.eu/eli/dir/2022/2555/oj/eng (accessed on 3 March 2025).
9. European Parliament, Directive (EU) 2022/2557 on the Resilience of Critical Entities. Available online: https://eur-lex.europa.eu/eli/dir/2022/2557/oj (accessed on 3 March 2025).
10. University of Arizona. Intelligence and Security Informatics Data Sets. Available online: https://www.azsecure-data.org (accessed on 3 March 2025).
11. National Institute of Standards and Technology (NIST). The NIST Cybersecurity Framework (CSF) 2.0. Available online: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.pdf (accessed on 3 March 2025).
12. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**. [CrossRef]
13. Truong, T.C.; Zelinka, I.; Plucar, J.; Čandík, M.; Šulc, V. Artificial Intelligence and Cybersecurity: Past, Presence, and Future. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems*; Advances in Intelligent Systems and Computing; Dash, S., Lakshmi, C., Das, S., Panigrahi, B., Eds.; Springer: Singapore, 2020; Volume 1056. [CrossRef]
14. Wiafe, I.; Koranteng, F.N.; Obeng, E.N.; Assyne, N.; Wiafe, A.; Gulliver, S.R. Artificial Intelligence for Cybersecurity: A Systematic Mapping of Literature. *IEEE Access* **2020**, *8*, 146598–146612. [CrossRef]

15. Kaur, R.; Gabrijelčič, D.; Klobučar, T. Artificial Intelligence for Cybersecurity: Literature Review and Future Research Directions. *Inf. Fusion* **2023**, *97*, 101804. Available online: https://www.sciencedirect.com/science/article/pii/S1566253523001136 (accessed on 3 March 2025). [CrossRef]
16. Psychogyios, K.; Papadakis, A.; Bourou, S.; Nikolaou, N.; Maniatis, A.; Zahariadis, T. Deep Learning for Intrusion Detection Systems (IDSs) in Time Series Data. *Future Internet* **2024**, *16*, 73. [CrossRef]
17. Dilek, S.; Cakır, H.; Aydın, M. Applications of Artificial Intelligence Techniques to Combating Cyber Crimes: A Review. *Int. J. Artif. Intell. Appl.* **2015**, *6*, 21–39. [CrossRef]
18. Deng, P.; Wang, J.-H.; Shieh, W.-G.; Yen, C.-P.; Tung, C.-T. Intelligent automatic malicious code signatures extraction. In Proceedings of the IEEE 37th Annual 2003 International Carnahan Conference on Security Technology, Taipei, Taiwan, 14–16 October 2003; pp. 600–603. [CrossRef]
19. Shabtai, A.; Kanonov, U.; Elovici, Y.; Glezer, C.; Weiss, Y. "Andromaly": A behavioral malware detection framework for android devices. *J. Intell. Inf. Syst.* **2012**, *38*, 161–190. [CrossRef]
20. Zhang, Z.; Ning, H.; Shi, F.; Farha, F.; Xu, Y.; Xu, J.; Zhang, F.; Choo, K.-K.R. Artificial intelligence in cyber security: Research advances, challenges, and opportunities. *Artif. Intell. Rev.* **2022**, *55*, 1029–1053. [CrossRef]
21. Alam, S. Cybersecurity: Past, Present and Future. *arXiv* **2024**. [CrossRef]
22. Yang, L.; Shami, A. Towards Autonomous Cybersecurity: An Intelligent AutoML Framework for Autonomous Intrusion Detection. In Proceedings of the Workshop on Autonomous Cybersecurity (AutonomousCyber'24), Association for Computing Machinery, New York, NY, USA, 14–18 October 2024; pp. 68–78. [CrossRef]
23. Large Language Models in Cybersecurity: State-of-the-Art, Farzad Nourmohammadzadeh Motlagh, Mehrdad Hajizadeh, Mehryar Majd, Pejman Najafi, Feng Cheng, Christoph Meinel. *arXiv* **2024**. [CrossRef]
24. Minaee, S.; Mikolov, T.; Nikzad, N.; Chenaghlu, M.; Socher, R.; Amatriain, X.; Gao, J. Large Language Models: A Survey. *arXiv* **2024**. [CrossRef]
25. Zhang, S.; Dong, L.; Li, X.; Zhang, S.; Sun, X.; Wang, S.; Li, J.; Hu, R.; Zhang, T.; Wu, F.; et al. Instruction Tuning for Large Language Models: A Survey. *arXiv* **2023**. [CrossRef]
26. Tihanyi, N.; Ferrag, M.A.; Jain, R.; Bisztray, T.; Debbah, M. CyberMetric: A Benchmark Dataset based on Retrieval-Augmented Generation for Evaluating LLMs in Cybersecurity Knowledge. In Proceedings of the 2024 IEEE International Conference on Cyber Security and Resilience (CSR), London, UK, 2–4 September 2024; pp. 296–302. [CrossRef]
27. Bhattacharjee, A.; Moraffah, R.; Garland, J.; Liu, H. Zero-shot LLM-guided Counterfactual Generation: A Case Study on NLP Model Evaluation. In Proceedings of the 2024 IEEE International Conference on Big Data (BigData), Washington, DC, USA, 15–18 December 2024; pp. 1243–1248. [CrossRef]
28. Ferrag, M.A.; Alwahedi, F.; Battah, A.; Cherif, B.; Mechri, A.; Tihanyi, N.; Bisztray, T.; Debbah, M. Generative AI and Large language models for cybersecurity: All Insights you need. *arXiv* **2025**, arXiv:2405.12750. Available online: https://arxiv.org/abs/2405.12750 (accessed on 3 March 2025).
29. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
30. Hassanin, M.; Moustafa, N. A Comprehensive Overview of Large Language Models (LLMs) for Cyber Defences: Opportunities and Directions. *arXiv* **2024**. [CrossRef]
31. Yuan, X.; Wang, J.; Zhao, H.; Yan, T.; Qi, F. Empowering LLMs with Toolkits: An Open-Source Intelligence Acquisition Method. *Future Internet* **2024**, *16*, 461. [CrossRef]
32. Sahoo, P.; Singh, A.K.; Saha, S.; Jain, V.; Mondal, S.; Chadha, A. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. *arXiv* **2024**. [CrossRef]
33. Han, Z.; Gao, C.; Liu, J.; Zhang, J.; Zhang, S.Q. Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey. *arXiv* **2024**. [CrossRef]
34. Cer, D.; Yang, Y.; Kong, S.-Y.; Hua, N.; Limtiaco, N.; John, R.S.; Constant, N.; Guajardo-Céspedes, M.; Yuan, S.; Tar, C. Universal Sentence Encoder. *arXiv* **2018**. [CrossRef]
35. Nikolaou, N.; Papadakis, A.; Psychogyios, K.; Zahariadis, T. Vulnerability Identification and Assessment for Critical Infrastructures in the Energy Sector. *Electronics* **2023**, *12*, 3185. [CrossRef]
36. Guidelines for Smart Grid Cybersecurity, Volume 3—Supportive Analyses and References, NISTIR 7628 Revision 1, 2014. Available online: https://nvlpubs.nist.gov/nistpubs/ir/2014/nist.ir.7628r1.pdf (accessed on 3 March 2025).
37. *ISO/IEC 27005:2022*; Information Security, Cybersecurity and Privacy Protection—Guidance on Managing Information Security Risks, Edition 4. ISO/IEC, International Standard: Geneva, Switzerland, 2022.
38. Amatriain, X. Prompt Design and Engineering: Introduction and Advanced Methods. *arXiv* **2024**. [CrossRef]
39. Schulhoff, S.; Ilie, M.; Balepur, N.; Kahadze, K.; Liu, A.; Si, C.; Li, Y.; Gupta, A.; Han, H.; Schulhoff, S. The Prompt Report: A Systematic Survey of Prompting Techniques. *arXiv* **2024**, arXiv:2406.06608. Available online: https://arxiv.org/abs/2406.06608 (accessed on 3 March 2025).

40. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F. LLaMA: Open and Efficient Foundation Language Models. *arXiv* **2023**, arXiv:2302.13971. Available online: https://arxiv.org/abs/2302.13971 (accessed on 3 March 2025).
41. Ahmed, T.; Devanbu, P. Few-shot training LLMs for project-specific code-summarization. In Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering (ASE'22), ACM, New York, NY, USA, 10–14 October 2022. [CrossRef]
42. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv* **2021**, arXiv:2106.09685. Available online: https://arxiv.org/abs/2106.09685 (accessed on 3 March 2025).
43. Demszky, D.; Movshovitz-Attias, D.; Ko, J.; Cowen, A.; Nemade, G.; Ravi, S. GoEmotions: A Dataset of Fine-Grained Emotions. *arXiv* **2020**, arXiv:2005.00547. Available online: https://arxiv.org/abs/2005.00547 (accessed on 3 March 2025).
44. GoEmotions Dataset. Available online: https://huggingface.co/datasets/google-research-datasets/go_emotions (accessed on 3 March 2025).
45. Zhou, X.; Cao, S.; Sun, X.; Lo, D. Large Language Model for Vulnerability Detection and Repair: Literature Review and the Road Ahead. *ACM Trans. Softw. Eng. Methodol.* **2024**. Available online: https://dl.acm.org/doi/10.1145/3708522 (accessed on 3 March 2025). [CrossRef]