Large Language Models in Cybersecurity: State-of-the-Art

Farzad Nourmohammadzadeh Motlagh[1], Mehrdad Hajizadeh[2], Mehryar Majd[1], Pejman Najafi[1], Feng Cheng[1] and Christoph Meinel[1]

[1] *Hasso-Plattner-Institute for Digital Engineering, University of Potsdam, Germany*

[2] *Technische Universitat Chemnitz, Germany*

{*farzad.motlagh, mehryar.majd, pejman.najafi, feng.cheng, christoph.meinel*}*@hpi.de,*

Keywords: LLM, Large Language Model, AI, Cybersecurity, Advanced Threats, Cyberattacks, Cyberdefense, Privacy and Security.

Abstract: The rise of Large Language Models (LLMs) has revolutionized our comprehension of intelligence bringing us closer to Artificial Intelligence. Since their introduction, researchers have actively explored the applications of LLMs across diverse fields, significantly elevating capabilities. Cybersecurity, traditionally resistant to data-driven solutions and slow to embrace machine learning, stands out as a domain. This study examines the existing literature, providing a thorough characterization of both defensive and adversarial applications of LLMs within the realm of cybersecurity. Our review not only surveys and categorizes the current landscape but also identifies critical research gaps. By evaluating both offensive and defensive applications, we aim to provide a holistic understanding of the potential risks and opportunities associated with LLM-driven cybersecurity.

1 INTRODUCTION

The evolution of generative artificial intelligence, notably large language models (LLMs), has influenced most disciplines of science and technology that support content generation in diverse applications (Neupane et al., 2023). In education, LLMs support educators in various tasks such as assignment assessment (Hsiao et al., 2023), question generation (Elkins et al., 2023), providing feedback (Guo and Wang, 2023), and essay grading (Yan et al., 2023). In the entertainment industry, LLMs demonstrate competitive performance in generating music captions (Deng et al., 2023b) as well as video game scripts (Latouche et al., 2023). Automation is introduced into customer service (Pandya and Holia, 2023), marketing (Gan et al., 2023; Yang et al., 2023b), and supply chain management (Hendriksen, 2023; Li et al., 2023a; Kosasih et al., 2023) through the integration of LLMs in business. Meanwhile, the utilization of LLMs in healthcare enables professionals by providing real-time clinical decision support (Rao et al., 2023; Fawzi, 2023), medical education (Kuckelman et al., 2023; Song et al., 2023), and prediction of disease progression (Shoham and Rappoport, 2023; Rasmy et al., 2021).

With advancements in cyber threats, the cybersecurity domain can also be equipped with cutting-edge tools, assisting cybersecurity practitioners who continuously seek solutions to implement advanced policies or strengthen technological protections against the disclosure of confidential information, unauthorized access, and other forms of data modification (Kaur et al., 2023). Thanks to LLMs' capability in breaking down complex natural language patterns, security experts are now enabled to explore more attack vectors in various contexts associated with textual data (Yang et al., 2023a).

Functionalities of LLMs are increasingly being integrated into the cybersecurity posture, contributing to promising enhancements in cybersecurity defense applications (Li et al., 2023b). Through analyzing vast amounts of text data, including security logs, these models can identify emerging vulnerabilities. Anomaly detection represents a key application of LLMs for identifying potential threats (Liu et al., 2023). Furthermore, LLMs mitigate potential risks by offering automated vulnerability fixes, aiming to improve organizations' security posture (Pearce et al., 2023).

However, with the continuous advancements of LLMs in cyber defense, it is crucial to acknowledge

Motlagh, F. N., Hajizadeh, M., Majd, M., Najafi, P., Cheng, F. and Meinel, C.
Large Language Models in Cybersecurity: State-of-the-Art.
DOI: 10.5220/0013377600003899

In *Proceedings of the 11th International Conference on Information Systems Security and Privacy (ICISSP 2025) - Volume 2*, pages 98-110
ISBN: 978-989-758-735-1; ISSN: 2184-4356

that these language models can also be leveraged by malicious actors. For example, LLMs can be misused by attackers to execute malware in target companies (Botacin, 2023), engage in defense evasion (Chatzoglou et al., 2023), and gain access to credentials (Rando et al., 2023). The potential to generate complex and personalized phishing messages further highlights the misuse of LLMs for deceiving people in an organization, paving the way for unauthorized access to companies' sensitive information (Saha Roy et al., 2023; Jiang, 2024). To further elaborate, WormGPT (Falade, 2023) is an AI-powered tool designed for cybercriminals to automate the generation of personalized phishing emails. Although it may sound somewhat similar to ChatGPT, WormGPT is not a friendly neighborhood AI; instead, its purpose is to produce malicious content. Furthermore, FraudGPT (Dutta, 2023) enabled attacker to create content to convince users to click on a particular generated link.

The dual nature of LLMs has transformed the cybersecurity realm by offering new challenges and opportunities. Developing robust defensive strategies to foresee attacks and address concerns related to the utilization of LLMs motivated us to formulate a taxonomy of strategies appearing in the field of cybersecurity. To define our contributions more precisely, this paper addresses:

- The intersection of LLMs' offensive approaches as a newly introduced dimension to cybersecurity is framed in this study in line with the Mitre attack framework (Corporation, 2023).
- Exploring LLM-empowered defensive strategies in dealing with potential threats and malware based on the NIST cybersecurity framework (Cybersecurity, 2014).
- Understanding the major functionalities of LLMs in current research trends alongside potential applications in the cybersecurity landscape.

The rest of paper is organized as follows: In Section 2, we provide an overview of LLMs . Moving forward to Section 3, we explore cyber threat defenses leveregred by LLMs where Section 4 outlines sophisticated attacks designed by LLMs. Finally, Section 5 concludes the challenges posed by LLMs in the context of cybersecurity.

2 BACKGROUND

LLMs are neural networks that learn from textual data to process various language-related tasks (Naveed et al., 2023). From Eliza as a pattern recognition chatbot in the 1960s (Weizenbaum, 1966), over the years several advancements pushed Natural Language Processing (NLP) forward, such as long short-term memories to handle a wide range of data (Hochreiter and Schmidhuber, 1997), Stanford CoreNLP suite (Manning et al., 2014) providing a collection of algorithms to perform intricate NLP tasks and continued with transformer architecture (Vaswani et al., 2017).

A breakthrough in Transformer-based models surged the field of NLP and led to the development of numerous kinds of effective LLMs. T5 (Raffel et al., 2020) applied language modeling in pre-trained LLMs, where spans are altered with a single mask. GPT-3 enhanced the performance of LLMs with size by increasing model parameters to 175B. PaLM-2 is trained on high-quality datasets (Anil et al., 2023) with an objective of cutting the cost of training and inference (Naveed et al., 2023). Llama, a set of decoder-only models aimed at minimizing the amount of activations in the backward step (Naveed et al., 2023; Touvron et al., 2023). Xuan Yuan 2.0, a Chinese financial chat model (Naveed et al., 2023; Zhang and Yang, 2023), AlexaTM (Soltan et al., 2022), PaLM-2 (Anil et al., 2023), as well as GLM-130B (Zeng et al., 2022) are a few instances of general purpose pre-trained LLMs. While pre-trained models offer an essential understanding of languages, as AI advances, fine-tuning LLMs boost business functions and satisfaction by fulfilling industry-specific criteria (Zhang et al., 2023b). A general-purpose LLaMA-GPT-4 (Peng et al., 2023), Goat (Liu and Low, 2023) for handling complicated arithmetic queries, HuaTuo (Wang et al., 2023) a medical knowledge model, Evol-Instruct (Xu et al., 2023) offering complicated prompts, and LLaMA 2-Chat fine-tuned using rejection sampling (Touvron et al., 2023) are exemplary instruction-tuning LLMs. Running in higher costs, extensive hardware requirements, cost of slow training on various tasks, limited LLMs utilization (Naveed et al., 2023). Retrieving support evidence from an external in-domain knowledge base (Zhang et al., 2023a), parameter tuning and knowledge distillation are among the techniques extensively researched for effective LLM deployment (Naveed et al., 2023).

Recently, the scientific literature has experienced a significant growth in the number of articles related to LLMs, principally driven by their proven efficacy across a wide range of functions. As a result, throughout various surveys, researchers attempted to categorize these advancements in LLM architecture (Naveed et al., 2023; Zhao et al., 2023; Zhou et al., 2023; Huang and Chang, 2022). Though previous studies have investigated literature reviews to highlight the safety aspects of LLMs (Iturbe et al., 2023; Addington, 2023; Kucharavy et al., 2023; Ishihara, 2023), the present study focuses primarily on the application of LLMs in the context of cyberdefense as well as cyber-attack.

3 DEFENSIVE APPLICATIONS OF LLMs

In the field of cybersecurity, the National Institute of Standards and Technology (NIST) provides a comprehensive structure to enhance organizations' cybersecurity status, as detailed in the NIST cybersecurity framework (Cybersecurity, 2014). According to its effectiveness and popularity in cyberdefense, we classify the diverse array of LLM-centered approaches that contributed in cyberdefense through the lens of NIST framework to better understand the impact of LLMs in cyberdefense. The framework consists of a structured approach to identify, protect, detect, respond to, and recover from cybersecurity threats and incidents.

3.1 Identify

The process of developing an organizational understanding to manage cybersecurity risk concerning systems, assets, data, and capabilities is referred to the *Identify* function in the context of the NIST framework (Cybersecurity, 2014). Identifying potential risks is a crucial phase in risk management, and LLMs aim to fulfill a transformational role in forming risk management in businesses. Johnson (Johnson, 2023) presents invaluable insights for policymakers on the applicability of LLMs to risk management. According to the author, LLMs go through business headlines, social media posts, economic indicators, legal documentation, and other key sources, emphasizing risk elements to deliver more accurate and predictive risk assessments that a human analyst might overlook. Lima et al. (de Lima et al., 2023) develop a risk matrix from application reviews using LLMs. Through user feedback, they proposed an automatic prompt extraction technique. These prompts were passed into LLMs, which classified the risks into five classes ranging from negligible to critical for further investigation. Naleszkiewicz (Naleszkiewicz, 2023) discusses LLM applications allowing companies to overcome traditional enterprise risk management challenges, such as operational and compliance risks. LLMs evaluate unstructured siloed data across various departments, acting as a bridge to provide an in-depth understanding of an organization's risk profile. Furthermore, LLMs boost risk modeling by generating expert opinions based on prior patterns, risk mitigation by generating contingency plans, and risk reporting by providing customized risk assessments.

3.2 Protect

Implementing safeguards to guarantee the delivery of essential services is reflected in *protect* function (Cybersecurity, 2014). It involves various mechanisms such as maintaining a proactive security posture or prioritizing cybersecurity awareness and training to empower the organization's workforce. In the current digital environment, proactive protection technologies are essential since they enable companies to anticipate and prevent troubles before they arise. For example, proactive technologies empower enterprises to minimize the likelihood of coming across inappropriate content, and thus reduce the possibility of experiencing ethical or legal challenges (Sun et al., 2023). Voros et al. (Vörös et al., 2023) harnessed the power of LLMs to enhance web content filtration. They have improved the accuracy of web content categorization by scanning of large amount of URLs. Another research accomplished by Yu et al. (Yu and Martin, 2023) investigates GPT-3's capacity to produce honeywords to trap the attackers if they are using deceptive generated passwords. First, they extract the components of the original password using a password-specific segmentation algorithm. These segments are then fed into GPT-3 as a prompt to generate a collection of passwords similar to the input password. A crucial element in this model's efficacy is the maintenance of strong password components called chunks given to the LLM (Sannihith Lingutla, 2023).

LLMs can play a valuable role in strengthening cybersecurity awareness and training within the protect function of the NIST framework. Tann et al. (Tann et al., 2023) apply LLMs to tackle professional certification topics and perform Capture The Flag (CTF) tasks to improve participants' cybersecurity education. LLMs have significance by enabling attendees to explore CTF test settings, providing explanations to concerns connected to professional certification, and highlighting the need to model cybersecurity breach scenarios in CTF sessions to support the development of more comprehensive skills. However, LLMs face limitations when it comes to responding to conceptual queries. Furthermore, LLMs can improve team collaboration by offering security question solutions that are suitable for inexperienced as well as experts. For instance, LLMs greatly increase the efficacy of penetration test teams by making it easier for team members to pass on information by offering more in-depth assessments and generating appropriate explanations to be on the same page about the detected risks. Moreover, LLMs serve as a connection between experts and publicly accessible web resources, in particular assisting specialists in remaining up to date on the most recent security concerns that are critical to their company (Dutta et al., 2018).

Automated vulnerability fixing with LLMs diminishes the risk of cyberattacks. A three-step process is described by Charalambous et al. (Charalambous et al., 2023) for addressing automotive vulnerability issues. Bounded Model Checking (BMC) is the first step in the process. It evaluates the user-provided source code to a property specification. The original code and the appropriate counterexample are provided to the LLM module by the BMC engine in the scenario that this phase's verification is unsuccessful and a security property violation is detected. Secondly, customized queries are sent to the LLM engine to produce a corrected version of the code. Lastly, the BMC module re-evaluates the code that the LLM module changed to formally determine whether the updated version matches the original security and safety requirements.

Automating flaw mitigation can be facilitated by LLMs if the defect is well-defined and the prompt provides additional information. While these models were fully effective in fixing simulated vulnerabilities, real-world scenarios presented challenges for their performance. The primary challenges stem from the numerous methods that information is presented, the complexities of prompt processing and code development in LLMs, and the significance of prompt phrasing, which can result in notable variations in the code required to be generated (Pearce et al., 2023). Furthermore, Sandoval et al. (Sandoval et al., 2023) performed an examination of potentially insecure code suggestions during the process of code development. Within a particular programming context that the authors had defined, they tested scenarios with and without AI support. Their findings indicate that users assisted by AI develop security flaws at a rate lower than ten percent, suggesting that using LLMs in their security-oriented research does not present major new security risks. Additionally, Yu et al. (Fengrui and Du, 2024) present a method for automating Tactics, Techniques, and Procedures (TTP) classification in few-shot learning scenarios. The method employs ChatGPT for data augmentation and Instruction-Supervised Fine-Tuning on large language models. Using ChatGPT results in diverse sample expansion that do not undermine the original text's contextual semantic.

3.3 Detect

The NIST framework's *Detect* function serves to identify cybersecurity events as they arise (Cybersecurity, 2014). Exploring anomaly detection in system logs is a crucial step toward developing effective detection methods through the use of LLMs. Recurrent Neural Network Language Models are used by Tuor et al. (Tuor et al., 2018) to present an unsupervised, online anomaly detection method for computer security log analysis. This approach simplifies the usual effort-intensive feature engineering stage, making it fast to implement, and is independent of the tools used for system configuration and monitoring. The authors have demonstrated the efficacy of their approach by utilizing the Los Alamos National Laboratory Cyber Security Dataset (Kent, 2016). Their findings indicate that the approach can be handled in real-time, generating and organizing log-line-level anomaly scores while taking into account inter-log-line context. The authors (Tuor et al., 2018) considered metrics including Average Percentile (AP) and Area under the Receiver Operator Characteristic Curve (AUC) to show how the false-positive rate dropped without significantly affecting the ability to detect unusual behavior (Kent, 2016).

GPT-2 is used by VulDetect(Omar and Shiaeles, 2023), a transformer-based vulnerability detection framework, to detect anomalies in system logs. Using a dataset containing both vulnerable and non-vulnerable code, the model is fine-tuned to detect anomalies that represent regular behavior. Malicious behavior is defined as any unexpected or unlikely outcome that the model possibly generated. Two benchmark datasets, SARD (Zhou and Verma, 2022) and SeVC (Shoeybi et al., 2019), were utilized by the authors to assess VulDetect's performance. The outcomes showed that VulDetect has a low false positive rate and is efficient in real-time vulnerability detection. Moreover, the integration of LLMs into penetration testing practices has the potential to revolutionize the world of threat detection. Threat detection could undergo a revolution if LLMs are incorporated into penetration testing procedures. Happe et al.'s investigation (Happe and Cito, 2023) focused on using LLMs to improve penetration testing. In line with their classification, LLMs provide advancement in two aspects of penetration testing: high-level and low-level operations. High-level assignments include conceptual investigation and strategic planning, such as finding out about emerging active directory attacks. On the other hand, tasks at a lower level incorporate consideration of practical activities involving system exploitation and vulnerability analysis. This entails looking for specific attack vectors for a particular system.

A further investigation by Deng et al. (Deng et al., 2023a) introduces PENTESTGPT, an automated penetration testing system driven by LLMs. Complex tasks such as question answering, summarization, and reasoning are readily handled with PENTESTGPT. Addressing context loss concerns and simulating human behavior in penetration testing are the objectives. Three self-interacting modules jointly form PENTESTGPT including reasoning, generation, and parsing. These modules collaborate to tackle penetration testing problems by using a divide-and-conquer approach. Specific subtasks are allocated to each module, which interact to effectively handle and compile the data generated during testing.

Ranade et al. (Ranade et al., 2021) improve the processing of threats, attacks, and vulnerabilities which is challenging due to the high volume of data, and the dynamic nature of evolving attack techniques. The primary objective of their research is an enhanced version of a BERT model, which aims to effectively perform several cybersecurity-related operations. Using Masked Language Modeling (MLM), the model was trained using unstructured and semi-structured open-source Cyber Threat Intelligence (CTI) data. Its evaluation encompassed diverse downstream tasks with potential applications in Security Operations Centers (SOCs). They additionally offer real-world examples of how to apply CyBERT to cybersecurity problems. Several subsequent works have furthered the advancements of this research in terms of both training efficiency and accuracy such as SecureBERT (Aghaei et al., 2022), CySecBERT (Bayer et al., 2022), and ClaimsBERT (Ameri et al., 2022). In this regard, Bayer et al. (Bayer et al., 2022) presented a word embedding model based on BERT and collected a dataset from multiple sources. This adaptation makes the model capable of coping with a wide range of cybersecurity tasks, namely malware detection, alert aggregation, and phishing website detection. Similarly, LILAC (Jiang et al., 2024) is a log parsing method that employs an adaptive parsing cache to boost the efficiency of log analysis procedures. LILAC attempts to tackle issues such as inconsistent outputs and a lack of specialized log parsing capacities by updating templates using LLMs' in-context learning (ICL) power and a novel adaptive parsing cache.

The LLMs can also facilitate auditory tasks to detect vulnerabilities among the smart contracts. David et al. (David et al., 2023) utilized LLMs to target vulnerabilities in the smart contracts and DeFi protocol layers. Their study detects 52 compromised DeFi protocols, as input data for the language model context, evaluating the impact of model temperature and context length on the language model's efficacy in smart contract auditing. The results indicated that incorporating LLMs into the audit workflow substantially boost the effectiveness and accuracy of analyzing an array of feasible attacks. On the other hand, Chen et al. (Chen et al., 2023) trained LLM on a dataset of 10,000 smart contracts and evaluated how well it detected nine different vulnerabilities. According to the authors' findings, LLMs frequently deliver false positive results when detecting smart contract vulnerabilities. This might be connected with interference from incomplete codes or LLMs' incapacity to understand code segments.

An LLM can be used to build a scenario comparable to an attacker's strategy for gaining access to an organization's property by exploiting a vulnerability. Garvey et al. (Garvey and Svendsen, 2023) study the viability of using Generative-AI to improve the development of Red Team scenarios in organizations. The authors (Garvey and Svendsen, 2023) propose employing LLMs to construct narratives based on prompts or questions as input. Subsequently, subject-matter specialists provide remarks, including modifying narratives, adding new elements, or integrating multiple items to develop more complex scenarios. The objective is to guarantee that the generated scenarios are plausible and adhere to the provided framework. They found that including elements inspired by fiction into LLMs improves creativity and imagination in the scenario development process.

Koide et al. (Koide et al., 2023) present a strategy for detecting phishing websites using LLMs. Their approach entails using a web crawler to retrieve data from websites and creating prompts for LLMs. Social engineering strategies are then identified by evaluating the context of entire web pages and URLs. The prompts rely on the Chain of Thought (CoT) prompting technique, which enables LLMs to elaborate on their reasoning. In addition, the study recommends an HTML simplification approach to improve efficiency. This entails lowering the token count by simplifying HTML text and removing HTML elements that lack text within tags, such as style, script, and comment tags. This operation is repeated until the token count reaches a certain threshold, thus boosting overall efficiency.

Sakaoglu introduced KARTAL(Sakaoglu, 2023), a fine-tuned Language Model for detecting vulnerabilities in web applications. A detector component in the KARTAL system is controlled by the prompts from the prompter component. These prompts are generated based on input gathered by the fuzzer component, which monitors application activity. The LLM detects logical vulnerabilities in web applications, specifically broken access control rules, by analyzing these prompts. This technique allows KARTAL to dynamically alter the definitions of broken access, allowing it to adapt to a variety of scenarios. This adaptability distinguishes it from less intelligent vulnerability scanners, allowing KARTAL to be more effective in its detection capabilities.

LLMs demonstrate their capacity to be an effective method across a wide range of vulnerability identification tasks. CyBERT (Ameri et al., 2021) unveils a classifier for detecting cybersecurity feature claims. The method incorporates fine-tuning a pretrained BERT language model to recognize cybersecurity claims throughout complex sequences observed in industrial control systems (ICS) device documentation. This is accomplished by aggregating reports for each feature from every source linked with an individual device, effectively determining in-conflict feature claims. The extraction of sequences from ICS-related documents is the initial stage in the procedure as these sequences are classified into broad claims, device claims, or cybersecurity claims. Then, the identified sequences are used to train CyBERT so it can classify new sequences.

SecurityLLM, a system developed for precise threat detection and data privacy, is presented by Ferrag et al. (Ferrag et al., 2023b). SecurityLLM utilizes Fixed-Length Language Encoding (FLLE) as a privacy-preserving encoding method, in conjunction with the Byte-level Byte-Pair Encoder (BBPE) Tokenizer forming text traffic data. The SecurityLLM framework is composed of two primary components: SecurityBERT, which detects cyber threats, and FalconLLM, which responds to and recovers from incidents. The method, which was trained on an IoT cybersecurity dataset, displays significant accuracy in identifying fourteen various types of cyber threats.

SecureFalcon (Ferrag et al., 2023a) is an LLM-based cybersecurity reasoning system targeted to detect software flaws. The method involves fine-tuning FalconLLM with the use of a FormAI dataset including C code instances. SecureFalcon (Ferrag et al., 2023a) uses binary classification to distinguish between vulnerable and non-vulnerable patterns and then validates corrected code using Bounded Model Checking. However, the study's adaptability is limited due to the FormAI dataset's exclusive focus on C codes.

3.4 Respond

The *Respond* function involves the formulation of actions to address the detected incident (Cybersecurity, 2014). The convergence of LLMs and honeypot paradigms enhances the capability to respond to malware threats. In exploring this synergy, McKee et al. (McKee and Noever, 2023) research the feasibility of using LLMs to improve cybersecurity in a honeypot setup. The researchers (McKee and Noever, 2023) demonstrate how these chatbots can create a responsive honeypot interface capable of responding to illicit activities. This method gives security professionals more time to respond to an ongoing cyber attack. Ten tasks connected with the development of honeypots are divided into three primary categories by the authors (McKee and Noever, 2023): networks, operating systems, and applications. Their results indicate that the LLM-based honeypot interfaces are able to maintain the attacker's interest over the course of several inquiries. In another study, Sladic et al. (Sladić et al., 2023) present an LLM-based technique for developing software honeypots. The devised honeypot named shelLM is designed to evaluate the credibility of the model through the use of security experts in an experiment. The specialists collaborated with ShelLM to assess how it responded to the commands of an attacker. ShelLM's ability to retain consistency over several sessions is a significant feature; the content of each terminal session is kept and used as a prompt for following sessions. This makes sure that regardless of when a session comes to an end interactions can carry on without interruption. Cambiaso et al. (Cambiaso and Caviglione, 2023) deliver a method for generating email messages to identified attackers in order to engage them and squander their resources. LLMs provide realistic responses based on human behavior, making scams less profitable. However, such automated responses need a significant amount of storage and computational power.

We provide a set of insights based on existing work in Table 1. The present pattern of published papers on the use of LLMs for cyber defense indicates that most studies are focused on the detection and protection roles of LLMs aligning with the NIST framework. However, a research gap, as shown in Figure 1, becomes evident in post-attack scenarios. Given the critical roles recovery and attack response play in the cybersecurity lifecycle, it is essential that further studies be centered around the development of innovative LLM-related solutions to maximize their potential in productive post-attack scenarios.

Table 1: Classified publications concerning the *defensive* applications of LLMs.

Paper	NIST Framework	Application	Model(s)
(Kereopa-Yorke, 2023)	Identify	LLMs enhance cybersecurity policies.	ChatGPT
(He and Vechev, 2023)	Protect	Using LLMs for secure code development without compromising functionality.	SVEN (GPT-2), (CodeGen) LM
(Tann et al., 2023)	Protect	LLMs solve Capture The Flag challenges to enhance employees' awareness and knowledge.	code-cushman-001, code-davinci-001,code-davinci-002, 1-jumbo, j1-large, polycoder, gpt2-csrc
(Pearce et al., 2023)	Protect	LLMs investigate software vulnerabilities.	GPT-3.5 Turbo, Gemini, Microsoft Bing
(Charalambous et al., 2023)	Protect	LLMs investigate software vulnerabilities.	GPT-3.5 Turbo
(Yu and Martin, 2023)	Protect	Generating honeywords using LLMs.	GPT-3
(Dutta et al., 2018)	Protect	Chatbots assist security experts in identifying open ports.	Rule-based
(Vörös et al., 2023)	Protect	LLM-based URL categorization for website classification.	eXpose (Conv), BERTiny, URLTran (BERT) T5 Large, GPT3 Babbage
(Sandoval et al., 2023)	Protect	LLMs investigate code vulnerabilities.	GPT-3
(Tuor et al., 2018)	Detect	Detecting anomalous behavior in network logs with LLMs.	RNN
(Omar and Shiaeles, 2023)	Detect	Detection of vulnerabilities in software code.	GPT-2
(Gao, 2023)	Detect	SecureBERT for anomaly detection.	CyBERT, SecureBERT (RoBERTa)
(Ranade et al., 2021)	Detect	CyBERT, a domain-specific BERT model to recognize specialized cybersecurity entities.	BERT-based Natural Language Filter
(Happe and Cito, 2023)	Detect	Penetration testing with LLMs.	GPT-3.5
(Ameri et al., 2021)	Detect	CyBERT, a cybersecurity feature claims classifier.	CyBERT, GPT-2
(Bayer et al., 2022)	Detect	CySecBERT for malware detection and alert aggregation.	CySecBERT
(Bayer et al., 2022)	Detect	SecureBERT for processing and understandin cybersecurity text, specifically Cyber Threat Intelligence (CTI).	SecureBERT
(Ferrag et al., 2023a)	Detect	Detection of vulnerabilities in software code.	SecureFalcon (FalconLLM)
(Fengrui and Du, 2024)	Protect	TTPs Classification	GPT-3.5
(Sladić et al., 2023)	Respond	Creating honeypots related to continuously monitoring and detecting threats.	GPT-3.5 Turbo (shelLM)
(McKee and Noever, 2023)	Respond	LLM as a honeypot interface against command-line attacks.	GPT-3.5
(Garvey and Svendsen, 2023)	Detect	investigates LLMs acting as red teamers in cybersecurity.	GPT-4 & Bard
(Koide et al., 2023)	Detect	LLM for detecting phishing sites leverages a web crawler to gather information and generate prompts.	GPT-3.5 & GPT-4
(Sakaoglu, 2023)	Detect	KARTAL, a web application vulnerability detection.	GPT-3.5 Turbo
(David et al., 2023)	Detect	LLMs to perform security audits on smart contracts.	GPT-4 (GPT-4-32k), Claude-v1.3-100k
(Deng et al., 2023a)	Detect	LLM-empowered automatic penetration testing tool.	PentestGPT (GPT-3.5 & GPT-4)
(Jiang et al., 2024)	Detect	Log parsing framework.	GPT-3.5
(Chen et al., 2023)	Detect	LLMs to perform security audits on smart contracts.	GPT-3.5 Turbo & GPT-4
(Cambiaso and Caviglione, 2023)	Respond	Replying to the scam emails using LLM.	GPT-3

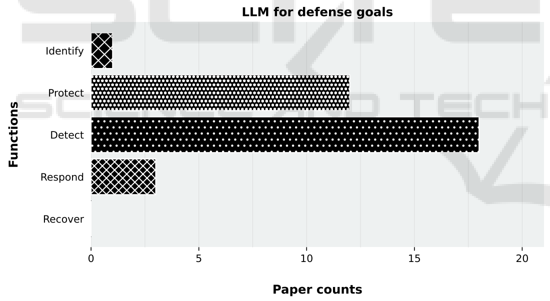

Figure 1: The present bar chart illustrates the distribution of studies mapped to each of the five elements of the NIST Cybersecurity Framework. Collected statistics indicate that the vast amount of studies are related to Protect and Detect functions emphasizing research gaps related to Identify, Respond and particularly Recover functions over collected publications.

4 ADDVERSARIAL APPLICATION OF LLMs

Applications of LLMs in cybersecurity extend beyond techniques for defense. In our exploration, we review LLMs' capacity to come up with sophisticated attacks. To this end, our approach involves with analyzing these approaches through the MITRE attack framework, which outlines various attacker tactics.

4.1 Reconnaissance

During a reconnaissance attack, adversaries actively or passively collect information about their target organization in order to identify upcoming operations (Xiong et al., 2022). Hazell (Hazell, 2023) provides an illustration of how LLMs assist during the reconnaissance stage by automating the data collection and analysis of potential victims. As a result, LLMs develop Python scripts to scrape websites that hold the desired information about users. Comparably, Salewski et al. (Salewski et al., 2023) enabled the LLMs to assume various roles by introducing the prompt with "If you were a persona", in which the target individual is substituted for the persona.

4.2 Initial Access

The initial access tactic includes the procedures adopted by attackers to obtain access as a foothold to a company's infrastructure (Xiong et al., 2022). Roy et al. (Saha Roy et al., 2023) highlight the role of LLMs in delivering malicious scripts where the attack structure is divided into four steps. In this regard, design objects are used to create concepts that are influenced by specific organizations, while credential-stealing objects are used to establish objects that require credentials, including login buttons or input fields. Credential Transfer objects are used to create functions that can provide the attacker with the credentials submitted on phishing websites. Lastly, the exploit generation object serves to implement a functionality based on the evasive exploit. The authors (Saha Roy et al., 2023) conduct a number of attacks, including text encoding, clickjacking, polymorphic URL, and QR code-based multi-stage attacks, to show how LLMs have the potential to be leveraged to generate a variety of phishing attack forms.

According to Hazell et al. (Hazell, 2023), LLMs are able to assist during the reconnaissance stage of a spear phishing attack, a process when attackers get sensitive information about their targets in order to develop compelling messages. According to John et al. (John and Philip, 2018), ML-based techniques group people according to their value and level of participation, and then utilize the timeliness of the target users to provide content and a phishing URL. Since people can adopt different personas in daily life and choose a variety of terms for a variety of circumstances, Kreps et al. (Kreps et al., 2022) discuss how GPT2 can manipulate target users' beliefs by generating stories, while Salewski et al. (Salewski et al., 2023) investigate the role of LLMs on various personas and adapt their language accordingly a process known as in-context impersonation. Based on LLMs ability to impersonate certain personalities, Salewski et al. (Salewski et al., 2023) concluded that LLMs can be applied to develop more effective phishing messages or social engineering attacks. With a dataset of phishing emails, Karanjai (Karanjai, 2022) investigates the effectiveness of generating convincing phishing emails with GPT2, GPT-3, and LSTM while taking into account the removal of HTML elements, URLs, and email addresses as well as tokenizing the text into words.

PassGPT, an LLM-based approach to password generation and modeling for password estimation, is presented by Rando et al. (Rando et al., 2023). PassGPT presents the idea of guided password generation, enabling the generation of passwords that adhere to established standards. Moreover, PassGPT, trained on password leaks, models each token independently, a character-by-character search space exploration in which generated passwords are sampled according to random restrictions.

The application of LLMs, particularly ChatGPT and AutoGPT, in malware generation is covered by Pa Pa et al. (Pa Pa et al., 2023). To determine if Auto-GPT minimizes the obstacle to malware generation, the authors (Pa Pa et al., 2023) investigated Auto-GPT running locally and tested it in the following manners: initially, by providing broad prompts like ”write a malware X," and next, by giving more specific malware and attack tool functionalities. Finally, additional tests have been explored to discover whether Anti-Virus (AV), Endpoint Detection and Response (EDR), and VirusTotal (VT) detect the generated malware.

4.3 Execution

Procedures resulting in adversary-controlled executable operating on a local or remote system are referred to as execution (Xiong et al., 2022). Using code generation tools to develop malware is one of the strategies employed by adversaries. The feasibility of employing large textual models to automatically generate malware along with the model's constraints when generating actual malware samples is studied by Botacin (Botacin, 2023). According to their findings, certain malware versions were recognized by all antivirus engines while others were not detected by any of the engines due to the use of LLMs to modify all or part of the malware's building blocks. The prompt engineering essential to develop malware that hides a PowerShell and schedules its daily execution at a given time was brought to light by Charan et al. (Charan et al., 2023). In addition to copying the CMD file to a designated directory and getting the scheduled task information as a successful malware verification, the script adds a registry value that will be run at system startup. The LLM-based malware is assessed by Pa pa et al. (Pa Pa et al., 2023). The authors (Pa Pa et al., 2023) reported that a number of the commercially available antivirus applications and Endpoint Detection and Response (EDR) solutions failed to detect the LLM-generated executables since some LLM-generated functions can establish connections toward attackers through the victim's machine (Beckerich et al., 2023).

4.4 Defense Evasion

The concept of defense evasion outlines the tactics attackers employ in order to prevent detection following a security breach (Xiong et al., 2022). According to Chatzoglou et al. (Chatzoglou et al., 2023), LLMs develop turnkey malware which lets adversaries evade antivirus and endpoint detection and response systems aiming to autonomous malicious code development. Process injection, multiprocessing, junk data, shellcode mem loading, encryption, and chosen shell code were among the techniques employed in their investigation. According to Chatzoglou et al. (Chatzoglou et al., 2023) LLMs establish an initial TCP listener

Table 2: Classified publications concerning the *adversarial* applications of LLMs.

Paper	MITRE Tactic(s)	Application	Model(s)
(Charan et al., 2023)	Execution	Generating code to perform actions that could be malicious	GPT-3
(Karanjai, 2022)	Initial Access	Generate phishing emails to bypass spam filters	GPT-2, GPT-3, RoBERTa
(Beckerich et al., 2023)	Execution - Command & Control	Use of LLMs as plug-ins to act as a proxy	GPT-4
(Saha Roy et al., 2023)	Initial Access - Collection	Generate Phishing Website via ChatGBT	GPT-3.5 Turbo
(Botacin, 2023)	Execution	Code generation and DLL injection	GPT-3
(Hazell, 2023)	Initial Access - Reconnaissance	Collecting victim data to develop an attack email	GPT-3.5, GPT-4
(Pa Pa et al., 2023)	Initial Access - Execution - Defense Evasion	Crafting malicious scripts	GPT-3.5 Turbo, GPT-4, text-davinci-003
(John and Philip, 2018)	Initial Access	Spear Phishing link	AWD-LSTM
(Chatzoglou et al., 2023)	Defense Evasion	Code obfuscation, file format modification	GPT-3.5
(Rando et al., 2023)	Initial Access - Credential Access	Password guessing using LLMs	GPT-2
(Salewski et al., 2023)	Initial Access - Reconnaissance	Impersonation for phishing aims	GPT-3.5 Turbo
(Kreps et al., 2022)	Initial Access	Generating content for misinformation	GPT-2

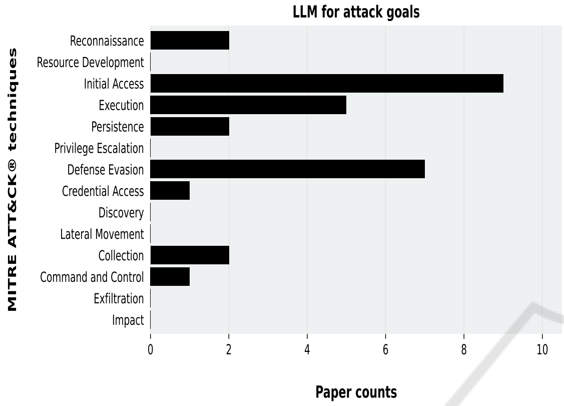

Figure 2: Concentration of recently published papers on attack approaches using LLM.

that resembles an SSH listener. This will let an attacker to connect and use Windows native APIs to execute Command Prompt (cmd) instructions. An open firewall port is required for the listener to function properly. Only three of the twelve antivirus applications were able to identify malware, according to the author's findings (Chatzoglou et al., 2023).

The study conducted by Pa Pa et al. (Pa Pa et al., 2023) assesses the effectiveness of malware scanners in detecting both obfuscated and non-obfuscated forms of code generated by LLMs. In contrast to LLM-based commonly used obfuscation techniques including base64 encoding or variable and function name modification, the authors (Pa Pa et al., 2023) demonstrated that generated non-obfuscated malware featured a reduced detection rate.

The use of evasive approaches by LLMs to evade detection by anti-phishing organizations is highlighted by Roy et al. (Roy et al., 2023). This study illustrates how LLMs assist attackers via clickjacking, fingerprinting browsers, or encoding content. Accordingly, the content of the phishing website is masked using these tactics, making it more challenging for automated anti-phishing crawlers to identify malicious information.

4.5 Credential Access

Approaches to get credentials through key-logging or credential dumping from a compromised machine refer to credential access (Xiong et al., 2022). Introduced by Rando et al. (Rando et al., 2023), PassGPT is an LLM-based password modeling solution. PassGPT uses GPT-2 architecture to estimate password strength and guess passwords. Additionally, the authors (Rando et al., 2023) analyze the probability distribution through passwords defined by PassGPT. In light of this, PassGPT delivers guided password generation, enabling constraints to choose character level randomization for the search space by setting parameters like password length or fixed characters with complete control over each character.

4.6 Collection

Collection refers to gathering information related to the attackers goals (Xiong et al., 2022). Methodologies that demonstrate how LLMs assist in gathering user data are covered by Roy et al. (Roy et al., 2023). The authors (Roy et al., 2023) investigate the applicability of LLMs in the design of credential taking objects with generating input forms. Furthermore, LLMs have the capability to distribute iFrame injection code to launch malicious websites within an official page. Roy et al. (Roy et al., 2023) demonstrate a scam attack implemented via ChatGPT to gather information without direct attempt aimed at automated data collection. The presented scam item has a hidden iFrame associated with a malicious as well as fake Amazon webpage, guaranteeing that the iFrame object does not activate any anti cross site scripting.

4.7 Command and Control

Attacks known as command and control arise when an attacker uses a victim channel to connect with underlying resources (Xiong et al., 2022). By leveraging LLMs for performing shell commands on a victim's resource, Beckerich et al. (Beckerich et al., 2023) demonstrate the notion of a command and control attack. In order to generate the executable and automate connection between the machine used by the victim and servers, the authors utilized an LLM-based plugin that acts as an interface for communicating with GPT-2. This method involves utilizing a connectivity feature to establish a connection to a certain website that hosts an attacker's command, followed by a query that ends in a URL. A list of valid user agents used by plugins is maintained regularly in order to mask the malicious component of the web server.

Figure 2 depicts the study trends on the use of LLMs in cyberattacks, and Table 2 provides a summary of the categorization. Figure 2 illustrates that initial access, defense evasion, and execution tactics are the primary points of concentration for the majority of attack methodologies. As a result, cybersecurity professionals must to give priority to these crucial phases while developing strategic protection methods against LLM-based attacks.

5 CONCLUSION

In this paper, we reviewed the state-of-the-art research in the applications of Large Language Models (LLMs) within the realm of cybersecurity. We demonstrated that while LLMs can provide effective solutions for strengthening defensive approaches, their potential misuse cannot be underestimated. Hence, we categorized related literature using the NIST cybersecurity framework and MITRE attack for applications of LLMs in cyberdefense and cyberattacks, respectively. Our review suggests that while there are numerous works evaluating the opportunities in defensive applications of LLMs, there is a lack of research in examining the risks of offensive applications. We hope this study paves the way for future research to assess the associated risks introduced by the rise of LLMs in cybersecurity.

REFERENCES

Addington, S. (2023). Chatgpt: Cyber security threats and countermeasures. *Available at SSRN 4425678*.

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

Ameri, K., Hempel, M., Sharif, H., Lopez Jr, J., and Perumalla, K. (2021). Cybert: Cybersecurity claim classification by fine-tuning the bert language model. *Journal of Cybersecurity and Privacy*, 1(4):615–637.

Ameri, K., Hempel, M., Sharif, H., Lopez Jr, J., and Perumalla, K. (2022). An accuracy-maximization approach for claims classifiers in document content analytics for cybersecurity. *Journal of Cybersecurity and Privacy*, 2(2):418–443.

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al. (2023). Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Bayer, M., Kuehn, P., Shanehsaz, R., and Reuter, C. (2022). Cysecbert: A domain-adapted language model for the cybersecurity domain. *arXiv preprint arXiv:2212.02974*.

Beckerich, M., Plein, L., and Coronado, S. (2023). Ratgpt: Turning online llms into proxies for malware attacks. *arXiv preprint arXiv:2308.09183*.

Botacin, M. (2023). Gpthreats-3: Is automatic malware generation a threat? In *2023 IEEE Security and Privacy Workshops (SPW)*, pages 238–254. IEEE.

Cambiaso, E. and Caviglione, L. (2023). Scamming the scammers: Using chatgpt to reply mails for wasting time and resources. *arXiv preprint arXiv:2303.13521*.

Charalambous, Y., Tihanyi, N., Jain, R., Sun, Y., Ferrag, M. A., and Cordeiro, L. C. (2023). A new era in software security: Towards self-healing software via large language models and formal verification. *arXiv preprint arXiv:2305.14752*.

Charan, P., Chunduri, H., Anand, P. M., and Shukla, S. K. (2023). From text to mitre techniques: Exploring the malicious use of large language models for generating cyber attack payloads. *arXiv preprint arXiv:2305.15336*.

Chatzoglou, E., Karopoulos, G., Kambourakis, G., and Tsiatsikas, Z. (2023). Bypassing antivirus detection: old-school malware, new tricks. *arXiv preprint arXiv:2305.04149*.

Chen, C., Su, J., Chen, J., Wang, Y., Bi, T., Wang, Y., Lin, X., Chen, T., and Zheng, Z. (2023). When chatgpt meets smart contract vulnerability detection: How far are we? *arXiv preprint arXiv:2309.05520*.

Corporation, M. (2023). Mitre attack.

Cybersecurity, C. I. (2014). Framework for improving critical infrastructure cybersecurity. *Framework*, 1(11).

David, I., Zhou, L., Qin, K., Song, D., Cavallaro, L., and Gervais, A. (2023). Do you still need a manual smart contract audit? *arXiv preprint arXiv:2306.12338*.

de Lima, V. M. A., Barbosa, J. R., and Marcacini, R. M. (2023). Learning risk factors from app reviews: A large language model approach for risk matrix construction.

Deng, G., Liu, Y., Mayoral-Vilches, V., Liu, P., Li, Y., Xu, Y., Zhang, T., Liu, Y., Pinzger, M., and Rass, S. (2023a). Pentestgpt: An llm-empowered automatic penetration testing tool. *arXiv preprint arXiv:2308.06782*.

Deng, Z., Ma, Y., Liu, Y., Guo, R., Zhang, G., Chen, W., Huang, W., and Benetos, E. (2023b). Musilingo: Bridging music and text with pre-trained language models for music captioning and query response. *arXiv preprint arXiv:2309.08730*.

Dutta, S., Joyce, G., and Brewer, J. (2018). Utilizing chatbots to increase the efficacy of information security practitioners. In *Advances in Human Factors in Cybersecurity: Proceedings of the AHFE 2017 International Conference on Human Factors in Cybersecurity, July 17- 21, 2017, The Westin Bonaventure Hotel, Los Angeles, California, USA 8*, pages 237–243. Springer.

Dutta, T. S. (2023). Fraudgpt: New black hat ai tool launched by cybercriminals.

Elkins, S., Kochmar, E., Serban, I., and Cheung, J. C. (2023). How useful are educational questions generated by large language models? In *International Conference on Artificial Intelligence in Education*, pages 536–542. Springer.

Falade, P. V. (2023). Decoding the threat landscape: Chatgpt, fraudgpt, and wormgpt in social engineering attacks. *arXiv preprint arXiv:2310.05595*.

Fawzi, S. (2023). A review of the role of chatgpt for clinical decision support systems. In *2023 5th Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pages 439–442. IEEE.

Fengrui, Y. and Du, Y. (2024). Few-shot learning of ttps classification using large language models.

Ferrag, M. A., Battah, A., Tihanyi, N., Debbah, M., Lestable, T., and Cordeiro, L. C. (2023a). Securefalcon: The next cyber reasoning system for cyber security. *arXiv preprint arXiv:2307.06616*.

Ferrag, M. A., Ndhlovu, M., Tihanyi, N., Cordeiro, L. C., Debbah, M., and Lestable, T. (2023b). Revolutionizing cyber threat detection with large language models. *arXiv preprint arXiv:2306.14263*.

Gan, C., Yang, D., Hu, B., Liu, Z., Shen, Y., Zhang, Z., Gu, J., Zhou, J., and Zhang, G. (2023). Making large language models better knowledge miners for online marketing with progressive prompting augmentation. *arXiv preprint arXiv:2312.05276*.

Gao, M. (2023). The advance of gpts and language model in cyber security. *Highlights in Science, Engineering and Technology*, 57:195–202.

Garvey, B. and Svendsen, A. (2023). Can generative-ai (chatgpt and bard) be used as red team avatars in developing foresight scenarios? *Analytic Research Consortium (ARC) August*.

Guo, K. and Wang, D. (2023). To resist it or to embrace it? examining chatgpt's potential to support teacher feedback in efl writing. *Education and Information Technologies*, pages 1–29.

Happe, A. and Cito, J. (2023). Getting pwn'd by ai: Penetration testing with large language models. *arXiv preprint arXiv:2308.00121*.

Hazell, J. (2023). Large language models can be used to effectively scale spear phishing campaigns. *arXiv preprint arXiv:2305.06972*.

He, J. and Vechev, M. (2023). Large language models for code: Security hardening and adversarial testing.

Hendriksen, C. (2023). Ai for supply chain management: Disruptive innovation or innovative disruption? *Journal of Supply Chain Management*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hsiao, Y.-P., Klijn, N., and Chiu, M.-S. (2023). Developing a framework to re-design writing assignment assessment for the era of large language models. *Learning: Research and Practice*, 9(2):148–158.

Huang, J. and Chang, K. C.-C. (2022). Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*.

Ishihara, S. (2023). Training data extraction from pre-trained language models: A survey. *arXiv preprint arXiv:2305.16157*.

Iturbe, E., Rios, E., Rego, A., and Toledo, N. (2023). Artificial intelligence for next generation cybersecurity: The ai4cyber framework. In *Proceedings of the 18th International Conference on Availability, Reliability and Security*, pages 1–8.

Jiang, L. (2024). Detecting scams using large language models. *arXiv preprint arXiv:2402.03147*.

Jiang, Z., Liu, J., Chen, Z., Li, Y., Huang, J., Huo, Y., He, P., Gu, J., and Lyu, M. R. (2024). Llmparser: A llm-based log parsing framework. *arXiv preprint arXiv:2310.01796*.

John, S. and Philip, T. (2018). Generative models for spear phishing posts on social media. In *NIPS Workshop On Machine Deception, California, USA. arXiv*.

Johnson, A. (2023). The transformative role of large language models in enterprise risk management.

Karanjai, R. (2022). Targeted phishing campaigns using large scale language models. *arXiv preprint arXiv:2301.00665*.

Kaur, R., Gabrijelčič, D., and Klobučar, T. (2023). Artificial intelligence for cybersecurity: Literature review and future research directions. *Information Fusion*, page 101804.

Kent, A. D. (2016). Cyber security data sources for dynamic network research. In *Dynamic Networks and Cyber-Security*, pages 37–65. World Scientific.

Kereopa-Yorke, B. (2023). Building resilient smes: Harnessing large language models for cyber security in australia. *arXiv preprint arXiv:2306.02612*.

Koide, T., Fukushi, N., Nakano, H., and Chiba, D. (2023). Detecting phishing sites using chatgpt. *arXiv preprint arXiv:2306.05816*.

Kosasih, E. E., Papadakis, E., Baryannis, G., and Brintrup, A. (2023). A review of explainable artificial intelligence in supply chain management using neurosymbolic approaches. *International Journal of Production Research*, pages 1–31.

Kreps, S., McCain, R. M., and Brundage, M. (2022). All the news that's fit to fabricate: Ai-generated text as a tool of media misinformation. *Journal of experimental political science*, 9(1):104–117.

Kucharavy, A., Schillaci, Z., Maréchal, L., Würsch, M., Dolamic, L., Sabonnadiere, R., David, D. P., Mermoud, A., and Lenders, V. (2023). Fundamentals of generative large language models and perspectives in cyber-defense. *arXiv preprint arXiv:2303.12132*.

Kuckelman, I. J., Paul, H. Y., Bui, M., Onuh, I., Anderson, J. A., and Ross, A. B. (2023). Assessing ai-powered patient education: a case study in radiology. *Academic Radiology*.

Latouche, G. L., Marcotte, L., and Swanson, B. (2023). Generating video game scripts with style. In *Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI 2023)*, pages 129–139.

Li, B., Mellou, K., Zhang, B., Pathuri, J., and Menache, I. (2023a). Large language models for supply chain optimization. *arXiv preprint arXiv:2307.03875*.

Li, H., Chen, Y., Luo, J., Kang, Y., Zhang, X., Hu, Q., Chan, C., and Song, Y. (2023b). Privacy in large language models: Attacks, defenses and future directions. *arXiv preprint arXiv:2310.10383*.

Liu, T. and Low, B. K. H. (2023). Goat: Fine-tuned llama outperforms gpt-4 on arithmetic tasks. *arXiv preprint arXiv:2305.14201*.

Liu, Y., Tao, S., Meng, W., Wang, J., Ma, W., Zhao, Y., Chen, Y., Yang, H., Jiang, Y., and Chen, X. (2023). Logprompt: Prompt engineering towards zero-shot and interpretable log analysis. *arXiv preprint arXiv:2308.07610*.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

McKee, F. and Noever, D. (2023). Chatbots in a honeypot world. *arXiv preprint arXiv:2301.03771*.

Naleszkiewicz, K. (2023). Harnessing llms in enterprise risk management: A new frontier in decision-making.

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Barnes, N., and Mian, A. (2023). A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.

Neupane, S., Fernandez, I. A., Mittal, S., and Rahimi, S. (2023). Impacts and risk of generative ai technology on cyber defense. *arXiv preprint arXiv:2306.13033*.

Omar, M. and Shiaeles, S. (2023). Vuldetect: A novel technique for detecting software vulnerabilities using language models. In *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*, pages 105–110. IEEE.

Pa Pa, Y. M., Tanizaki, S., Kou, T., Van Eeten, M., Yoshioka, K., and Matsumoto, T. (2023). An attacker's dream? exploring the capabilities of chatgpt for developing malware. In *Proceedings of the 16th Cyber Security Experimentation and Test Workshop*, pages 10–18.

Pandya, K. and Holia, M. (2023). Automating customer service using langchain: Building custom open-source gpt chatbot for organizations. *arXiv preprint arXiv:2310.05421*.

Pearce, H., Tan, B., Ahmad, B., Karri, R., and Dolan-Gavitt, B. (2023). Examining zero-shot vulnerability repair with large language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 2339–2356. IEEE.

Peng, B., Li, C., He, P., Galley, M., and Gao, J. (2023). Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Ranade, P., Piplai, A., Joshi, A., and Finin, T. (2021). Cybert: Contextualized embeddings for the cybersecurity domain. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 3334–3342. IEEE.

Rando, J., Perez-Cruz, F., and Hitaj, B. (2023). Passgpt: Password modeling and (guided) generation with large language models. *arXiv preprint arXiv:2306.01545*.

Rao, A., Kim, J., Kamineni, M., Pang, M., Lie, W., and Succi, M. D. (2023). Evaluating chatgpt as an adjunct for radiologic decision-making. *medRxiv*, pages 2023–02.

Rasmy, L., Xiang, Y., Xie, Z., Tao, C., and Zhi, D. (2021). Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ digital medicine*, 4(1):86.

Roy, S. S., Naragam, K. V., and Nilizadeh, S. (2023). Generating phishing attacks using chatgpt. *arXiv preprint arXiv:2305.05133*.

Saha Roy, S., Vamsi Naragam, K., and Nilizadeh, S. (2023). Generating phishing attacks using chatgpt. *arXiv e-prints*, pages arXiv–2305.

Sakaoglu, S. (2023). Kartal: Web application vulnerability hunting using large language models: Novel method for detecting logical vulnerabilities in web applications with finetuned large language models.

Salewski, L., Alaniz, S., Rio-Torto, I., Schulz, E., and Akata, Z. (2023). In-context impersonation reveals large language models' strengths and biases. *arXiv preprint arXiv:2305.14930*.

Sandoval, G., Pearce, H., Nys, T., Karri, R., Garg, S., and Dolan-Gavitt, B. (2023). Lost at c: A user study on the security implications of large language model code assistants. *arXiv preprint arXiv:2208.09727*.

Sannihith Lingutla, S. (2023). Enhancing password security: advancements in password segmentation technique for high-quality honeywords.

Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catanzaro, B. (2019). Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Shoham, O. B. and Rappoport, N. (2023). Cpllm: Clinical prediction with large language models. *arXiv preprint arXiv:2309.11295*.

Sladić, M., Valeros, V., Catania, C., and Garcia, S. (2023). Llm in the shell: Generative honeypots. *arXiv preprint arXiv:2309.00155*.

Soltan, S., Ananthakrishnan, S., FitzGerald, J., Gupta, R., Hamza, W., Khan, H., Peris, C., Rawls, S., Rosenbaum, A., Rumshisky, A., et al. (2022). Alexatm 20b: Few-shot learning using a large-scale multilingual seq2seq model. *arXiv preprint arXiv:2208.01448*.

Song, H., Xia, Y., Luo, Z., Liu, H., Song, Y., Zeng, X., Li, T., Zhong, G., Li, J., Chen, M., et al. (2023). Evaluating the performance of different large language models on health consultation and patient education in urolithiasis. *Journal of Medical Systems*, 47(1):125.

Sun, N., Ding, M., Jiang, J., Xu, W., Mo, X., Tai, Y., and Zhang, J. (2023). Cyber threat intelligence mining for proactive cybersecurity defense: A survey and new perspectives. *IEEE Communications Surveys & Tutorials*.

Tann, W., Liu, Y., Sim, J. H., Seah, C. M., and Chang, E.-C. (2023). Using large language models for cybersecurity capture-the-flag challenges and certification questions. *arXiv preprint arXiv:2308.10443*.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Tuor, A. R., Baerwolf, R., Knowles, N., Hutchinson, B., Nichols, N., and Jasper, R. (2018). Recurrent neural network language models for open vocabulary event-level cyber anomaly detection. In *Workshops at the thirty-second AAAI conference on artificial intelligence*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Vörös, T., Bergeron, S. P., and Berlin, K. (2023). Web content filtering through knowledge distillation of large language models. *arXiv preprint arXiv:2305.05027*.

Wang, H., Liu, C., Xi, N., Qiang, Z., Zhao, S., Qin, B., and Liu, T. (2023). Huatuo: Tuning llama model with chinese medical knowledge. *arXiv preprint arXiv:2304.06975*.

Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.

Xiong, W., Legrand, E., Åberg, O., and Lagerström, R. (2022). Cyber security threat modeling based on the mitre enterprise attack matrix. *Software and Systems Modeling*, 21(1):157–177.

Xu, C., Sun, Q., Zheng, K., Geng, X., Zhao, P., Feng, J., Tao, C., and Jiang, D. (2023). Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Yan, L., Sha, L., Zhao, L., Li, Y., Martinez-Maldonado, R., Chen, G., Li, X., Jin, Y., and Gašević, D. (2023). Practical and ethical challenges of large language models in education: A systematic literature review. *arXiv preprint arXiv:2303.13379*.

Yang, J., Jin, H., Tang, R., Han, X., Feng, Q., Jiang, H., Yin, B., and Hu, X. (2023a). Harnessing the power of llms in practice: A survey on chatgpt and beyond. *arXiv preprint arXiv:2304.13712*.

Yang, Q., Ongpin, M., Nikolenko, S., Huang, A., and Farseev, A. (2023b). Against opacity: Explainable ai and large language models for effective digital advertising. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 9299–9305.

Yu, F. and Martin, M. V. (2023). Honey, i chunked the passwords: Generating semantic honeywords resistant to targeted attacks using pre-trained language models. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 89–108. Springer.

Zeng, A., Liu, X., Du, Z., Wang, Z., Lai, H., Ding, M., Yang, Z., Xu, Y., Zheng, W., Xia, X., et al. (2022). Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Zhang, X. and Yang, Q. (2023). Xuanyuan 2.0: A large chinese financial chat model with hundreds of billions parameters. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 4435–4439.

Zhang, Y., Wang, Y., Cheng, F., Kurohashi, S., et al. (2023a). Reformulating domain adaptation of large language models as adapt-retrieve-revise. *arXiv preprint arXiv:2310.03328*.

Zhang, Z., Zheng, C., Tang, D., Sun, K., Ma, Y., Bu, Y., Zhou, X., and Zhao, L. (2023b). Balancing specialized and general skills in llms: The impact of modern tuning and data strategy. *arXiv preprint arXiv:2310.04945*.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Zhou, C., Li, Q., Li, C., Yu, J., Liu, Y., Wang, G., Zhang, K., Ji, C., Yan, Q., He, L., et al. (2023). A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419*.

Zhou, X. and Verma, R. M. (2022). Vulnerability detection via multimodal learning: datasets and analysis. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, pages 1225–1227.