computation

MDPI

Article

From Vulnerability to Defense: The Role of Large Language Models in Enhancing Cybersecurity

Wafaa Kasri [1,†], Yassine Himeur [2,*,†], Hamzah Ali Alkhazaleh [2], Saed Tarapiah [3], Shadi Atalla [2], Wathiq Mansoor [2] and Hussain Al-Ahmad [2]

[1] Faculty of Science and Technology, Tissemsilt University, Bougara 38000, Algeria; kasri.waf@gmail.com
[2] College of Engineering and Information Technology, University of Dubai, Academic City, Dubai 14143, United Arab Emirates; halkhazaleh@ud.ac.ae (H.A.A.); satalla@ud.ac.ae (S.A.); wmansoor@ud.ac.ae (W.M.); halahmad@ud.ac.ae (H.A.-A.)
[3] Department of Telecommunication Engineering, An-Najah National University, Nablus P.O. Box 7, Palestine; s.tarapiah@najah.edu
* Correspondence: yhimeur@ud.ac.ae
† These authors contributed equally to this work.

Abstract: The escalating complexity of cyber threats, coupled with the rapid evolution of digital landscapes, poses significant challenges to traditional cybersecurity mechanisms. This review explores the transformative role of LLMs in addressing critical challenges in cybersecurity. With the rapid evolution of digital landscapes and the increasing sophistication of cyber threats, traditional security mechanisms often fall short in detecting, mitigating, and responding to complex risks. LLMs, such as GPT, BERT, and PaLM, demonstrate unparalleled capabilities in natural language processing, enabling them to parse vast datasets, identify vulnerabilities, and automate threat detection. Their applications extend to phishing detection, malware analysis, drafting security policies, and even incident response. By leveraging advanced features like context awareness and real-time adaptability, LLMs enhance organizational resilience against cyberattacks while also facilitating more informed decision-making. However, deploying LLMs in cybersecurity is not without challenges, including issues of interpretability, scalability, ethical concerns, and susceptibility to adversarial attacks. This review critically examines the foundational elements, real-world applications, and limitations of LLMs in cybersecurity while also highlighting key advancements in their integration into security frameworks. Through detailed analysis and case studies, this paper identifies emerging trends and proposes future research directions, such as improving robustness, addressing privacy concerns, and automating incident management. The study concludes by emphasizing the potential of LLMs to redefine cybersecurity, driving innovation and enhancing digital security ecosystems.

Keywords: cybersecurity; deep learning; large language models; intrusion detection; malware detection; phishing attack detection

Academic Editor: Filippo Palombi

Received: 3 December 2024
Revised: 30 December 2024
Accepted: 4 January 2025
Published: 29 January 2025

Citation: Kasri, W.; Himeur, Y.; Alkhazaleh, H.A.; Tarapiah, S.; Atalla, S.; Mansoor, W.; Al-Ahmad, H. From Vulnerability to Defense: The Role of Large Language Models in Enhancing Cybersecurity. *Computation* **2025**, *13*, 30. https://doi.org/10.3390/computation13020030

1. Introduction

In the modern era of technology, cybersecurity has emerged as one of the most critical priorities for individuals, businesses, and governments around the world. As digital transformation continues to reshape how we live, work, and interact, the importance of safeguarding digital ecosystems cannot be overstated. The landscape of cybersecurity threats is ever-evolving and increasingly complex [1], characterized by sophisticated cyberattacks [2], large-scale data breaches, ransomware incidents, and zero-day vulnerabilities [3]. These

threats not only compromise sensitive information but also disrupt operations, damage reputations, and impose significant financial losses. The rise of advanced technologies, including artificial intelligence (AI), Internet of Things (IoT), and cloud computing, has further expanded the attack surface, creating new challenges for securing digital assets [4,5]. Consequently, addressing these threats requires innovative and proactive strategies that go beyond traditional defense mechanisms. Organizations must invest in cutting-edge technologies, robust policies, and continuous threat intelligence to enhance their resilience against cyber risks. Moreover, fostering cybersecurity awareness and collaboration among stakeholders is crucial to building a secure digital future [6,7].

In this context, the emergence of LLMs [8] represents a transformative advancement in cybersecurity, offering groundbreaking capabilities to address increasingly complex challenges and enhance cyber resilience. These advanced models, including Generative Pre-trained Transformer (GPT) [8,9] and Bidirectional Encoder Representations from Transformers (BERT) [10], have garnered significant attention due to their remarkable proficiency in understanding and generating natural language with human-like accuracy [11]. Originally designed to support a range of linguistic tasks such as language translation [12], text summarization [9] and sentiment analysis [13], language analysis, and fake news detection [14], LLMs have since evolved to play a pivotal role in various domains, including cybersecurity.

In the realm of cybersecurity, LLMs enable the automation of threat detection, real-time monitoring, and contextual analysis of cyber risks. They excel at parsing vast amounts of unstructured data to identify vulnerabilities, generate actionable insights, and predict potential attack vectors [15]. Additionally, LLMs enhance incident response by providing detailed reports, guiding mitigation efforts, and enabling faster decision-making. Their ability to comprehend nuanced language patterns also supports the detection of phishing attempts, social engineering tactics, and malicious communication. Moreover, LLMs contribute to workforce training by simulating cyber scenarios and improving understanding of evolving threats. By integrating LLMs, cybersecurity systems can achieve unprecedented efficiency and adaptability, paving the way for a more secure digital future [16].

Through a critical examination of recent advancements, case studies, and real-world applications, this review presents a comprehensive review of LLMs in cybersecurity, emphasizing their transformative role in enhancing threat detection, malware analysis, phishing detection, and policy automation. By leveraging advanced natural language processing capabilities, LLMs demonstrate significant potential in improving efficiency, adaptability, and context-driven insights in cybersecurity frameworks. This study also delves into the ethical, technical, and operational challenges associated with deploying LLMs, proposing strategies to address these limitations while highlighting future opportunities. The integration of LLMs into cybersecurity is further enriched through in-depth case studies and analyses of key applications, such as incident response, intrusion detection, vulnerability management, and social engineering detection. The findings underscore the importance of continuous innovation and robust frameworks to harness the full potential of LLMs while mitigating risks such as bias, adversarial attacks, and scalability constraints. Overall, the main contributions of this review are summarized as follows:

- Explores the integration of LLMs in key cybersecurity tasks such as threat detection, phishing detection, and incident response.
- Discusses ethical and technical challenges, including privacy, bias, and adversarial vulnerabilities in LLM applications.
- Presents a taxonomy and evaluation of existing LLM frameworks for malware analysis, intrusion detection, and vulnerability management.
- Highlights innovative use cases like LLM-driven automation in incident response and social engineering prevention.
- Proposes strategies for future research, focusing on robustness, continual learning, and tailored educational programs for cybersecurity.

Figure 1 offers a clear and intuitive depiction of the comprehensive structure of our proposed LLM-based cybersecurity review, guiding readers through its framework. The review begins in Section 2, where we explore the foundational concepts behind LLMs. Next, Section 3 delves into the diverse applications of LLMs within the realm of cybersecurity. Section 4 shifts focus to the security policies and compliance considerations associated with LLMs. In contrast, Section 5 examines the vulnerabilities of LLMs to various cyberattacks. Building on this, Section 7 highlights the challenges involved in deploying LLMs for cybersecurity purposes. Looking ahead, Section 8 sheds light on potential research directions to enhance the performance and reliability of LLMs in this domain. Finally, Section 9 concludes the paper, summarizing the key findings and their implications.

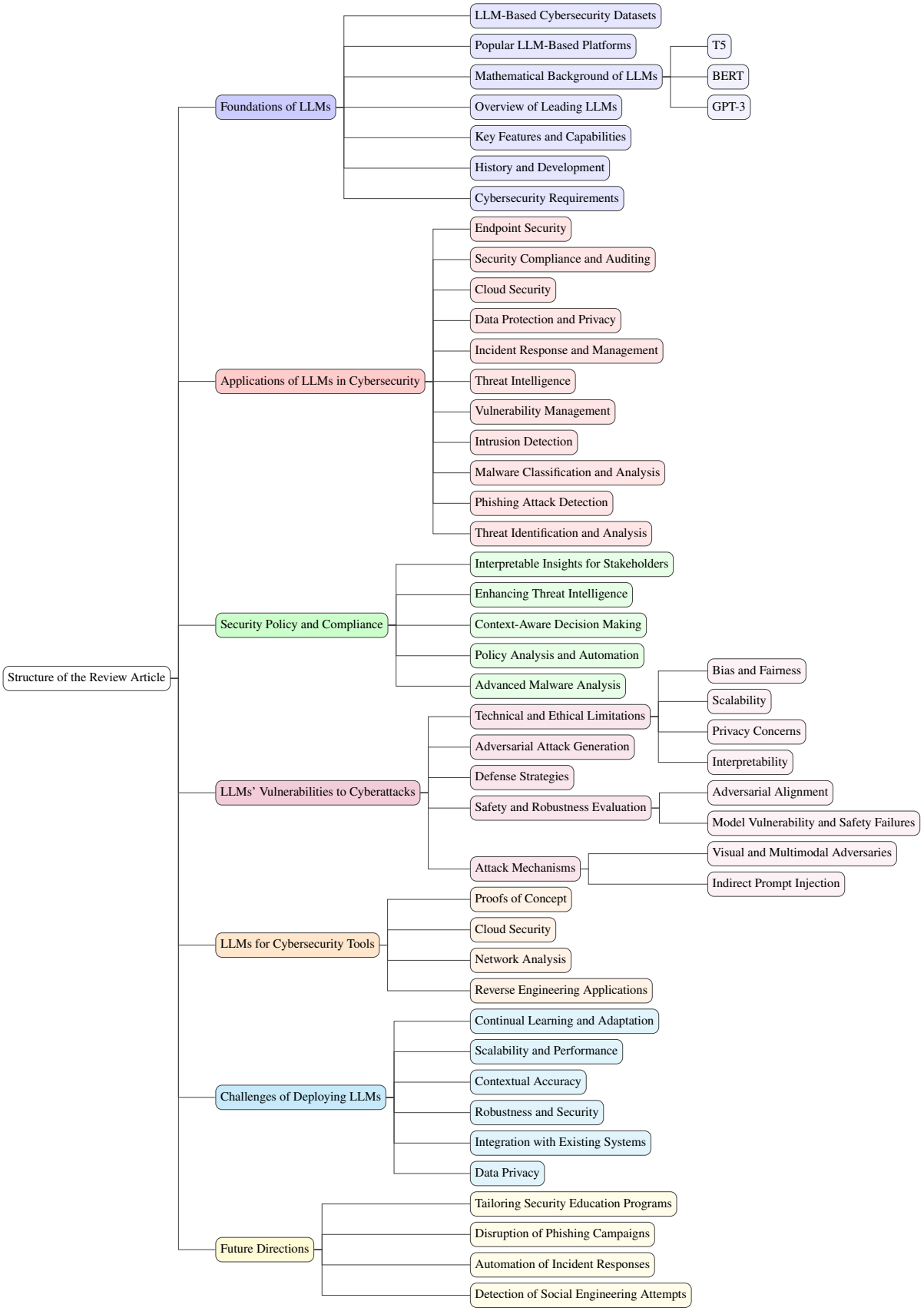

Figure 1. Structure of the proposed LLM-based cybersecurity review.

2. Foundations of LLMs

2.1. Cybersecurity Requirements

Today's interconnected digital environment presents cybersecurity with a wide range of difficulties and urgent requirements. Organizations struggle to fight against cyberattacks that could steal critical data [17], disrupt operations, and erode confidence due to the ever-evolving nature of cyber threats and the sophisticated strategies used by bad actors [18]. Ransomware, supply chain weaknesses, Advanced Persistent Threats (APTs), and Internet of Things (IoT) security issues are major concerns that necessitate ongoing attention and flexible security solutions [19]. Cloud migrations bring additional challenges, and data protection laws demand strict adherence to compliance standards. The severe lack of cybersecurity experts and the necessity of encouraging a cybersecurity-aware culture among stakeholders and employees exacerbate these problems. A diversified strategy is required to address these issues, including proactive risk management, strong threat intelligence, and raising investment. GPT-3 and other LLMs have become effective instruments for meeting particular cybersecurity demands. LLMs can be trained to detect and classify a variety of cybersecurity threats, such as malware, phishing emails, and suspicious network activity. They can also be trained to identify anomalous behavior in system logs, network traffic, or user activities by learning normal patterns and flagging deviations that may indicate potential security incidents or breaches [20]. These capabilities can help mitigate cybersecurity threats. Security analysts can concentrate on more complicated risks by using LLMs to automate repetitive security operations like analyzing security incidents, creating incident response playbooks, and prioritizing alerts. LLMs can mimic social engineering attacks or produce realistic phishing emails to teach users how to spot and react to possible dangers.

2.2. History and Development

The development of LLMs has evolved from early rule-based systems to advanced neural network-based architectures. Beginning with simple algorithms in the mid-20th century, the field has witnessed significant milestones, including the introduction of sequence-to-sequence learning and attention mechanisms, culminating in the development of transformers that have revolutionized natural language processing [20].

Figure 2 serves as a visual roadmap for understanding how LLMs can be systematically integrated into cybersecurity workflows. It highlights the synergy between cutting-edge AI models and traditional cybersecurity practices, showcasing their potential to revolutionize the detection, analysis, and mitigation of complex cyber threats. This framework underscores the importance of automation, adaptability, and continuous improvement in safeguarding digital ecosystems.

2.3. Key Features and Capabilities

LLMs are cutting-edge AI systems built on deep learning architectures. Trained on vast datasets spanning multiple domains and languages, these models excel in understanding, processing, and generating human-like text. Their versatility allows them to handle a wide range of NLP tasks—including text classification, summarization, translation, and anomaly detection—without needing task-specific fine-tuning [21,22].

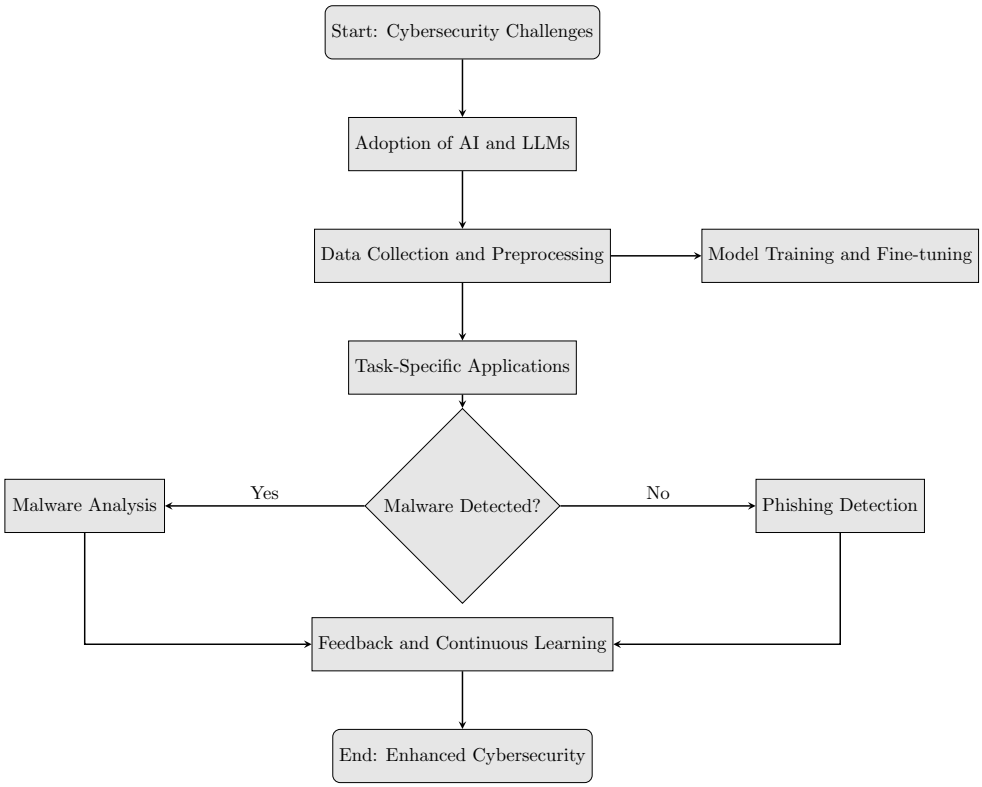

Figure 2. LLMs' integration into cybersecurity frameworks.

2.3.1. Foundational Role in Enhancing Cybersecurity

LLMs have become game changers in cybersecurity, tackling both strategic and operational challenges. Their unique capabilities make them indispensable for bolstering security frameworks across various domains. Here is a closer look at their roles [15]:

- **Threat Intelligence Augmentation:** LLMs analyze diverse data sources, such as threat reports, logs, and cybersecurity databases, to identify vulnerabilities, attack patterns, and emerging threats. By synthesizing information from different contexts, they provide actionable insights, enabling quicker and more informed decision-making [23].
- **Automated Log and Anomaly Analysis:** Security logs are often dense and complex, requiring significant effort to review manually. LLMs streamline this process by parsing logs, detecting unusual patterns, and flagging potential security incidents [24]. This not only improves detection accuracy but also reduces response times.
- **Social Engineering Prevention:** Cyberattacks frequently exploit social engineering tactics like phishing emails or deceptive messages. LLMs are adept at identifying linguistic cues that suggest phishing attempts or fraudulent activities [25]. They can alert users in real time or even automate responses to suspected threats.
- **Automated Vulnerability Assessment:** By examining technical documentation, code repositories, and system configurations, LLMs can pinpoint vulnerabilities or misconfigurations [26]. This accelerates the vulnerability discovery process and reduces dependence on manual audits.
- **Incident Response and Documentation:** During security breaches, LLMs assist by generating detailed incident reports and automating routine response tasks, such as notifying stakeholders or outlining containment measures [27]. This streamlines incident management and minimizes response delays.
- **Cybersecurity Training and Awareness:** LLMs enhance training programs by creating realistic, context-specific simulations, such as tailored phishing scenarios [28]. This

approach boosts employee awareness of evolving attack methods and improves overall cybersecurity posture.

- **Policy Generation and Risk Assessment:** LLMs can review regulatory requirements and compliance frameworks to identify gaps or inconsistencies [29]. They also generate detailed risk assessments by correlating information from various sources, helping decision-makers act on potential threats proactively.

2.3.2. Interpretable and Context-Aware Decision-Making

A major strength of LLMs in cybersecurity lies in their ability to understand context and provide human-readable explanations for their decisions. Unlike rigid rule-based systems, LLMs use their contextual knowledge to distinguish between benign and malicious activities. This interpretability fosters trust and enhances usability for cybersecurity professionals [30].

2.3.3. Bridging the Gap Between Domains

LLMs serve as bridges between different cybersecurity domains. For instance, they integrate threat intelligence with vulnerability management, synthesizing insights from varied datasets to present a holistic view of an organization's security posture. This adaptability makes LLMs critical for building interconnected, efficient cybersecurity ecosystems [31].

2.3.4. Continuous Learning and Adaptability

As cyber threats evolve, LLMs remain resilient through fine-tuning and retraining with domain-specific data [32]. This capacity for continuous learning ensures they stay relevant and effective, making them future-proof solutions for the ever-changing cybersecurity landscape.

2.4. Overview of Leading LLMs

LLMs have undergone significant evolution over the past decade, with several models leading advancements in NLP and related fields. Among these, OpenAI's GPT series stands out, particularly with GPT-3 and its successors [33]. GPT-3, built with 175 billion parameters, showcases exceptional capabilities in understanding and generating human-like text. Its applications range from conversational agents to creative writing, programming assistance, and beyond, setting a new standard for contextual coherence and semantic understanding in text generation [34]. Google's BERT (Bidirectional Encoder Representations from Transformers) has revolutionized the domain of language comprehension tasks. By employing a bidirectional training approach on vast amounts of text, BERT captures nuanced contextual relationships in language [35]. Derivatives of BERT, such as RoBERTa (Robustly Optimized BERT Approach) and T5 (Text-To-Text Transfer Transformer), have further refined these capabilities. RoBERTa enhances pretraining strategies, while T5 adopts a unified framework to handle a variety of NLP tasks by reframing them into text-to-text problems [36].

Other noteworthy LLMs include Microsoft's Turing-NLG, which is renowned for its vast scale and generative abilities, and OpenAI's Codex, designed specifically for programming-related tasks like code completion and debugging [37]. Additionally, Meta's LLaMA (Large Language Model Meta AI) focuses on efficiency, achieving high performance with fewer parameters compared to other LLMs [38]. These LLMs have paved the way for transformative applications in text generation, summarization, translation, and more [39]. They have also laid the foundation for specialized adaptations in fields such as cybersecurity, healthcare, education, and software development, where domain-specific fine-tuning enables enhanced utility and performance [40].

2.5. Mathematical Background

The mathematical foundation of LLMs primarily revolves around neural networks, particularly transformer architectures. These models leverage self-attention mechanisms to process sequences of text, allowing them to weigh the importance of different words within a sentence or document. Key mathematical concepts include vector representations of words (embeddings), positional encoding to maintain the sequence order, and the use of multi-head attention to enable the model to focus on different parts of the input sequence for prediction tasks. Training involves adjusting millions, or even billions, of parameters through gradient descent to minimize a loss function, typically cross-entropy for language tasks.

2.5.1. GPT-3

The self-attention mechanism in the GPT-3 model is mathematically represented as [41]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

where Q, K, and V represent the queries, keys, and values matrices derived from the input embeddings, respectively, and d_k denotes the dimensionality of the keys. The softmax function ensures the attention weights across the input sequence sum to 1.

2.5.2. BERT

BERT uses a transformer architecture with a unique twist: it processes words in relation to all the other words in a sentence, rather than one at a time. This is achieved through the attention mechanism, which in BERT's case, is bidirectional. Mathematically, this involves calculating attention scores for each word, using a softmax function to weigh these scores, and then applying them to produce contextually enriched word embeddings. The core formula involves the softmax of the dot product of query and key vectors, divided by a scaling factor, influencing the final output embeddings [10].

2.5.3. T5

The T5 model, Text-to-Text Transfer Transformer, employs the encoder–decoder architecture of the transformer. It is distinctive for treating every NLP task as a text-to-text problem, using a unified approach for both inputs and outputs. The core mathematical operations involve self-attention mechanisms similar to other transformers, calculating attention scores to determine the relevance of different parts of the text to each other. It also uses cross-attention mechanisms in the decoder, allowing it to focus on relevant parts of the encoder's output. The model is trained on a "span corruption" objective, predicting missing parts of the input text, which generalizes well across different types of NLP tasks [42].

2.6. Popular LLM-Based Platforms

LLMs have transformed the field of NLP, enabling a wide range of applications across industries. Table 1 provides a comparative overview of some of the most prominent LLM platforms, highlighting their developers, primary use cases, parameter sizes, and accessibility. This diversity reflects the growing demand for LLMs in tackling tasks such as text generation, natural language understanding, sentiment analysis, multilingual processing, and programming-related functions [8,10]. The applications of LLMs are as varied as the platforms themselves. Advanced models like GPT-4 (OpenAI) [8] and PaLM (Google) [43] support a wide range of generative tasks, including creative writing, summarization, and problem-solving. Other models, such as BERT (Google) [10] and Falcon 180B (Technology Innovation Institute) [44], are optimized for NLP-specific tasks like senti-

ment analysis and commercial use. Meanwhile, community-driven platforms like Vicuna 13-B [45] address domain-specific needs in customer service, healthcare, and education.

The size of these models varies significantly, reflecting their intended scope and complexity. For example, PaLM boasts an impressive 540 billion parameters [43], making it one of the largest LLMs, while lighter models such as XGen-7B [46] are designed for specific tasks with fewer parameters, enabling faster processing and deployment in constrained environments. Accessibility is another critical factor that differentiates LLM platforms. Open-source models, including BLOOM [47], Mistral 7b [48], and GPT-NeoX [49], provide unrestricted access, fostering collaboration and innovation within the research community. In contrast, restricted models like GPT-4 [8], LLaMA 2 [50], and Turing-NLG [51] are proprietary and require licensing or authorization for use, reflecting their developers' focus on controlled deployment and proprietary advancements.

The developer landscape showcases contributions from both tech giants like OpenAI, Google, and Microsoft and open-source initiatives such as Hugging Face and EleutherAI. This blend of corporate and community-driven efforts illustrates a dynamic ecosystem where innovation thrives, addressing challenges across research, industry, and everyday digital interactions. By comparing these platforms, it becomes clear how LLMs are tailored to specific needs, balancing factors like accessibility, scalability, and performance. Their continued evolution promises further advancements in AI, reshaping how we interact with technology across various domains [8,43,47].

Table 1. Comparison of LLMs.

LLM Platform	Developer (Company)	Applications/Tasks	Number of Parameters	Accessibility
LLaMA 2	Meta	Generative text model, chatbot, programming tasks	7 to 70 billion	Restricted
BLOOM	Hugging Face	Text continuation, multilingual text generation, programming	176 billion	Open-source
BERT	Google	Natural language processing tasks (e.g., sentiment analysis, clinical note analysis)	Varies (several models)	Open-source
Falcon 180B	Technology Innovation Institute (UAE)	Various NLP tasks, research, commercial use	180 billion	Restricted
OPT-175B	Meta	Research use cases	125 M to 175 billion	Restricted
XGen-7B	Salesforce	Long context window text generation, commercial and research use	7 billion	Restricted
GPT-NeoX	EleutherAI	Text generation, sentiment analysis, research, marketing	20 billion	Open-source
GPT-J	EleutherAI	Text generation, sentiment analysis, research, marketing	6 billion	Open-source
Vicuna 13-B	ShareGPT Community	Conversational AI, customer service, healthcare, education, finance	13 billion	Restricted
Mistral 7b	Mistral AI	Text generation, sentiment analysis, coding, spam detection, chatbots	7.3 billion	Open-source
GPT-3.5	OpenAI	Text generation, coding, customer support, creative content	175 billion	Restricted
GPT-4	OpenAI	Wide range of generative tasks including writing, explaining, summarizing, translating	Over 500 billion	Restricted
Turing-NLG	Microsoft	Text generation, summarization, language translation	17 billion	Restricted
PaLM	Google	Problem solving, conversation, summarization	540 billion	Restricted

2.7. LLM-Based Cybersecurity Datasets

Benchmark datasets are indispensable in evaluating LLMs within cybersecurity, offering a multifaceted approach to assess their knowledge and capabilities. They provide a standardized evaluation framework, ensuring fair comparison across models while covering a comprehensive range of cybersecurity concepts. These datasets incorporate real-world data to ensure relevance and test LLMs' contextual understanding and reasoning, crucial for identifying threats and suggesting countermeasures. Moreover, they highlight knowledge gaps and facilitate incremental improvements by serving as benchmarks for progress, ultimately fostering model transparency and building trust among users. The critical role of benchmark datasets extends beyond assessment to driving advancements in LLM capabilities, ensuring they meet the practical demands of cybersecurity challenges effectively.

The evaluation of LLMs in cybersecurity relies on diverse and specialized datasets tailored to benchmark their capabilities across various tasks. CyberMetric is a comprehensive dataset comprising 10,000 questions drawn from standards, certifications, research papers, books, and other cybersecurity publications. It serves as a robust tool for assessing LLMs' general knowledge in topics like cryptography, reverse engineering, and risk assessment [52]. Similarly, SecQA provides a question-answering dataset specifically designed to evaluate LLMs' understanding of computer security. The multiple-choice questions in SecQA, generated using GPT-4, are based on the "Computer Systems Security: Planning for Success" textbook and focus on testing the models' ability to apply core security principles effectively [53].

In addition to theoretical benchmarks, practical datasets like the NYU CTF Dataset and LLMSecEval focus on real-world applications of LLMs in cybersecurity. The NYU CTF Dataset aggregates a variety of Capture the Flag (CTF) challenges from prominent competitions, enabling the evaluation of LLMs in solving offensive security tasks such as vulnerability detection and mitigation [54]. On the other hand, LLMSecEval provides 150 natural language prompts describing code snippets prone to vulnerabilities identified in MITRE's Top 25 Common Weakness Enumeration (CWE). By including secure implementation examples, LLMSecEval facilitates the comparative evaluation of code produced by LLMs, making it a valuable resource for analyzing the models' proficiency in code security [55]. Together, these datasets provide a holistic framework for assessing LLMs in both theoretical and practical cybersecurity contexts.

3. Applications of LLMs in Cybersecurity

The integration of LLMs in cybersecurity operations enables organizations to enhance threat detection, incident response, and overall security posture. By leveraging the natural language processing capabilities of LLMs, organizations can analyze large volumes of data, extract valuable insights, and take proactive measures to mitigate cyber threats, ultimately strengthening their defenses against evolving security challenges [56].

By automating incident response processes, evaluating security alarms, and providing security teams with insightful information, LLMs are essential to enhancing security operations. Using past data and industry best practices, they assist in prioritizing issues, looking into security warnings, and developing incident response playbooks. LLMs let enterprises to react to security problems quickly and effectively, lessening the impact of breaches and interruptions, by enhancing the capacities of human analysts.

Threat intelligence analysis, vulnerability management, and policy enforcement are just a few of the areas where LLMs improve security operations [57]. They can identify software and system vulnerabilities, analyze and contextualize threat information feeds, and enforce security policies in accordance with organizational and legal demands. LLMs empower security teams to proactively identify and mitigate security threats by automating

typical security processes and delivering real-time information, hence strengthening the overall security posture.

LLMs continuously learn from new data and feedback, enabling them to adapt to evolving cyber threats and changing business environments. Through the use of ML techniques, LLMs can improve their efficiency by refining their models to better detect and respond to emerging threats [58]. This agility allows organizations to stay ahead of cyber adversaries and effectively defend against evolving attack strategies.

Ferrag and al [59] introduce SecurityBERT, an innovative architecture utilizing the BERT model for cyber threat detection in IoT networks. SecurityBERT employs a novel privacy-preserving encoding technique called Privacy-Preserving Fixed-Length Encoding (PPFLE) in combination with the Byte-level Byte-Pair Encoder (BBPE) Tokenizer to structure network traffic data effectively. The model outperforms traditional ML and DL methods, such as CNNs and RNNs, in identifying cyber threats. Using the Edge-IIoTset dataset, SecurityBERT achieved a remarkable 98.2 overall accuracy in detecting fourteen different attack types, exceeding previous benchmarks set by hybrid models like GAN-Transformer and CNN-LSTM architectures. With an inference time of less than 0.15 s on an average CPU and a compact model size of 16.7 MB, SecurityBERT is well-suited for real-time traffic analysis and deployment on resource-constrained IoT devices. Figure 3 presents a high-level workflow of the SecurityBERT model.

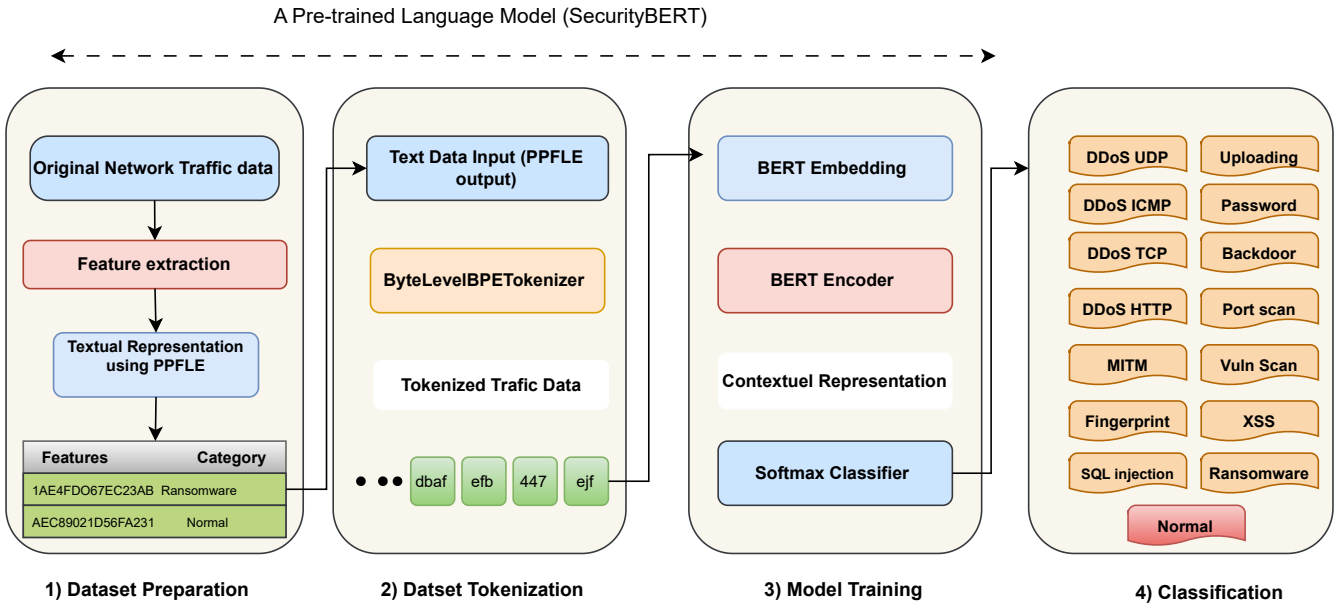

Figure 3. High-level workflow of SecurityBERT model.

3.1. Threat Identification and Analysis

The ability of LLMs to analyze broad amounts of textual data from different sources [41,60] is a game changer in cybersecurity Traditional cybersecurity approaches often struggle to keep pace with the rapidly changing threat landscape, including emerging threats, vulnerabilities and attack patterns [61]. However, LLMs offer a powerful solution through their natural language understanding capabilities to analyze unstructured data. By monitoring security blogs, forums, social media platforms, and other textual data sources, LLMs can identify patterns, trends, and irregularities that could indicate potential cyber threats. They can sift through large amounts of data in real time, extracting information relevant and have provided valuable insights for security researchers. This allows organizations to proactively identify and manage cyber risks, rather than reacting to incidents after they happen. In addition, LLMs help enhance threat reporting capabilities by detecting

emerging threats and vulnerabilities at an early stage [20]. By carefully analyzing logs from various sources, LLMs can identify signs of compromise, malicious activity, and attack techniques used by cyber adversaries This enables security researchers to deploy them to prioritize work, efficiently allocate resources, and develop strategies to protect against evolving cyber threats.

3.2. Phishing Attack Detection

LLMs have demonstrated significant potential in enhancing phishing detection, revolutionizing how cybersecurity threats are identified and mitigated. By leveraging their vast training on diverse datasets, LLMs can adeptly recognize and classify phishing content, from deceptive emails to fraudulent websites. Their ability to understand context and nuances in language enables them to detect sophisticated phishing attempts that might elude traditional detection systems. Furthermore, the integration of LLMs into phishing detection tools allows for continuous learning and adaptation to new and evolving threats, offering a dynamic defense mechanism that can significantly reduce the success rate of phishing attacks. This adaptability, coupled with high accuracy rates in detecting phishing indicators, underscores the transformative impact of LLMs in the realm of cybersecurity.

In this regard, several studies have been proposed in the literature to explore the use of LLMs for phising detection. Trad et al. [62] and Koide et al. [63] both delve into the utilization of LLMs for detecting phishing URLs and emails, respectively. Trad et al. compare the effectiveness of prompt-engineering and fine-tuning LLMs for phishing URL detection, finding that while prompt-engineered LLMs are quick and fairly effective, fine-tuned models surpass them significantly in performance. On the other hand, Koide et al. introduce ChatSpamDetector, which leverages LLMs to provide both detection and explanatory reasoning behind the classification of emails, enhancing user trust and understanding of phishing threats.

Chataut et al. [64] and Lee et al. [65] further the discussion by assessing the capabilities of LLMs against sophisticated phishing techniques. Chataut et al. test various LLMs against a curated dataset of phishing and legitimate emails, illustrating the nuanced capabilities and limitations of these models in real-world applications. Lee et al. approach the problem from the angle of brand impersonation in phishing webpages, using multimodal LLMs to detect discrepancies between webpage content and known brand characteristics, showing high efficacy in identifying phishing attempts.

In another vein, Patel et al. [66] and Roy et al. [67] explore the generative capabilities of LLMs in creating and detecting phishing content. Patel et al. evaluate the ability of several LLMs to detect "419 Scam" emails, highlighting the high accuracy of models like ChatGPT 3.5 in phishing detection. Roy et al. develop PhishLang, a lightweight LLM that excels in detecting phishing URLs with high accuracy and efficiency, demonstrating its utility in practical, resource-constrained environments.

Moreover, Bethany et al. [68] and Mahendru et al. [69] address the organizational impact and the broader implications of phishing attacks facilitated by LLMs. Bethany et al. conduct a longitudinal study on the use of LLMs to generate phishing emails targeting a large university, proposing ML-based detection techniques that show a high F1 score in identifying such emails. Mahendru et al. assess the performance of LLMs against the Deberta V3 model in detecting phishing content across various data sources, with Deberta V3 slightly outperforming LLMs in certain scenarios.

Heiding et al. [70] investigate the effectiveness of both manually and automatically (using GPT-4) created phishing emails. It also explores the combination of GPT-4 with manual strategies (V-Triad) for crafting phishing emails and assesses their effectiveness through a red teaming approach. The study includes a detailed analysis of user responses to phishing

attempts and compares the detection capabilities of human participants with that of LLMs. Uddin et al. [71] present an optimized, fine-tuned transformer-based DistilBERT model for detecting phishing emails. The model's effectiveness is demonstrated through high precision, recall, and F1 scores, and the use of Explainable-AI (XAI) techniques like LIME and Transformer Interpret to provide insights into the model's decision-making process.

The study in [72] proposes a model that detects phishing attacks based on the text of suspicious web pages using natural language processing (NLP) and DL algorithms. The effectiveness of various DL algorithms like LSTM and GRU is compared, and the model's performance is validated through high accuracy rates. The study in [73] addresses the limitations of reference-based phishing detectors by proposing an automated knowledge collection pipeline that assembles a large-scale multimodal brand knowledge base, KnowPhish. The performance of the KnowPhish Detector (KPD), which uses this knowledge base, is evaluated, showing substantial improvements in detecting phishing webpages.

The research in [74] introduces a lightweight phishing detection algorithm that differentiates phishing from legitimate websites based solely on URLs for use in mobile devices. The performance of deep transformers like BERT and ELECTRA is tested against standard and custom vocabularies for URL-based phishing detection [74]. Wang [75] develops an LLM agent framework that dynamically fetches and utilizes online information for phishing detection, overcoming the constraints of traditional static reference-based systems. The framework demonstrates superior performance compared to existing solutions, with significant accuracy improvements.

Maneriker [76] conducts a comprehensive analysis of transformer models on the phishing URL detection task. The proposed URLTran uses transformers to enhance the performance of phishing URL detection significantly, including robustness against classical adversarial phishing attacks. The study in [77] introduces an improved phishing and spam detection model, IPSDM, based on fine-tuning the BERT family of models. It demonstrates superior classification accuracy, precision, recall, and F1 score over baseline models, addressing concerns of overfitting and class imbalance.

Lastly, Nguyen et al. [78] and Roy et al. [79] discuss the integration of LLMs in user-centric anti-phishing systems. Nguyen et al. present a framework that combines LLMs with user insights to generate meaningful anti-phishing warnings, achieving over 80% effectiveness in phishing detection. Similarly, Roy et al. evaluate the potential misuse of LLMs in generating phishing content and propose a BERT-based detection tool that effectively identifies malicious prompts, aiding in the prevention of phishing attacks generated by LLMs.

Table 2 presents a comparative analysis of various studies that investigate the effectiveness of LLMs in detecting phishing attacks. It provides detailed insights into the models used, datasets, key contributions, best performance metrics, and limitations of each study. These studies explore the use of models such as GPT-4, GPT-3.5, BERT, RoBERTa, MobileBERT, and others, evaluating their capabilities for detecting phishing emails, URLs, and webpages using different datasets and techniques like fine-tuning, prompt engineering, and multimodal approaches. Important findings from the table include the impressive performance of fine-tuned models such as DistilBERT, which achieved a high F1 score of 0.99, and KnowPhish, which uses a multimodal knowledge graph for phishing detection, obtaining an F1 score of 92.05%. Additionally, studies leveraging GPT-4 for phishing detection and content generation show high accuracy rates, with some models achieving 99.7% accuracy. However, limitations such as computational cost, data dependence, and class imbalance are prevalent, highlighting the challenges and areas for further improvement in the field. Overall, the table underscores the potential of LLMs in enhancing phishing

detection systems across various modalities, but also points out the need for optimization in terms of scalability and resource efficiency.

Table 2. Comparison of studies on LLMs in phishing detection.

Ref.	Model(s) Used	Dataset/Data Type	Main Contribution	Best Performance Value	Limitation
[62]	GPT-3.5-turbo, Claude 2, GPT-2, Bloom, Baby LLaMA, DistilGPT-2	Phishing URL dataset	Comparison of prompt-engineering and fine-tuning techniques for phishing URL detection	F1 score of 97.29%, AUC 99.56%	High computational costs for fine-tuning
[63]	GPT-4	Comprehensive phishing email dataset	ChatSpamDetector for detecting phishing emails with reasoning explanations	Accuracy: 99.70%	High resource demands for continuous usage
[70]	GPT-4, Claude, PaLM, LLaMA	Custom red-teaming dataset	Comparison of LLMs and V-Triad for phishing email generation and detection	V-Triad phishing emails click-through rate: 69–79%, Claude detection: 100%	High variance in click-through rates
[64]	GPT-3.5, GPT-4, custom ChatGPT	Phishing and legitimate emails	Comparison of different LLMs for phishing email detection	Custom ChatGPT performance higher than GPT-4	Variability in results between models
[65]	GPT-4, Claude 3	Newly collected dataset of phishing webpages	Multimodal system using LLMs for brand-based phishing detection	F1 score: GPT-4 0.92, Claude 3 0.90	Need for consistent meta-data collection
[66]	ChatGPT 3.5, GPT-3.5-Turbo-Instruct, ChatGPT	419 scam phishing email dataset	Evaluation of LLMs for phishing email detection	Confidence scores: 8–10 across models	Dependence on pre-defined criteria
[67]	MobileBERT, GPT-3.5 Turbo	Phishing website dataset	Development of lightweight PhishLang model for phishing detection	Accuracy: 96%, Precision: 95%, Recall: 96%	Limited browser extension compatibility
[68]	Custom LLMs	University infrastructure dataset	Use of LLMs for lateral phishing detection in real-world conditions	F1 score: 98.96%	Requires integration with existing infrastructure
[69]	DeBERTa V3, GPT-4, Gemini 1.5	Public phishing dataset (email, HTML, SMS, etc.)	Comparative analysis of LLMs and DeBERTa for phishing detection	DeBERTa V3 Recall: 95.17%, GPT-4 Recall: 91.04%	Challenges in fine-tuning and transfer learning
[78]	LLM-driven framework	Phishing email dataset	Human-centric framework combining LLM and user input for phishing detection	Effectiveness: 80%, no false positives/negatives	Limited dataset size
[79]	ChatGPT, GPT-4, Claude, Bard	Generated phishing email and website prompts	Analysis of LLMs' potential to create phishing content	Detection tool accuracy: 96% (websites), 94% (emails)	Vulnerabilities in LLM-generated phishing content
[71]	DistilBERT (fine-tuned)	Phishing email dataset	Optimized transformer model for phishing detection with Explainable AI (LIME, Transformer Interpret)	Precision: 0.97, Recall: 1.00, F1 Score: 0.99, Accuracy: 98.48%	Class imbalance issues, mitigated via preprocessing
[72]	LSTM, BiLSTM, GRU, BiGRU	Suspicious web page text	Proposed NLP and DL approach using GloVe embedding for phishing detection	BiGRU achieved 97.39% accuracy	Loss of semantic richness between words due to non-sequential word input
[73]	LLM-based (KnowPhish Detector)	Multimodal phishing detection dataset (TR-OP)	Developed KnowPhish for enhancing RBPD by combining multimodal knowledge graphs	F1 Score: 92.05%, Precision: 97.84%, Recall: 86.90%	Scalability issues with manually constructed knowledge base
[74]	BERT, ELECTRA, ANN	Phishing website dataset (URL-based features)	Lightweight phishing detection model suitable for mobile devices	Accuracy: 86.2% (URL-based)	URL-based features alone insufficient for phishing detection
[75]	LLM agent-based framework	Dynamic phishing detection system	LLM-based dynamic reference system for phishing detection	Accuracy: 94.5%	Dependent on external data fetching for effectiveness
[76]	URLTran (transformers)	Phishing URL dataset	Transformer-based model for phishing URL detection, with adversarial robustness	True Positive Rate (TPR): 86.80%, FPR: 0.01%	Susceptibility to adversarial attacks
[77]	BERT, RoBERTA (fine-tuned IPSDM)	Phishing and spam email dataset (balanced and imbalanced)	Fine-tuned BERT/ RoBERTA models for phishing and spam detection (IPSDM)	Accuracy: 97.50%, Precision: 0.98 (RoBERTA)	Bias towards majority class, mitigated with ADASYN sampling

3.3. Malware Classification and Analysis

LLMs can effectively analyze textual data from diverse sources, including malware reports, security blogs, technical documents, and threat intelligence feeds. This analysis enables them to extract valuable features and insights that greatly assist in the classification

and analysis of malware [58]. One of the key advantages of LLMs is their ability to learn patterns and characteristics of known malware families and attack techniques [80]. This knowledge allows them to accurately classify and categorize new malware samples, even in situations where traditional signature-based detection methods may fail. By leveraging their understanding of these patterns, LLMs can effectively identify and categorize malicious activities, thereby aiding in malware behavior analysis. They can interpret descriptions of observed behaviors and correlate them with known malware behaviors and attack patterns, providing security analysts with a deeper understanding of the threats they are dealing with. Furthermore, LLMs can also assist in the interpretation of code snippets, scripts, and command-line instructions associated with malware samples. This capability provides valuable insights into the functionality, propagation mechanisms, and potential impact of malware on targeted systems. By analyzing these aspects [81], LLMs contribute to a more comprehensive understanding of the nature and potential consequences of malware attacks.

Hu et al. [82] introduced MalGPT, a DL-based causal language model that significantly improves adversarial malware generation by enabling single-shot evasion in black-box settings. Building on the theme of improving malware detection, Sanchez et al. [83] extended the use of LLMs by integrating them with system call analysis for malware detection, highlighting the importance of context size in enhancing detection rates. Both studies emphasize the role of LLMs in addressing the challenges of detecting sophisticated cyberattacks, particularly in high-stakes environments.

Similarly, Ferrag et al. [59] contributed to the cybersecurity domain by focusing on IoT networks. They proposed SecurityBERT, a transformer-based model for detecting cyber threats in IoT devices, achieving high accuracy and low inference time, making it suitable for real-time applications. Complementing this, Demirci et al. [84] proposed Stacked BiLSTM and GPT-2 models for analyzing assembly instructions of executable files, demonstrating that DL models are also effective in detecting malware at the code level, with high F1 scores across various datasets.

Continuing the focus on enhancing malware detection techniques, Gao et al. [85] introduced a novel approach that leverages control-flow graphs (CFG) and Graph Isomorphism Networks (GIN) for malware classification, achieving impressive detection rates. Along similar lines, Zahan et al. [86] presented SecurityAI, a workflow combining GPT-3 and GPT-4 for detecting malicious code in the npm ecosystem, significantly improving over traditional static analysis techniques and highlighting the potential of LLMs in automating code review tasks.

While these studies showcase the capabilities of DL and LLMs in strengthening cybersecurity, Madani et al. [87] raised an important concern regarding the potential misuse of LLMs. They explored the risks posed by code metamorphism, proposing a framework using LLMs to generate and test next-generation metamorphic malware, emphasizing the ethical challenges associated with AI-driven code mutation. Similarly, in Android security, Khan et al. [88] proposed a multi-level technique that combines graph-based representations and LLMs to capture both high-level structural and semantic features, further enhancing malware detection in Android applications by addressing both the structural and contextual aspects of mobile threats.

Finally, Fang et al. [89] provided a comprehensive evaluation of LLMs for code analysis, demonstrating the effectiveness of GPT-4 in analyzing non-obfuscated code across multiple programming languages. This study underscores the utility of LLMs in automating code analysis tasks, while also acknowledging the challenges associated with obfuscated code, thus providing valuable insights for future research in code analysis and cybersecurity.

Algorithm 1 outlines a structured framework for leveraging LLMs in malware detection and analysis. It begins by collecting and preprocessing malware-related data, followed by feature extraction using an LLM to identify meaningful patterns. The extracted features are used to train the model for malware classification, enabling it to categorize malware into predefined types and analyze its behavior by correlating observed activities with known attack patterns. The framework also incorporates LLMs for code and command analysis to understand malware functionality and impacts. To enhance detection, it integrates LLMs with complementary methods, such as system call analysis and control-flow graphs. Performance is evaluated using metrics like accuracy and F1 score, while measures are implemented to address challenges such as adversarial samples and obfuscated code. Finally, the model is continuously updated with new data to adapt to evolving threats, providing a comprehensive and adaptive approach to malware detection. Figure 4 represents the steps of a large language model (LLM)-based Malware Detection Framework.

Algorithm 1: LLM-based Malware Detection Framework

Input: Dataset $D = \{d_1, d_2, \ldots, d_n\}$, Pretrained LLM $\mathcal{M}_{\text{LLM}}$
Output: Malware classification and behavior insights

Step 1: Data Collection and Preprocessing
Collect raw textual and technical data related to malware.
Preprocess the dataset D: $D_{\text{preprocessed}} = \mathcal{P}(D)$.

Step 2: Feature Extraction
Extract features $F = \mathcal{M}_{\text{LLM}}(D_{\text{preprocessed}})$.

Step 3: Malware Classification
Define malware classes $C = \{c_1, c_2, \ldots, c_k\}$.
Train LLM to map features to classes: $\mathcal{C} : F \to C$.
Minimize classification loss $\mathcal{L}$.

Step 4: Behavior Analysis
Identify observed behaviors B and correlate with known patterns P: $\mathcal{A} : B \to P$.

Step 5: Code and Command Analysis
Analyze code snippets $S = \{s_1, s_2, \ldots, s_p\}$: $\mathcal{F} : S \to I$.

Step 6: Model Integration
Integrate with additional methods (e.g., system calls $\mathcal{S}$, CFGs $\mathcal{G}$):

$$\mathcal{D} = \alpha \mathcal{M}_{\text{LLM}}(D_{\text{preprocessed}}) + \beta \mathcal{S}(D) + \gamma \mathcal{G}(D)$$

Step 7: Performance Evaluation
Evaluate metrics such as Accuracy, F1 Score, and Detection Rate.

Step 8: Addressing Challenges
Mitigate adversarial or obfuscated samples O: $\hat{\mathcal{M}}_{\text{LLM}} = \mathcal{R}(\mathcal{M}_{\text{LLM}}, O)$.

Step 9: Continuous Learning
Update model parameters with new data D_{new}:

$$\theta_{t+1} = \theta_t - \eta \nabla \mathcal{L}(D_{\text{new}})$$

return Malware classification, behavior insights, and improved security frameworks.

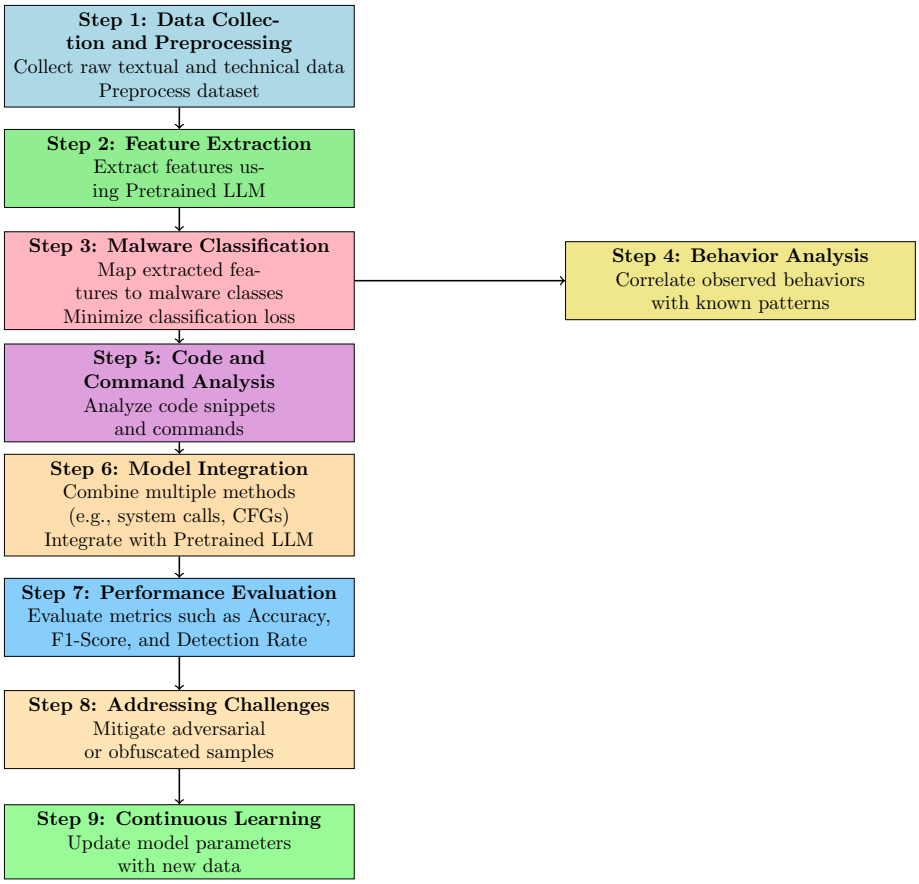

Figure 4. Steps of using LLMs for malware detection.

Table 3 presents a comprehensive comparison of various studies focused on malware detection and code analysis, highlighting the different models used, datasets or data types, main contributions, best performance values, and limitations. Key findings include the use of DL and LLMs to address complex malware detection challenges. For instance, Hu et al. [82] introduced MalGPT for adversarial malware generation, achieving a 24.51% evasion rate, while Ferrag et al. [59] proposed SecurityBERT, which achieved 98.2% accuracy in IoT threat detection. Similarly, Gao et al. [85] used a graph-based approach for malware classification, with a detection rate of 97.44%. Studies like Zahan et al. [86] and Fang et al. [89] demonstrated the effectiveness of GPT models for malicious code detection and code analysis, achieving F1scores as high as 97% with GPT-4. However, limitations such as high computational complexity, inference time constraints, and challenges in handling obfuscated code were noted across several studies. This table underscores the diversity of approaches and the promising potential of LLMs in advancing cybersecurity.

Table 3. Comparison of studies on malware detection and code analysis.

Ref.	Model(s) Used	Dataset/Data Type	Main Contribution	Best Performance	Limitation
[82]	MalGPT (DL-based causal language model)	Real-world malware dataset from VirusTotal	Single-shot evasion using MalGPT for AMG	24.51% evasion rate	Focus on black-box attacks, single-shot evasion approach
[83]	LLMs (BigBird, Longformer)	Over 1TB of system call data	Malware classification using system call data with LLMs	0.86 F1 score	Computational complexity due to large context size
[59]	SecurityBERT (BERT-based)	Edge-IIoTset cybersecurity dataset	Privacy-preserving IoT threat detection using BERT	98.2% accuracy	Inference time on resource-constrained IoT devices

Table 3. *Cont.*

Ref.	Model(s) Used	Dataset/Data Type	Main Contribution	Best Performance	Limitation
[84]	Stacked BiLSTM, GPT-2, DistilBERT	Malicious and benign PE files	Assembly instruction-based malware detection using deep language models	98.3% F1 score	Focused on assembly-level instruction analysis
[85]	GIN, MiniLM, MLP	Malware Geometric Dataset (CFG from PE files)	Malware classification using CFGs and GIN	97.44% detection rate	High complexity of CFG-based approach
[86]	GPT-3, GPT-4, CodeQL	Benchmark dataset of npm packages	LLM-based malicious code detection in the npm ecosystem	99% precision, 97% F1 score (GPT-4)	Static analysis needed as a pre-screening step
[87]	LLMs (ChatGPT, Google Bart)	Code mutation datasets	Framework for self-testing program mutation engines for malware	N/A	Ethical concerns over misuse by malware creators
[88]	Graph Convolutional Networks (GCN), LLMs	Android apps (source-level and graph-based features)	Multi-level malware detection in Android using GCN and LLMs	N/A	Complexity due to the integration of structural and semantic features
[89]	GPT-4	Non-obfuscated code (C, JavaScript, Python)	Systematic evaluation of LLMs for code analysis	97.4% accuracy	Struggles with obfuscated code

3.4. Intrusion Detection

The use of LLMs in intrusion detection represents a transformative shift in cybersecurity strategies, integrating advanced AI to address complex and evolving threats. LLMs, especially those based on the transformer architecture, are being adapted for various cybersecurity applications, from network intrusion detection to the protection of vehicular and Internet of Things (IoT) networks. Their ability to process and interpret vast amounts of unstructured data allows for enhanced detection capabilities that can identify subtle patterns and anomalies that traditional systems might overlook. Furthermore, the incorporation of these models into existing cybersecurity frameworks not only enhances detection accuracy but also improves the explainability of security alerts, thus aiding cybersecurity professionals in quick and effective decision-making. This synergy between LLMs and intrusion detection systems paves the way for more robust defenses against an increasingly sophisticated landscape of cyber threats.

The exploration of LLMs in intrusion detection is extensive, showcasing their potential through diverse approaches and methodologies aimed at enhancing network security. For instance, studies by [90,91] evaluate the application of LLMs, specifically GPT variants, in network intrusion detection systems (NIDS). While [90] assesses the feasibility and explanatory power of LLMs in detecting malicious NetFlows, ref. [91] emphasizes the efficiency of in-context learning for automatic intrusion detection, highlighting significant performance improvements without the need for further model training.

Adapting transformer-based models like BERT for cybersecurity tasks is explored in several studies. The authors in [92] introduce CAN-BERT for detecting intrusions in vehicle networks, demonstrating its superiority over traditional methods. Meanwhile, refs. [22,93] both leverage BERT's capabilities to enhance intrusion detection in Internet of Vehicles (IoV) and general network environments, respectively, showing notable advances in detection accuracy and robustness. In this regard, Li et all. [94] integrate BERT within a conditional generative adversarial network framework to address class imbalances in intrusion detection datasets, achieving high classification performance across multiple datasets.

Algorithm 2 outlines an LLM-based Intrusion Detection Framework, leveraging large language models for feature extraction, intrusion classification, and explainable insights. It integrates real-time detection, performance evaluation, and continuous learning to adapt

to evolving threats. Advanced applications, including knowledge graphs and domain adaptation, enhance its effectiveness for robust cybersecurity solutions.

Algorithm 2: LLM-based Intrusion Detection Framework

Input: Dataset $D = \{d_1, d_2, \ldots, d_n\}$, Pretrained LLM $\mathcal{M}_{\text{LLM}}$
Output: Intrusion detection insights and robust security framework

Step 1: Data Collection and Preprocessing
Collect raw intrusion-related data (e.g., network traffic, system logs, IoT data).
Preprocess D: $D_{\text{processed}} = \mathcal{P}(D)$, and balance if needed:
$D_{\text{balanced}} = \mathcal{A}(D_{\text{processed}})$.

Step 2: Feature Extraction
Extract features using LLM: $F = \mathcal{M}_{\text{LLM}}(D_{\text{processed}})$.

Step 3: Intrusion Detection Model Design
Define intrusion categories $C = \{c_1, c_2, \ldots, c_k\}$ and map features to categories:
$\mathcal{C} : F \to C$.
Adapt domain-specific models: $\mathcal{M}_{\text{ID}} = \mathcal{M}_{\text{LLM}} + \mathcal{H}$.

Step 4: Model Training and Fine-Tuning
Train model to minimize loss: $\mathcal{L} = -\sum_{i=1}^{m}\sum_{j=1}^{k} y_{ij} \log \hat{y}_{ij}$.
Apply in-context learning: $\mathcal{M}'_{\text{LLM}} = \text{Adapt}(\mathcal{M}_{\text{LLM}}, D_{\text{context}})$.

Step 5: Real-Time Detection
For incoming data $S = \{s_1, s_2, \ldots, s_t\}$, classify each sample: $\hat{y}_i = \mathcal{C}(\mathcal{M}_{\text{LLM}}(s_i))$.

Step 6: Explainability and Decision Support
Generate interpretable insights: $I = \mathcal{E}(\mathcal{M}_{\text{LLM}}, F)$.

Step 7: Performance Evaluation
Evaluate metrics: Accuracy $= \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$, F1-Score $= \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision+Recall}}$.

Step 8: Integration into Security Frameworks
Combine LLM outputs with other systems: $\mathcal{F}_{\text{Sec}} = \alpha \mathcal{M}_{\text{LLM}} + \beta \mathcal{N}_{\text{IDS}} + \gamma \mathcal{I}_{\text{IoT}}$.

Step 9: Continuous Learning and Adaptation
Update parameters with new data: $\theta_{t+1} = \theta_t - \eta \nabla \mathcal{L}(D_{\text{new}})$.
Refine using feedback: $\mathcal{M}_{\text{LLM}} = \mathcal{M}_{\text{LLM}} + \mathcal{T}(F_{\text{feedback}})$.

Step 10: Advanced Applications
Construct knowledge graph: $G = \mathcal{K}(\mathcal{M}_{\text{LLM}}, T)$.
Adapt to new domains: $\hat{D} = \mathcal{D}_{\text{Adapt}}(D, D_{\text{target}})$.

return Intrusion detection insights and updated security framework.

Further expanding the scope, Lin et al. [95] implement an LLM for intrusion detection at scale using a large volume of command-line data, which significantly outperforms conventional methods. This is echoed by [33,96], who employ LLMs to improve the security of satellite and IoT networks, respectively, by enhancing model accuracy and reducing computational demands. Moving on, the authors in [97] explore the explanatory potential of LLMs in network intrusion detection by providing understandable insights into decision-making processes, which enhances the usability of ML-based NIDS. This aspect of making AI comprehensible is critical for its acceptance and effectiveness in real-world applications.

Moreover, the study in [98] focuses on enhancing the domain adaptation capabilities of NIDS using NLP techniques and the BERT framework, demonstrating improved results on data from different domains. Similarly, Tran et al. [99] address how data quality affects

the performance of ML-based intrusion detection systems, using a series of experiments with multiple datasets and models, including BERT and GPT-2, to analyze the impact of data quality issues like duplications and overlaps.

The study in [100] proposes a bi-directional GPT model for detecting intrusions in the Controller Area Network (CAN) bus protocol, emphasizing the model's superior performance in detecting intrusions, particularly spoofing attacks, compared to traditional methods. Moving forward, Hun et al. [101] propose using a large language model to construct a knowledge graph from unstructured open-source threat intelligence for intrusion detection, highlighting the model's effectiveness in named-entity recognition and Tactic, Technique, and Procedure (TTP) classification. Finally, the syudy in [102] introduces the FlowTransformer framework, which utilizes transformer models for NIDS to capture complex network behaviors. It evaluates different transformer architectures and highlights the significant impact of the choice of classification head on performance. Figure 5 presents the CAN-BERT model; it is designed for intrusion detection in Controller Area Networks (CANs), commonly used in vehicles. It leverages the BERT language model to analyze CAN bus traffic and detect anomalies or cyberattacks. By treating CAN messages as a sequence of tokens, the model can understand the contextual relationships between them, allowing it to detect subtle anomalies indicative of intrusions. CAN-BERT outperforms traditional rule-based and ML approaches by capturing both local and global patterns in CAN traffic, making it highly effective for real-time intrusion detection in automotive systems.

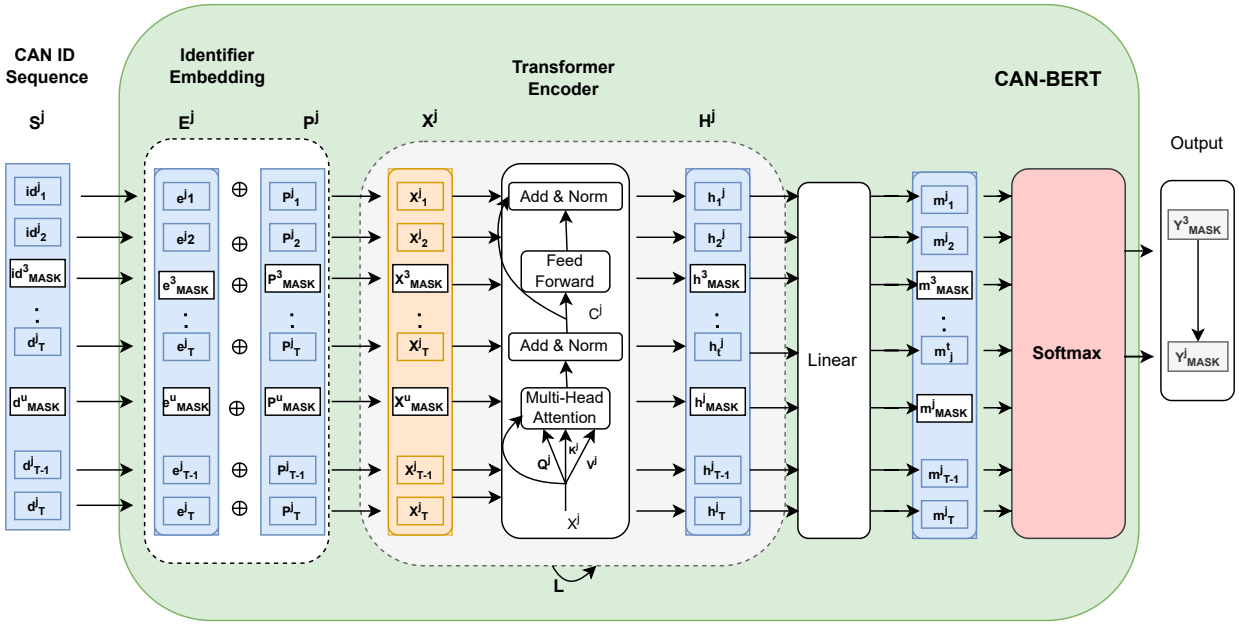

Figure 5. Flowchart of the CAN-BERT model presented in [92].

Table 4 provides a comprehensive comparison of various studies that leverage LLMs for IDS. It highlights key aspects such as the models employed, datasets used, notable contributions, performance metrics, and the limitations of each approach. A major takeaway is the impressive performance of several models, including GPT-4 and BERT-based frameworks, which achieved accuracy and F1 scores exceeding 95%. Particularly noteworthy are the IoV-BERT-IDS and CAN-BERT models, which demonstrated exceptional capabilities in addressing specific intrusion scenarios, such as those involving vehicle networks and the IoV. However, these advancements are not without challenges. Limitations include high computational requirements, difficulties in scaling, managing heterogeneous data, and addressing imbalanced datasets. These challenges highlight the complexity

and resource-intensive nature of utilizing LLMs for intrusion detection, even as their performance continues to show great promise.

Figure 6 visually represents the steps of an LLM-based Intrusion Detection Framework, a structured methodology for detecting and mitigating intrusions in a networked environment using LLMs.

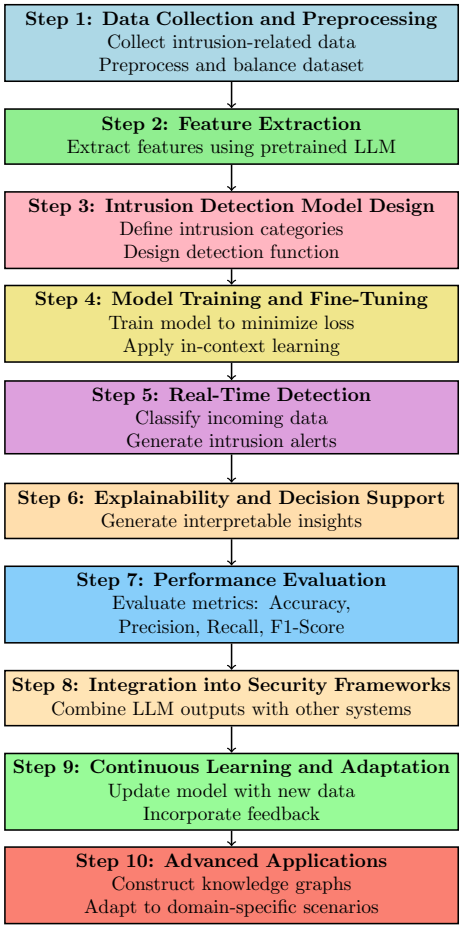

Figure 6. Steps of the LLM-based Intrusion Detection Framework.

Table 4. Comparison of LLMs-based intrusion detection studies.

Ref.	Model(s) Used	Dataset/Data Type	Main Contribution	Best Performance	Limitation
[90]	GPT-4, LLama3, KTO	NF-CSE-CIC-IDS2018-v2	Evaluation of LLMs for NIDS and explainability in threat detection	Precision: 55.02%, Recall: 58.79%	High computational requirements and limited attack detection accuracy
[91]	GPT-4	Real network intrusion dataset	Pre-trained LLM framework for automatic network intrusion detection with in-context learning	Over 95% Accuracy and F1 Score	Computational cost and need for large in-context examples
[92]	CAN-BERT (BERT-based)	Car Hacking Dataset 2020	BERT-based detection of CAN bus cyberattacks	F1 Score: 0.81–0.99	Lack of encryption and authentication in CAN protocol
[95]	LLM (command-line model)	30 M training samples, 10 M test samples	Large-scale pre-training for AI-based intrusion detection on command lines	Precision (PO@100): 100%, PO@1000: 99.8%	High demand on data and computational resources
[22]	IoV-BERT-IDS	CICIDS, BoT-IoT, IVN-IDS datasets	LLM-based IDS for IoV with bidirectional contextual semantics	High accuracy and generalization capabilities	Complexity of extracting bidirectional contextual features
[93]	BERT-based IDS	Network traffic data	BERT-based framework for enhanced feature extraction and intrusion detection	Superior accuracy and reduced false positive rates	Limited scalability in diverse environments

Table 4. *Cont.*

Ref.	Model(s) Used	Dataset/Data Type	Main Contribution	Best Performance	Limitation
[94]	CGAN, BERT	CSE-CIC-IDS2018, NF-ToN-IoT-V2	Multi-class intrusion detection using CGAN and BERT to address class imbalance	F1 Scores: 98.230%, 98.799%, 89.007%	Challenge in handling imbalanced datasets
[97]	Decision Trees, LLMs	NF-BoT dataset	LLM-based explanations for decision trees in NID systems	Accuracy: 98.7%	Requires background knowledge in ML for interpretation
[33]	PLLM-CS (Transformer-based)	UNSW_NB15, TON_IoT datasets	LLM for cyber threat detection in satellite networks	100% accuracy on UNSW_NB15 dataset	High cost of deployment and fine-tuning
[103]	XG-NID (GNN-based)	Heterogeneous graph of flow and packet data	Real-time NIDS with flow and packet-level data fusion and LLMs for explainability	F1 Score: 97%	Complexity in handling heterogeneous data and real-time analysis

3.5. Vulnerability Management

Recent studies have highlighted the transformative role of LLMs in enhancing vulnerability detection and repair across various software engineering tasks. For instance, Zhou et al. [104] conducted a systematic literature review, covering 36 papers that examine the application of LLMs in vulnerability detection and repair. Their work identifies critical challenges and outlines a roadmap for future research. Similarly, Zibaeirad et al. [105] introduced VulnLLMEval, a framework for evaluating LLMs' capabilities in identifying and patching vulnerabilities in C code, revealing that while LLMs struggle with complex vulnerabilities, they provide a robust dataset for performance assessment. Zhang et al. [106] further explored the effectiveness of LLMs, specifically ChatGPT-4 and Claude, in fixing real-world memory corruption vulnerabilities in C/C++ code, demonstrating their strengths in localized fixes but limitations with more intricate issues.

In tandem with these foundational studies, Steenhoek et al. [107] evaluated eleven state-of-the-art LLMs for vulnerability detection, emphasizing the models' challenges in accurately identifying bugs despite some success with tailored prompting techniques. This is echoed in Boi et al. [108], who proposed a novel approach for detecting vulnerabilities in smart contracts using LLMs. Their method leverages advanced NLP capabilities, achieving high detection accuracy, thereby showing potential for improving security in decentralized applications. Additionally, Jensen et al. [109] investigated the application of LLMs in aiding code reviews, finding that while they can flag vulnerabilities, their performance varies significantly based on model choice and prompting strategies.

Moreover, Mathews et al. [110] focused on Android applications, illustrating LLMs' efficacy in identifying vulnerabilities through a comprehensive AI-driven workflow. Their results indicated a high true positive rate, contributing valuable insights into practical vulnerability detection methodologies. Liu et al. [111] addressed the lack of benchmarks for assessing LLMs in vulnerability detection, introducing VulDetectBench, which reveals varying performance levels across different models. Lastly, Nong et al. [112] introduced LLMPATCH, an automated patching system utilizing LLMs, showcasing significant advancements in generating effective patches for real-world vulnerabilities without prior training. Together, these studies underscore the growing importance of LLMs in advancing security measures in software development, highlighting both their potential and the challenges that remain. Table 5 presents a comparison between several studies highlighting the accuracy of LLMs in vulnerability detection. Zibaeirad et al. introduce VulnLLMEval, which assesses LLMs in C code vulnerability patching but notes difficulties with complex vulnerabilities. Zhang et al. evaluate ChatGPT-4 and Claude, showing strengths in localized fixes for memory corruption vulnerabilities. Steenhoek et al. analyze eleven LLMs, while Boi et al. demonstrate high detection accuracy in smart contracts. Mathews et al. report a

high true positive rate for Android vulnerabilities, and Nong et al. present LLMPATCH for effective patch generation.

Table 5. Comparison of studies on LLMs for vulnerability detection and repair.

Ref.	Model(s) Used	Dataset/Data Type	Main Contribution	Best Performance Value Achieved	Limitation
[113]	Various LLMs	36 papers from SE, AI, and security venues	Systematic review of LLMs for vulnerability tasks	N/A	No existing comprehensive survey
[105]	Various LLMs	307 real-world vulnerabilities (C code)	Introduction of VulnLLMEval framework	N/A	Struggles with complex vulnerabilities
[106]	ChatGPT-4, Claude	223 real-world C/C++ code snippets	Evaluation of LLMs in fixing memory corruption	Local fixes effective, complex issues challenging	Decreased effectiveness with intricate bugs
[107]	11 state-of-the-art LLMs	Seminal code generation datasets	Evaluation of prompting techniques for vulnerability detection	0.5–0.63 balanced accuracy	Errors in model reasoning, struggled with localization
[108]	LLMs	Dataset of known smart contract vulnerabilities	Novel tool for detecting smart contract vulnerabilities	High detection accuracy	Limitations in traditional tools not fully addressed
[109]	OpenAI models, open-source LLMs	HumanEval, MBPP datasets	Evaluation of LLMs in code reviews	36.7% accurate vulnerability descriptions	Performance varies widely by model
[110]	LLMs	Ghera benchmark for Android apps	AI-driven workflow for Android vulnerability detection	91.67% true positive rate	Analysis limited to specific configurations
[111]	17 LLMs (open/closed-source)	VulDetectBench benchmark	Benchmark for assessing vulnerability detection in LLMs	Over 80% on identification tasks	Less than 30% on detailed analysis tasks
[112]	Pre-trained LLMs	Real-world vulnerable code	LLMPATCH for automated vulnerability patching	98.9% F1 score	No training/fine-tuning; relies on adaptive prompting
[114]	LLMs	Various web application datasets	Approach using LLMs for web vulnerability detection	High accuracy with minimal data	Traditional methods still face challenges

3.6. Threat Intelligence

Several recent studies have examined the use of LLMs in cyber threat intelligence (CTI) and cybersecurity, showcasing various innovative methods and approaches. Hasan et al. introduce a framework for enhancing cybersecurity at the edge devices using lightweight ML models in conjunction with LLM-driven threat intelligence. This decentralized system emphasizes real-time analysis of local data streams, reducing latency and enhancing privacy through the local processing of sensitive information. The collaborative learning features of LLMs further allow for peer-to-peer knowledge sharing among edge devices, dynamically mitigating emerging cyber threats. This framework offers scalability and flexibility, making it well suited for diverse network environments [115].

In a similar vein, Clairoux-Trepanier et al. assess the accuracy of LLMs, particularly the OpenAI GPT-3.5-turbo model, for analyzing CTI data from cybercrime forums. Their study found that LLMs were highly effective in extracting actionable intelligence, achieving an average accuracy score of 98%. However, the researchers highlight areas for improvement, such as enhancing the model's ability to distinguish between factual events and stories and improving its handling of verb tenses [116]. Complementing this, Wu et al. propose the Knowledge Graph Verifier (KGV), which integrates LLMs and knowledge graphs to assess the quality of CTI by fact-checking key claims. Their innovative framework constructs knowledge graphs using paragraphs as nodes, enhancing the semantic understanding of the LLM while simplifying the labeling process [117].

Jo et al. also contribute to the development of CTI systems with Vulcan, a neural language model-based approach for extracting static cyber threat data from unstructured text. Vulcan excels in recognizing and relating CTI entities, achieving high accuracy in named-entity recognition and relation extraction tasks. This system demonstrates potential for reducing the time and labor required for analyzing cyber threats while providing a detailed understanding of evolving threat profiles [118]. Similarly, Liu and Zhan introduce an

approach to automate the construction of CTI knowledge graphs using LLMs. Their method focuses on extracting attack-related entities and relationships, proving highly effective in low-resource scenarios and outperforming existing systems in terms of efficiency [119].

Further advancing the field, researchers such as Hu et al. propose LLM-TIKG, a system that constructs knowledge graphs from unstructured open-source CTI reports. Leveraging GPT's few-shot learning capabilities, this model automates data annotation and augmentation, achieving impressive results in named-entity recognition and attack pattern classification [101]. On the proactive defense side, researchers like Karuna et al. introduce WILEE, a system that automates cyber threat hunting by generating queries based on abstract threat descriptions. This approach emphasizes the need for automation in threat hunting, using AI to scale detection across large networks [120].

Together, these studies underscore the transformative role LLMs play in cyber threat intelligence, from real-time threat detection at the edge to the automation of CTI extraction and analysis from unstructured text. Each approach addresses distinct challenges, whether it be enhancing the accuracy of threat detection or improving the efficiency of knowledge graph construction, providing a rich foundation for future cybersecurity research. Table 6 presents recent studies highlight the role of LLMs in cyber threat intelligence (CTI). Hasan et al. employ lightweight models for decentralized threat analysis at edge devices, boosting privacy. Clairoux-Trepanier et al. achieve a high accuracy of 98% using GPT-3.5-turbo for actionable intelligence extraction. Wu et al.'s Knowledge Graph Verifier improves semantic understanding through LLM integration. Jo et al.'s Vulcan excels in named-entity recognition, while Liu and Zhan efficiently automate CTI knowledge graph construction. Hu et al. leverage GPT for data annotation, and Karuna et al. scale threat detection with AI-generated queries.

Table 6. Comparison of studies on cyber threat intelligence using LLMs.

Ref.	Model(s) Used	Dataset/Data Type	Main Contribution	Best Performance	Limitation
[115]	LLMs with lightweight ML models	Real-time local data streams from edge devices	Distributed threat intelligence on edge devices to enhance security	N/A	Requires coordination among edge devices
[116]	GPT-3.5-turbo	Cybercrime forum conversations (XSS, Exploit_in, RAMP)	Assessing the accuracy of LLMs for CTI extraction	98% accuracy	Needs improvement in distinguishing between factual events and stories
[117]	Knowledge Graph Verifier with LLMs	OSCTI claims and documents	CTI quality assessment using knowledge graphs with paragraphs as nodes	N/A	Limited to the created dataset for threat intelligence reliability verification
[118]	Neural language model-based NER and RE	Unstructured text from CTI reports	Extracting static CTI data and analyzing semantic relationships	0.972 (NER) and 0.985 (RE) F-scores	Limited to specific types of static CTI data
[119]	ChatGPT for knowledge graph construction	13 CTI reports	Efficient extraction of attack-related entities and relationships	Outperforms AttacKG and REBEL	Low scalability for larger datasets
[121]	LLMs	Unstructured threat hunting data	Applying LLMs for proactive cyber threat hunting	N/A	Challenges related to bias, fairness, and computational efficiency
[120]	WILEE system with AI and DSL	High-level threat descriptions and implementations	Automating query generation for cyber threat hunting	N/A	Lacks validation against diverse threat landscapes
[101]	GPT-based LLM for knowledge graph construction	Unstructured OSCTI reports	Constructing knowledge graphs from unstructured CTI	87.88% precision (NER), 96.53% precision (TTP classification)	Requires large amounts of labeled data for model fine-tuning

3.7. Incident Response and Management

Several studies showcase the transformative impact of AI and LLMs in enhancing incident management across various sectors. Jiang et al. focus on large-scale cloud systems, introducing Xpert, an AI-powered framework that automates KQL query recommendations to enhance incident resolution at Microsoft [122]. Grigorev et al. (2024) further explore this concept by integrating LLMs with ML to classify the severity of traffic incidents based on accident reports, showing superior results in multiple datasets across regions [123].

Chen et al. build on Artificial Intelligence for IT Operations (AIOps) for cloud services, identifying challenges in diagnosing unprecedented incidents and proposing IcM BRAIN, an AI framework to improve incident response efficiency at Microsoft [124]. Similarly, Tharayil et al. leverage LLMs to automate the ticket dispatching process in customer service, reducing error rates and improving efficiency [125].

Grigorev et al. also introduce IncidentResponseGPT, a system that generates region-specific traffic response plans using generative AI, expediting decision-making and resource allocation [126]. In civil aviation, Tulechki applies NLP to analyze and classify incident reports, improving information access for aviation professionals [127]. Lastly, Sufi presents a GPT-based framework for extracting cyber threat features from textual descriptions, aiding non-technical strategists in analyzing cyber incidents with high accuracy [128]. This comparison of studies utilizing LLMs in incident management presented by Table 7 emphasizes the models employed, datasets used, main contributions, and performance metrics. Notable findings include high accuracy achieved by models such as GPT-3.5-turbo, with performance metrics reaching around 98% in threat intelligence extraction. Other models, like IncidentResponseGPT and Vulcan, excelled in automating incident response and recognizing entities, respectively. However, limitations include challenges in generalizing across domains and handling unprecedented incidents, highlighting the complexity and resource demands associated with implementing LLMs in incident management.

Table 7. Comparison of studies on AI LLMs in incident management.

Ref.	Model(s) Used	Dataset/Data Type	Main Contribution	Best Performance	Limitation(s)
[122]	LLMs	Incident logs from Microsoft	Xpert: AI-powered framework for automating KQL query recommendations for incident resolution	Not explicitly mentioned	Limited to large-scale cloud systems
[123]	Hybrid of LLMs and ML	Traffic accident reports from multiple regions	Severity classification of traffic incidents	95.2% accuracy in severity classification	May not generalize well to different traffic environments
[124]	AIOps-based models	Cloud service incident data	IcM BRAIN: AI framework to improve incident response efficiency in cloud services	Not explicitly mentioned	Challenges in handling unprecedented incidents
[125]	NLP, Generative Models	Customer service incident tickets	Automating ticket dispatching process, reducing error rates	87% accuracy in incident categorization	Performance limited to structured incident reports
[126]	Generative AI (IncidentResponseGPT)	Traffic incident data across regions	Region-specific traffic response plans	92% efficiency in resource allocation	Limited to traffic management domain
[127]	NLP	Civil aviation incident reports	Classification and risk analysis of civil aviation incidents	Not explicitly mentioned	Limited to civil aviation domain
[128]	GPT-based Framework	Historical cyber incident reports	Extracting cyber threat features for non-technical users	High accuracy (not explicitly mentioned) in feature extraction	Limited to textual cyber threat reports

3.8. Data Protection and Privacy

The emergence of LLMs has brought forth significant privacy and ethical concerns, prompting extensive research in this area. Wu et al. [129] examine ChatGPT, highlighting its potential across various sectors, while emphasizing the importance of addressing security and ethical implications in its integration. Kibriya et al. [130] further investigate privacy risks, categorizing them into training and inference stages, and stress the need for stakeholder collaboration to implement effective privacy-preserving mechanisms.

In a related study, Plant et al. [131] focus on data leakage risks associated with LLMs, revealing that larger models are more susceptible to adversarial attacks. They advocate for privacy-preserving algorithms despite their impact on model performance. Complementarily, Brown et al. [132] argue that existing data protection techniques inadequately address the complexities of natural language and suggest that language models should be trained on explicitly public data. LLM Privacy Policy explores the application of LLMs in automating privacy compliance analysis, demonstrating high efficiency in evaluating privacy policy disclosures. Lastly, Peris et al. [133] discuss the importance of integrating privacy measurement and preservation techniques throughout the LLM lifecycle, offering strategies for mitigating privacy risks.

Table 8 provides an overview of studies focused on data protection and privacy in the context of LLMs. It compares various aspects, including the models used, datasets or data types analyzed, primary contributions, best performance, and identified limitations. The table highlights diverse contributions, such as exploring security and ethical implications, addressing privacy concerns during training and inference, and evaluating data leakage risks. Some studies propose privacy-preserving techniques and compliance analysis with LLMs, achieving strong performance metrics. However, limitations include challenges in generalization, reliance on specific datasets, and balancing model utility with privacy preservation efforts.

Table 8. Comparison of studies on data protection and privacy.

Ref.	Model(s) Used	Dataset/Data Type	Main Contribution	Best Performance	Limitation(s)
[117]	ChatGPT	N/A	Explores security, privacy, and ethical implications of ChatGPT.	N/A	Highlights challenges in adopting LLMs.
[130]	N/A	N/A	Investigates privacy concerns in LLMs during training and inference.	N/A	Requires stakeholder collaboration for effective privacy solutions.
[131]	Various LLMs	Multi-lingual dataset on sentiment analysis	Evaluates data leakage risks and advocates for privacy-preserving algorithms.	F1 score correlated with complexity	Impact on model utility with privacy-preserving methods.
[134]	ChatGPT, Llama 2	Privacy policy annotations	Demonstrates effectiveness of LLMs in automated privacy compliance analysis.	F1 score $> 93\%$	Limited to specific corpora for evaluation.
[132]	N/A	N/A	Discusses the inadequacy of current data protection techniques for language models.	N/A	Proposes training on publicly produced text, which may limit data availability.
[133]	Pretrained LLMs	N/A	Introduces privacy measurement and preservation techniques during LLM lifecycle.	N/A	Focus on industrial applications may not generalize to all contexts.

3.9. Cloud Security

Cloud security refers to the set of policies, technologies, and practices designed to protect data, applications, and infrastructure in cloud computing environments. It addresses threats such as data breaches, unauthorized access, and service disruptions by employing measures like encryption, access control, and vulnerability management [135]. Cloud security also ensures compliance with privacy standards and regulatory requirements. As organizations increasingly rely on cloud services, robust cloud security is critical for safeguarding sensitive information, maintaining business continuity, and ensuring the integrity of shared resources in multi-cloud environments [136].

Algorithm 3 represents the process for implementing an LLM-based Cloud Security Framework that leverages LLMs to enhance cloud security operations through automation and insights. It outlines a step-by-step methodology starting from data collection and preprocessing, followed by feature extraction using pretrained LLMs, and designing task-specific models for functions like vulnerability scoring and root cause analysis. The framework emphasizes model fine-tuning, transfer learning, automation integration, and performance evaluation, ensuring robustness and explainability. It also addresses challenges such as data heterogeneity while refining the framework based on user feedback. This approach aims to optimize cloud security processes by incorporating intelligent, automated systems powered by LLMs, enhancing both efficiency and accuracy in detecting and mitigating security threats.

Figure 7 summarizes the steps involved in the LLM-based Cloud Security Framework, which leverages LLMs to enhance cloud security through automation, insights, and integration into existing systems.

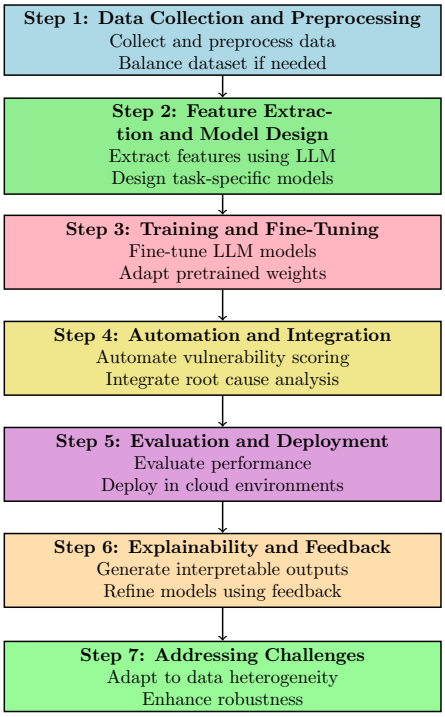

Figure 7. Steps of the LLM-based Cloud Security Framework.

The studies presented in this section illustrate the integration of AI and LLMs to enhance security and operational efficiency in cloud computing environments. Bulut et al. [137] introduce NL2Vul, a framework that automates the assessment of vulnerabilities in Cloud Security Posture Management (CSPM) tools. By employing deep neural networks and transfer learning, NL2Vul minimizes human intervention while providing

vulnerability scores based on data from the National Vulnerability Database (NVD) and GitHub Issues. This method addresses the overwhelming increase in vulnerabilities and aims to improve the accuracy of risk assessments in multi-cloud environments. In a related effort, Chen et al. [138] present RCACopilot, an innovative on-call system that automates root cause analysis (RCA) for cloud incidents. By leveraging LLMs, RCACopilot matches incidents to handlers, aggregates critical information, and predicts root causes, significantly improving the RCA accuracy to 0.766, thus streamlining support operations.

Algorithm 3: LLM-based Cloud Security Framework

Input: Dataset $D = \{d_1, d_2, \ldots, d_n\}$, Pretrained LLM $\mathcal{M}_{\text{LLM}}$
Output: Enhanced cloud security through automation and insights

Step 1: Data Collection and Preprocessing
Collect data from sources like vulnerability databases, incident logs, and support requests.
Preprocess the dataset: $D_{\text{processed}} = \mathcal{P}(D)$.
Balance the dataset if needed: $D_{\text{balanced}} = \mathcal{A}(D_{\text{processed}})$.

Step 2: Feature Extraction and Model Design
Extract features: $F = \mathcal{M}_{\text{LLM}}(D_{\text{processed}})$.
Design task-specific models:

- Vulnerability scoring: $\mathcal{V} : F \rightarrow \text{Scores}$
- Root cause analysis: $\mathcal{R} : F \rightarrow \text{Causes}$

Step 3: Training and Fine-Tuning
Fine-tune the model: $\theta_{\text{fine-tuned}} = \arg\min_\theta \mathcal{L}(F, Y)$.
Use transfer learning to adapt pre-trained weights: $\theta_{\text{new}} = \theta_{\text{pretrained}} + \Delta\theta$.

Step 4: Automation and Integration
Develop automation frameworks:

- NL2Vul for vulnerability scoring: $\text{Scores} = \mathcal{V}(\mathcal{M}_{\text{LLM}}(D_{\text{balanced}}))$
- RCACopilot for root cause analysis: $\text{Causes} = \mathcal{R}(\mathcal{M}_{\text{LLM}}(D_{\text{balanced}}))$

Integrate into cloud security tools: $\mathcal{T}_{\text{cloud}} = \alpha\mathcal{V} + \beta\mathcal{R}$.

Step 5: Evaluation and Deployment
Evaluate performance metrics:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad \text{RCA Accuracy} = 0.766$$

Deploy in cloud environments: $\mathcal{D}_{\text{deployed}} = \mathcal{M}_{\text{LLM}} \oplus \mathcal{T}_{\text{cloud}}$.

Step 6: Explainability and Feedback
Generate interpretable outputs: $I = \mathcal{E}(\mathcal{M}_{\text{LLM}}, F)$.
Refine models with user feedback: $\mathcal{M}_{\text{updated}} = \mathcal{M}_{\text{LLM}} + \mathcal{T}(F_{\text{feedback}})$.

Step 7: Addressing Challenges
Adapt to data heterogeneity: $\hat{D} = \mathcal{D}_{\text{Adapt}}(D_{\text{source}}, D_{\text{target}})$.

return Enhanced security insights and automated cloud security framework.

Linking these findings to the broader context, Kilhoffer et al. [139] explore the assessment of privacy standards using a fine-tuned LLM approach. Their research emphasizes the necessity of establishing robust privacy controls within cloud frameworks, which are still developing compared to their security counterparts. They analyze 1511 controls across nine certifiable standards, uncovering a focus on security and risk rather than data rights.

Similarly, Mouratidis et al. [140] propose a novel security modeling language tailored for cloud environments, enabling the integration of security requirements with cloud computing concepts. Their approach is complemented by automated analysis techniques that enhance models with security knowledge, showcasing a foundational advancement in understanding cloud security requirements.

Furthermore, studies by Baghdasaryan et al. [141] and Cao et al. [142] expand on enhancing customer support and vulnerability detection through intelligent systems. Baghdasaryan et al. develop a recommender system that utilizes LLMs to match support requests with previous resolutions, aiming to reduce mean resolution times. This system highlights the practical application of AI to foster proactive support solutions. In parallel, Cao et al. introduce LLM-CloudSec, an unsupervised approach for fine-grained vulnerability analysis, which employs a retrieval-augmented generation method for classifying vulnerabilities at a detailed level, thus advancing the field of automated security analysis in cloud applications.

Lastly, Stutz et al. [143] examine the broader implications of integrating AI into cloud computing and cybersecurity, addressing challenges such as data access and cyber threats. Their chapter highlights the potential of AI in anomaly detection and mitigation strategies within smart environments, suggesting a future direction for AI-enhanced security measures across various cloud applications. Collectively, these studies underscore the transformative role of AI and LLMs in bolstering security and operational efficiency in cloud computing, paving the way for more resilient and proactive cloud infrastructures. Table 9 illustrates the integration of LLMs to enhance security and operational efficiency in cloud computing. Notable contributions include NL2Vul, which automates vulnerability assessment, and RCACopilot, achieving a root cause analysis accuracy of 0.766. Additionally, research emphasizes the need for robust privacy controls and presents a security modeling language tailored for cloud environments. Other advancements include LLM-CloudSec for detailed vulnerability analysis and recommender systems for customer support, highlighting the transformative potential of AI and LLMs in creating resilient cloud infrastructures despite ongoing challenges in implementation.

Table 9. Comparison of studies on cloud security using LLMs.

Ref.	Model(s) Used	Dataset/Data Type	Main Contribution	Best Performance	Limitation(s)
[137]	Deep Neural Networks	National Vulnerability Database (NVD), GitHub Issues	Proposed NL2Vul framework for automating vulnerability scoring	N/A	Relies on human-driven efforts for initial data preparation
[138]	Large Language Model (LLM)	Real-world incident dataset from Microsoft	Introduced RCACopilot for automating root cause analysis (RCA)	RCA accuracy of 0.766	Limited to incidents matching predefined types
[139]	Fine-tuned BERT	Privacy standards dataset (1511 controls)	Analyzed privacy controls across standards using LLM	N/A	Focuses on security aspects, lacking attention to data rights
[141]	LLMs	Support request and knowledge base data	Developed a recommender system for customer support	N/A	Inaccurate accuracy estimates due to insufficient datasets
[142]	Retrieval Augmented Generation (RAG)	Juliet C++ test suite, D2A dataset	Proposed LLM-CloudSec for fine-grained vulnerability analysis	CWE-based classification and line-level analysis	Limited by the complexity of heterogeneous cloud environments
[140]	Security modeling language	Cloud computing system models	Introduced a novel language and techniques for security modeling	N/A	Requires stakeholder input for effective modeling
[143]	AI-driven anomaly detection	Various smart environment datasets	Explored AI's role in improving cybersecurity in cloud environments	N/A	Challenges in data access and integration for security purposes

3.10. Security Compliance and Auditing

The reviewed studies explore various applications of LLMs, focusing on cybersecurity, auditing, legal compliance, and privacy.

McIntosh et al. [27] evaluated cybersecurity frameworks like COBIT and ISO 42001 for integrating LLMs, finding gaps in risk oversight but recognizing ISO 42001's strength in managing AI systems. In insider threat detection, Song et al. introduced Audit-LLM, a multi-agent framework that improves detection accuracy using LLMs [144], while Xia et al. developed AuditGPT for smart contract auditing, showing superior performance in identifying violations [145].

Fotoh and Mugwira examined LLMs' ethical implications in external auditing, highlighting both efficiency gains and concerns around accuracy and independence [146]. Cartwright et al. focused on privacy compliance of LLMs like ChatGPT, revealing varied adherence to privacy standards [147], while Hassani demonstrated LLMs' potential to automate legal compliance checks in the food industry [148].

Zhu et al. proposed a framework combining LLMs and DL for automating regulatory compliance in construction projects [149]. Lastly, Chard et al. audited privacy practices in AI, identifying vulnerabilities in ChatGPT's handling of sensitive data [150]. Overall, these studies emphasize the growing role of LLMs in enhancing automation and efficiency, while also underscoring the need for improved oversight and compliance measures.

Table 10 highlights the transformative role of LLMs in security compliance and auditing across diverse domains, including cybersecurity, privacy, legal regulations, and blockchain systems. Studies demonstrate the adaptability of LLMs in tasks such as insider threat detection, automated compliance checking, and privacy assessment. Notable examples include AuditGPT's precise identification of Ethereum smart contract violations and multi-agent frameworks outperforming traditional methods in detecting insider threats. Moreover, models like Claude Sonet have shown strong adherence to regulatory standards, emphasizing their potential for enhancing compliance efficiency. Despite these achievements, challenges such as domain-specific limitations, faithfulness issues, and ethical concerns—like auditor independence and privacy risks—remain significant obstacles.

The findings underscore the importance of developing robust frameworks to address these challenges while harnessing the benefits of LLMs. Studies emphasize the need for improved oversight mechanisms to mitigate risks like hallucination and data leakage. They also highlight the resource-intensive nature of fine-tuning LLMs for specialized tasks, limiting scalability. Future directions should focus on enhancing model transparency, expanding applicability to broader regulatory contexts, and ensuring ethical and secure use in sensitive environments. These advancements will be crucial in maximizing the potential of LLMs as reliable tools for automating and optimizing compliance and auditing processes.

Table 10. Comparison of studies on security compliance and auditing using LLMs.

Ref.	Model(s) Used	Dataset/Data Type	Main Contribution	Best Performance	Limitation(s)
[27]	N/A (Cybersecurity Frameworks Analysis)	Cybersecurity Governance, Risk, and Compliance (GRC) Frameworks	Comparative gap analysis of GRC frameworks for LLM adoption	N/A	Inadequacies in LLM risk oversight
[144]	Audit-LLM (multi-agent framework with COT reasoning)	CERT r4.2, CERT r5.2, PicoDomain (Insider Threat Detection)	Multi-agent collaboration for enhanced insider threat detection	Superior to existing baselines	Faithfulness hallucination in LLM conclusions
[146]	ChatGPT	Prompts simulating audit scenarios	Ethical implications of LLMs in external auditing	N/A	Inaccurate responses, concerns over auditor independence

Table 10. *Cont.*

Ref.	Model(s) Used	Dataset/Data Type	Main Contribution	Best Performance	Limitation(s)
[145]	AuditGPT	222 ERC Rules (Ethereum Smart Contracts)	ERC rule violation detection in smart contracts using LLMs	Detected 418 violations with only 18 false positives	Limited scope to ERC rule types
[147]	ChatGPT-4o, Claude Sonet, Gemini Flash	Hypothetical case studies for privacy compliance	Privacy compliance assessment of LLMs under the EU AI Act	Robust compliance by Claude Sonet	Inconsistencies in Gemini Flash's anonymization
[148]	BERT, GPT models	Legal provisions in food safety domain	Automated legal compliance checks in food safety regulations	Significant accuracy improvement in legal provision classification	Requires fine-tuning for specific legal tasks
[149]	LLM, DL	Regulatory texts in BIM domain	Automated compliance checking framework for construction projects	Improved accuracy over traditional methods	Limited application outside the AEC field
[150]	ChatGPT	Audited privacy practices in ChatGPT	Developed audit framework for privacy vulnerabilities in AI systems	Identified key privacy issues	Data leakage in sensitive contexts

3.11. Endpoint Security

Several recent studies explore the impact of LLMs and advanced AI techniques on cybersecurity. For instance, Sharif et al. propose DrSec, a system designed to enhance endpoint detection and response (EDR) by employing self-supervised learning to pre-train foundation language models (LMs). These models can be adapted to various downstream tasks, reducing false positives and improving security incident detection. The study highlights the ability of DrSec to reduce alert fatigue and better identify processes within large datasets, showcasing its superiority over current security methods in alert triage and process identification [151].

In line with the examination of LLM applications in cybersecurity, Motlagh et al. focus on the broader context of LLM use within this domain. They provide a detailed survey of LLM applications, both defensive and offensive, within cybersecurity. The study categorizes the current landscape and identifies key research gaps, offering a holistic understanding of the risks and opportunities LLMs present in this critical field [152], Black et al.also explore LLMs' role in code generation, where they identify vulnerabilities in LLM-produced code and propose techniques to reduce these weaknesses. They demonstrate that providing specific prompts can enhance security, emphasizing the importance of careful prompt design when using LLMs for tasks like code generation [153].

Further building on LLM evaluations, Shao et al conduct an empirical study focusing on how LLMs perform in solving Capture the Flag (CTF) challenges. They develop workflows to automate the solving process and compare LLM performance to human participants, revealing that LLMs often outperform humans in certain cybersecurity tasks. This study contributes to understanding the potential of LLMs in practical offensive security challenges [154].

In a similar vein, Ren et al. introduce CodeAttack, a novel framework that examines how LLMs handle safety in code generation tasks. The study shows that LLMs like GPT-4, Claude-2, and Llama-2 are vulnerable to safety issues when generating code, with more than 80% of their safety guardrails bypassed in specific conditions. The authors highlight the need for more robust algorithms to address these safety challenges and mitigate potential risks in AI-driven code generation [155].

Addressing broader risk factors, Cui et al. present a taxonomy that systematically categorizes the risks associated with LLM systems, from input modules to language models and toolchains. Their paper emphasizes the importance of establishing risk assessment benchmarks and provides insights into mitigation strategies across different aspects of

LLM implementation [156]. Finally, Szabó and Bilicki propose a new approach to web application security by leveraging GPT models for static source code analysis. Their study focuses on detecting vulnerabilities like CWE-653, achieving an impressive detection rate of 88.76% and showing that LLMs can significantly enhance vulnerability [157] detection in static code. These studies collectively highlight the growing role of LLMs in both defensive and offensive cybersecurity tasks while also addressing the risks and vulnerabilities these systems might introduce. Together, they point to a future where AI-driven models are integral to cybersecurity but where careful management and mitigation strategies are crucial for ensuring their safe deployment.

Table 11 illustrates the impactful contributions of LLMs to endpoint security, highlighting both their effectiveness and limitations in addressing complex cybersecurity challenges. For example, DrSec, presented by [151], enhances endpoint detection and response (EDR) by leveraging self-supervised learning to improve process identification and alert triage. While achieving a 75.11% PR AUC in alert triage, the study acknowledges limitations like false positives and constrained supervised learning data. Similarly, ref. [157] employs GPT models for static code analysis, achieving a high detection rate of 88.76% for CWE-653 vulnerabilities in source code. These findings emphasize the precision and adaptability of LLMs for specific cybersecurity tasks, although scalability and domain generalization remain challenges.

Table 11. Comparison of studies on endpoint security using LLMs.

Ref.	Model(s) Used	Dataset/Data Type	Main Contribution	Best Performance	Limitation(s)
[151]	Self-supervised learning, LMs	Event-sequence data (91 M processes, 2.55 B events)	DrSec system for EDR, improved process identification and alert triage	75.11% PR AUC in alert triage	False positives, limited supervised learning data
[152]	Various LLMs	Literature review, no specific dataset	Comprehensive review of LLM applications in cybersecurity (defensive/offensive)	N/A	Research gaps in LLM applications, theoretical focus
[153]	Commercial LLMs	Generated code via prompt types	Analysis of code generation safety, reducing vulnerabilities via prompt design	Vulnerability reduction through prompt design	No guarantee of fully safe code, limited model interactions
[154]	LLMs for CTF challenges	Real-world CTF challenges dataset	Evaluation of LLMs in solving CTF challenges (HITL and fully automated workflows)	LLMs outperform humans in solving challenges	Limited to selected CTF tasks, no generalization
[155]	GPT-4, Claude-2, Llama-2	Code inputs for CodeAttack framework	CodeAttack framework to test LLM safety generalization for code inputs	80%+ success in bypassing safety guardrails	Safety risks not fully mitigated
[156]	N/A (Taxonomy paper)	Various LLM systems	Comprehensive risk taxonomy and mitigation strategies for LLM systems	N/A	No empirical performance evaluation
[157]	GPT models for static code analysis	Static source code (front-end applications)	Detection of CWE-653 vulnerability using GPT prompts for source code inspection	88.76% detection rate	Limited to specific vulnerability type (CWE-653)

The studies also reveal notable gaps and vulnerabilities in LLM implementations. For instance, the CodeAttack framework by [155] demonstrates that more than 80% of safety guardrails in GPT-4 and similar models can be bypassed, underscoring critical safety risks in code generation and interaction. Moreover, ref. [154] evaluates LLMs in Capture the Flag (CTF) cybersecurity challenges, showing that LLMs outperform human participants in solving these tasks. However, their applicability is limited to selected

CTF problems, lacking generalization to broader contexts. Overall, while these studies showcase the potential of LLMs to revolutionize endpoint security through improved detection and response mechanisms, they also stress the need for enhanced safety protocols, comprehensive testing frameworks, and further research to address limitations in scalability and vulnerability mitigation.

4. Security Policy and Compliance

The implementation of LLMs in cybersecurity policy and compliance demonstrates transformative potential for analyzing and managing complex, evolving threats. These models excel at processing unstructured data sources, such as textual descriptions, code snippets, and technical documentation, to extract meaningful insights that enhance malware classification, behavior analysis, and policy enforcement [158,159]. By leveraging advanced natural language understanding, LLMs can identify patterns and attributes in malware reports, threat intelligence feeds, and technical documents, aiding in the identification of malware families and attack methodologies [158].

4.1. Advanced Malware Analysis

LLMs surpass traditional signature-based detection methods by dynamically learning patterns from diverse and complex datasets. For example, through deep feature extraction, LLMs categorize unknown malware samples by identifying subtle, behavior-based indicators that traditional systems might miss [159]. This ability ensures proactive detection of zero-day vulnerabilities and emerging threats, enhancing the accuracy and efficiency of malware detection pipelines [158,159]. Moreover, LLMs aid in correlating observed malware actions with established attack patterns, such as the MITRE ATT&CK framework, providing actionable insights for incident response teams [160].

4.2. Policy Analysis and Automation

In the domain of security policy and compliance, LLMs streamline the evaluation and implementation of regulatory requirements. By parsing and interpreting regulatory texts, such as GDPR, HIPAA, or industry-specific security standards, LLMs automate compliance assessments, ensuring organizations meet legal and operational mandates efficiently [161,162]. The integration of LLMs into governance tools facilitates continuous monitoring of policy adherence, automatically flagging discrepancies or gaps in security frameworks [163]. For instance, automated LLM-driven audits can assess system configurations against pre-defined compliance standards, reducing manual overhead and minimizing errors [164].

4.3. Context-Aware Decision Making

LLMs also enable contextual analysis of security incidents, offering nuanced interpretations of malware behaviors within broader security policies. By integrating contextual metadata, such as organizational policies or historical incident data, LLMs prioritize vulnerabilities based on business impact, helping organizations allocate resources effectively [165]. This capability is particularly valuable for crafting dynamic, adaptive security policies that evolve alongside emerging threats [166].

4.4. Enhancing Threat Intelligence

Through integration with threat intelligence platforms, LLMs extract and contextualize insights from diverse sources, including cybersecurity feeds, dark web activity, and public repositories [167]. By linking these insights to compliance mandates, LLMs provide a holistic view of an organization's security posture, enabling preemptive policy

adjustments [168]. For example, when new malware behaviors are detected, LLMs can recommend specific policy updates to mitigate identified risks [169].

4.5. Interpretable Insights for Stakeholders

One of the key advantages of LLMs is their ability to generate interpretable insights tailored for diverse stakeholders. From security analysts to C-suite executives, LLMs can generate detailed reports or high-level summaries, bridging the gap between technical complexities and strategic decision-making [170]. For instance, they can produce compliance dashboards that track adherence to multiple regulatory frameworks, flagging areas of non-compliance and suggesting remediation actions [171].

The table represents a taxonomy of studies that leverage LLMs in various cybersecurity applications. It categorizes these studies based on key aspects such as the application domain, cybersecurity task, datasets used, evaluation metrics, advantages and limitations, and the techniques or methodologies employed. This structured format provides an overview of how LLMs are applied to enhance tasks like incident response, threat intelligence, vulnerability detection, anomaly detection, phishing detection, and security policy analysis. For example, studies like CyBERT [172] focus on text classification for incident response, while BioBERT [173] and SciBERT [174] excel in threat intelligence tasks such as named-entity recognition (NER) and semantic understanding.

The taxonomy highlights the diverse use cases of LLMs, from improving scalability and resource efficiency to enhancing explainability and accuracy in cybersecurity tasks. Each entry identifies specific datasets (e.g., ICS, Edge-IIoTset, Android) and evaluation metrics (e.g., F1 score [173], accuracy [20], task-specific metrics [174]) to benchmark performance. Moreover, it outlines the advantages and limitations of each approach, such as scalability [59], resource constraints [175], or explainability issues [176], providing insights into the trade-offs involved in deploying these models. Additionally, it showcases the methodologies used, such as fine-tuning [20,173], pre-training [172,174], parameter reduction [175], or integrating privacy-preserving techniques [59], offering a comprehensive perspective on the technological advancements driving LLM adoption in cybersecurity. This table serves as a valuable reference for researchers and practitioners exploring the integration of LLMs into cybersecurity frameworks.

Table 12 presents a comprehensive taxonomy of studies focusing on the use of LLMs in enhancing various aspects of cybersecurity. The table categorizes these studies based on application domains, specific cybersecurity tasks, datasets used, evaluation metrics, advantages and limitations, and the techniques and methodologies employed. This structured overview provides a valuable summary of how LLMs are being applied across diverse domains, including incident response, threat intelligence, anomaly detection, vulnerability detection, and phishing detection. It highlights both the versatility of LLMs and the unique approaches adopted in leveraging them for specific cybersecurity challenges.

One of the key insights from the table is the diversity of applications and tasks where LLMs have demonstrated significant utility. For instance, models like CyBERT and SecureBERT focus on incident response, using pre-training and fine-tuning techniques to achieve high accuracy in tasks like text classification and sentiment analysis. Similarly, BioBERT and SciBERT are adapted for threat intelligence tasks, such as named-entity recognition and semantic understanding, emphasizing resource efficiency and scalability. Notably, models like SecurityBERT and HuntGPT are tailored for IoT security and malware detection, showcasing the adaptability of LLMs to domain-specific challenges like privacy-preserving methods and explainable AI.

The taxonomy also sheds light on the trade-offs between advantages and limitations in using LLMs for cybersecurity. While models such as FalconLLM and VulRepair excel

in improving accuracy and performance for text classification and vulnerability detection, others like MalBERT and ChatSpam underscore the importance of scalability and detailed reasoning. However, challenges such as the need for fine-tuning, interpretability, and addressing resource constraints persist across many applications. The inclusion of innovative approaches, such as explainable AI in HuntGPT and benchmarking frameworks in CyberBench, highlights the ongoing efforts to enhance the transparency, efficiency, and reliability of LLM-based solutions in cybersecurity. This taxonomy not only underscores the growing role of LLMs in this critical domain but also points to areas requiring further research and development to maximize their potential.

Table 12. A taxonomy for studies about using LLMs in enhancing cybersecurity.

Study	Application Domain	Cybersecurity Task	Datasets	Evaluation Metrics	Advantages and Limitations	Techniques and Methodologies
CyBERT [172]	Incident Response	Text Classification	ICS	Accuracy	Scalability	Pre-training
BioBERT [173]	Threat Intelligence	Named-Entity Recognition	Biomedical	F1 Score	Resource Efficiency	Fine-tuning
SciBERT [174]	Threat Intelligence	Semantic Understanding	Scientific	Task-specific	Scalability	Pre-training
Par. Red [175]	Anomaly Detection	Anomaly Detection	-	Task-specific	Resource Efficiency	Parameter Reduction
SecureBERT [20]	Incident Response	Sentiment Analysis	Cybersecurity	Accuracy	Scalability	Fine-tuning
SecurityBERT [59]	IoT Security	Anomaly Detection	Edge-IIoTset	Accuracy	Resource Efficiency	Privacy-preserving
LLMSecEval [176]	Security Policy Analysis	Evaluation	LLM	Security	Explainability	Security Evaluation
Misuse [177]	Phishing Detection	Text Generation	-	Efficiency	-	Misuse Detection
Inter. [178]	Threat Intelligence	Named-Entity Recognition	ATT&CK	Interpretability	Explainability	Interpretability Methods
MalBERT [58]	Malware Detection	Text Classification	Android	Accuracy	Scalability	Static Analysis
WebBERT [179]	Vulnerability Detection	Text Classification	HTTP requests	Success Rate	Accuracy	NLP Techniques
CyberBench [180]	Threat Intelligence	Various	Cyber	Various	Benchmarking	Benchmarking
VulRepair [181]	Vulnerability Detection	Text Generation	Vulnerability fixes	Performance	Accuracy Improvement	NMT Techniques
FalconLLM [182]	Vulnerability Detection	Text Classification	FormAI	Accuracy	Performance	Fine-tuning
ChatSpam [183]	Phishing Detection	Text Classification	Email	Accuracy	Detailed Reasoning	LLMs
HTML [184]	Security Policy Analysis	Various	MiniWoB	Accuracy	Performance	Fine-tuning
Vulnerability-Det [57]	Vulnerability Detection	Software Vulnerability Detection	LLM	Technical Accuracy	Explainable AI	Fine-tuning
HuntGPT [185]	Anomaly Detection	Malware Detection	KDD99	Technical Accuracy	Explainable AI	Random Forest

5. LLMs' Vulnerabilities to Cyberattacks

Despite their impressive capabilities, the application of LLMs in cybersecurity is not without limitations. The accuracy and reliability of these models depend heavily on the quality and scope of their training data, which may not always encompass the nuanced and rapidly evolving landscape of cyber threats. Furthermore, ethical and security considerations emerge, particularly in the use of LLMs for malware development or the exploration of vulnerabilities, underscoring the need for responsible usage and ongoing evaluation of these powerful tools.

5.1. Attack Mechanisms

The realm of LLMs has seen a proliferation of studies aimed at understanding and mitigating security vulnerabilities. Among these, the exploration of attack mechanisms stands out, especially concerning how malicious actors can compromise these models.

These attacks are primarily categorized into indirect prompt injections and vulnerabilities tied to visual and multimodal inputs.

5.1.1. Indirect Prompt Injection Attacks

A series of studies have shed light on the susceptibility of LLMs to indirect prompt injections, revealing the ease with which malicious prompts can be integrated without the awareness of the model or its users. The study titled "Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection" delves into how real-world applications integrating LLMs can be compromised, demonstrating the subtlety with which malicious instructions can be embedded into seemingly benign inputs [186]. Complementing this, ref. [187] systematically evaluates the effectiveness of extracting and leveraging prompts for malicious purposes, highlighting the inherent risk of treating prompts as non-sensitive information. Furthermore, ref. [188] introduces a methodology for embedding backdoors into LLMs through the iterative injection of triggers, illustrating a novel yet alarming vector for compromising these models.

5.1.2. Visual and Multimodal Adversaries

The vulnerability of LLMs extends beyond textual inputs to include visual and multimodal adversarial attacks. Ref. [189] explores how adversarially crafted visual examples can manipulate LLMs' outputs, suggesting that these models can be 'jailbroken' to produce unintended or malicious outputs when exposed to carefully crafted visual stimuli. Similarly, "(Ab)using Images and Sounds for Indirect Instruction Injection in Multi-Modal LLMs" demonstrates how images and sounds can serve as conduits for indirectly injecting malicious instructions into multimodal LLMs, underscoring the complexity of securing these models against a broad spectrum of input types [190]. Lastly, ref. [191] reveals how generative models, including LLMs, can be manipulated in real time using adversarial images, posing significant implications for the security of generative AI systems.

5.2. Safety and Robustness Evaluation

The safety and robustness of LLMs are crucial concerns as these models find broader application across various sectors. Research in this area has focused on identifying vulnerabilities within LLMs and devising methodologies to evaluate and enhance their safety and robustness. These investigations are pivotal in understanding the extent to which LLMs can be trusted and relied upon, particularly in tasks requiring high levels of safety and ethical considerations.

5.2.1. Model Vulnerability and Safety Failures

A series of studies have focused on the vulnerabilities of LLMs to safety training failures. Wei et al. [192] explore the susceptibility of LLMs to "jailbreak" attacks, which manipulate models to produce undesirable outputs. This study identifies two main failure modes in safety training: competing objectives and mismatched generalization. The research underscores the persistence of vulnerabilities in state-of-the-art models like GPT-4 and Claude v1.3, despite extensive safety efforts. It advocates for safety mechanisms that match the sophistication of the models themselves, challenging the notion that scaling up models is a sufficient solution to safety concerns. Qiu et al. [193] introduce a benchmark that evaluates both the safety and robustness of LLMs. Their innovative approach involves embedding malicious instructions within regular tasks to assess how well models maintain safety and instruction fidelity. This benchmark is critical for developing balanced LLMs that are sensitive to safety without compromising task performance. Shayegani et al. [194] provide a comprehensive survey of vulnerabilities in LLMs revealed by adversarial attacks. This survey spans various attack methodologies, including textual-only and multi-modal

attacks, and discusses potential defenses. By offering a structured overview of the field, this work aims to foster a deeper understanding of LLM vulnerabilities and encourage the development of more secure models.

5.2.2. Adversarial Alignment

Research into adversarial alignment explores the resilience of "ethically aligned" models against adversarial attacks. Typically, Carlini [195] investigates whether "ethically aligned" neural networks can withstand adversarial inputs designed to elicit harmful content. This study reveals the limitations of current NLP-based optimization attacks and highlights the vulnerability of multimodal models to adversarial image perturbations, suggesting that text-only models may also be susceptible to more sophisticated attacks. Morover, Zou et al. [196] introduce a method for generating universal and transferable adversarial attacks on aligned language models. Unlike previous approaches requiring significant human ingenuity, their method automates the creation of adversarial prompts that induce models to generate objectionable content. This work demonstrates the transferability of such attacks across various LLMs, raising crucial questions about preventing undesirable content generation. Lastly, Cao et al. [197] propose a Robustly Aligned LLM (RA-LLM) to defend against alignment-breaking attacks. Their approach introduces a robust alignment checking function to existing LLMs, effectively reducing the success rates of adversarial and handcrafted jailbreak prompts. This study not only highlights the feasibility of defending against such attacks but also emphasizes the importance of continuous efforts to align LLMs with human values.

Together, these studies paint a nuanced picture of the current state of LLM safety and robustness. They reveal not only the inherent challenges in safeguarding these models from adversarial threats but also the ongoing efforts to understand, evaluate, and improve their security. As LLMs continue to evolve and their applications grow, the insights gained from such research will be critical in guiding the development of more resilient, safe, and ethically aligned models.

Table 13 presents a comparative overview of studies evaluating the safety and robustness of LLMs. It categorizes the studies based on the ML models assessed, their applications, datasets used, advantages, and limitations. The table highlights a diverse range of approaches, from vulnerability analyses of advanced LLMs like GPT-4 and Claude v1.3 to defensive mechanisms implemented in models such as GPT-3.5 and Llama 2. It captures the evolving landscape of safety evaluation, emphasizing adversarial scenarios, model alignment, and self-assessment strategies to mitigate risks in real-world applications.

Key findings from the table include the identification of significant advancements in evaluating and addressing adversarial vulnerabilities in LLMs. Studies like [193] provide a comprehensive evaluation of text safety and output robustness using custom benchmark datasets, while [197] introduces robustly aligned models that defend against attacks without requiring retraining. Similarly, ref. [198] demonstrates the potential of automating red-team evaluations to uncover safety loopholes. However, the limitations of these studies, such as a lack of generalizability across diverse adversarial scenarios or dependency on initial conditions, underline the need for broader, multi-modal evaluations and practical defenses.

The table also underscores the gap between theoretical advances and their real-world applicability. For instance, while studies like [196,199] excel in demonstrating adversarial attack transferability and stealthy jailbreak prompts, their effectiveness is often constrained by computational complexity or the varying robustness of LLMs. These findings collectively emphasize the need for integrating collaborative, multi-agent frameworks, as proposed in [200], and developing adaptive strategies that balance theoretical rigor with practical deployment challenges.

Table 13. Comparison of studies on safety and robustness evaluation of LLMs.

Ref.	ML Model	Application	Dataset	Advantage	Limitation
[192]	GPT-4, Claude v1.3	Model Vulnerability	Red-team evaluation sets	In-depth analysis of safety training failures	Focuses more on theory than practical solutions
[193]	Various LLMs	Text Safety and Output Robustness	Custom benchmark dataset	Comprehensive evaluation of robustness and safety	May not cover all potential adversarial scenarios
[194]	Various LLMs	Adversarial Attack Survey	N/A	Extensive review, includes a range of attack methods	Primarily a survey, lacks new empirical data
[195]	Aligned neural networks	Adversarial Alignment	Multi-modal models	Highlights vulnerabilities in adversarial settings	Limited to text models, needs broader modal evaluations
[196]	Aligned Language Models	Adversarial Attacks	Vicuna-7B, 13B	Demonstrates transferability of attacks	Effectiveness varies significantly across models and settings
[197]	Robustly Aligned LLM (RA-LLM)	Defending Against Attacks	Open-source LLMs	Provides a defense mechanism without retraining	Theoretical analysis may not reflect all real-world conditions
[201]	GPT-3.5, Llama 2	Defensive Mechanisms	Various adversarial prompts	Enables LLMs to self-assess and defend without additional training	May not detect subtler forms of manipulation
[202]	GPT-2	Detecting Attacks	Adversarial prompts dataset	Utilizes perplexity to identify attacks	High false positive rates, mitigated with additional ML models
[203]	Various LLMs	Baseline Defenses	Not specified	Evaluates multiple simple defensive strategies	Does not introduce novel defensive mechanisms, focuses on evaluation
[200]	Various LLMs	Multi-agent Defense	Harmful and safe prompts	Employs a collaborative framework among multiple LLM agents	Effectiveness dependent on the division and coordination of tasks
[198]	GPT, LLaMa-2, Vicuna	Auto-generated Jailbreak Prompts	Not specified	Automates red-teaming of LLMs with high success rates	Initial seed quality affects performance
[199]	Aligned LLMs	Generating Stealthy Jailbreak Prompts	Not specified	Generates stealthy, semantically meaningful jailbreak prompts	Potential detection through advanced perplexity testing
[204]	Various LLMs	Interpretable Adversarial Attacks	Not specified	Produces readable, effective jailbreak prompts; transfers to black-box LLMs	Method complexity and computational demand might be high

5.3. Defense Strategies

The development of defensive mechanisms and strategies is crucial to safeguard against adversarial attacks. Recent studies have focused on innovative approaches for defending LLMs, exploring prompt-based defenses, model hardening techniques, and automated attack generation to ensure the models' safety and integrity. Specifically, Helbling et al. [201] propose a novel defense mechanism that enables LLMs to self-screen their generated responses for potential harmful content without requiring any fine-tuning or iterative output generation. By embedding the generated content into a pre-defined prompt and analyzing it with another instance of an LLM, this approach effectively reduces the success rate of various types of attacks to virtually zero on prominent models such as GPT 3.5 and Llama 2. Moving on, Alon et al. [202] introduce a method that employs perplexity evaluation to detect adversarial suffixes aimed at deceiving LLMs into generating dangerous responses. Despite the challenge of false positives in plain perplexity filtering, they demonstrate that a Light-GBM model trained on perplexity and token length can effectively

distinguish most adversarial attacks, thus providing a viable defense strategy. Moreover, Jain et al. [203] evaluate various defense strategies, including detection, input preprocessing, and adversarial training. They discuss the practicality and effectiveness of each defense in different settings and highlight the unique challenges and opportunities for securing LLMs compared to computer vision models. Lastly, Zeng et al. (2024) [200] propose a multi-agent defense framework that collaboratively filters harmful responses from LLMs by assigning different roles to LLM agents. This framework not only enhances instruction-following capabilities but also adapts to various sizes and types of LLMs, demonstrating its effectiveness in defending against jailbreak attacks while maintaining performance for normal user requests.

5.4. Adversarial Attack Generation

On the front of adversarial attack generation, Yu et al. [198] introduce, a black-box fuzzing framework that automates the generation of jailbreak templates. This framework significantly outperforms human-crafted templates in red-teaming LLMs, revealing the necessity for ongoing efforts to bolster LLM robustness. Liu et al. (2023) in [199] present a novel attack that generates stealthy jailbreak prompts capable of eluding perplexity-based defenses. By employing a hierarchical genetic algorithm, AutoDAN automates the generation of semantically meaningful prompts that showcase strong attack capabilities and transferability. Lastly, Zhu et al. in [204] develop a gradient-based adversarial attack that combines the strengths of readability and jailbreak success. AutoDAN's interpretable and diverse prompts not only bypass perplexity filters but also generalize to unforeseen harmful behaviors, underscoring the importance of versatile defense mechanisms.

5.5. Specialized Attack and Defense Themes

Recent studies have shed light on niche vulnerabilities and unique attack vectors against LLMs, highlighting the evolving landscape of cyber threats and the need for specialized defense mechanisms. These studies focus on multilingual, privacy, and encryption-based attacks, presenting novel challenges to the safety alignment of LLMs. Li et al. [205] delve into addressing concerns surrounding the privacy implications of LLMs like ChatGPT and the New Bing. Despite efforts to safeguard dialog safety, the expansive training datasets of LLMs, such as GPT-3's 45TB of text, raise questions about the inclusion of private information and subsequent privacy threats. Through extensive experimentation, the study unveils that integrated applications of LLMs could pose new privacy threats, urging a reevaluation of privacy measures in the age of generative AI. Yuan et al. [206] present an intriguing analysis, exploring the potential for ciphers to bypass the safety mechanisms of LLMs designed primarily for natural language processing. Their novel framework, CipherChat, evaluates the generalizability of safety alignments to ciphers, revealing that certain ciphers can effectively circumvent GPT-4's safety protocols in multiple domains. The discovery of a "secret cipher" capability within LLMs, particularly effective in role play and demonstrated in natural language, underscores the necessity for extending safety alignments to include non-natural languages. Deng et al. [207] investigate the multilingual jailbreak challenges in LLMs, revealing that while LLMs have been extensively tested for safety in English, multilingual applications present a whole new dimension of risks. The study identifies two scenarios: unintentional, where non-English prompts inadvertently bypass safety mechanisms, and intentional, where malicious instructions are deliberately embedded in multilingual prompts. The findings highlight the significant increase in unsafe content generation, especially for low-resource languages, and propose a "Self-Defense" framework for generating multilingual training data for safety fine-tuning.

This approach shows promise in substantially reducing the generation of unsafe content across languages.

Table 14 provides a comprehensive overview of studies that leverage large language models (LLMs) to detect and combat cyberattacks. The table systematically compares various studies based on key attributes, including the LLM model used, datasets employed, targeted applications, best performance values, and identified limitations. It highlights the diversity of LLM-based approaches, ranging from domain-specific models like SecureBERT and CySecBERT, tailored for cybersecurity text processing, to general-purpose LLMs such as GPT-4 and ChatGPT, evaluated in specific security contexts like prompt injection and spear phishing scenarios. This comparison underscores the versatility of LLMs in addressing different facets of cybersecurity challenges, such as vulnerability recognition, phishing prevention, and policy enhancement.

Table 14. Comparison of studies evaluating LLMs for detecting cyberattacks.

Ref.	ML Model Used	Dataset/Data Used	Application	Best Performance Value	Limitations
[20]	SecureBERT	Large corpus of cybersecurity text	Focuses on transforming CTI text into machine-readable format using SecureBERT.	Outperforms similar models in MLM and other standard NLP tasks	Not specified
[26]	CySecBERT	High-quality, domain-specific dataset	Tailored for the cysec domain, evaluates on domain-dependent and intrinsic tasks.	Best performance in cybersecurity scenario-specific tasks	Catastrophic forgetting during further training
[208]	GPT-4, Mistral, Meta Llama 3 70B-Instruct	Novel benchmark dataset for LLM security risks	Evaluates LLMs on prompt injection and code interpreter abuse, introducing FRR.	Shows capability in handling "borderline" benign requests	Unresolved risk of attack conditioning
[209]	Code Llama, DeepSeek-Coder, StarCoder2	Instruction tuning datasets with adversarial code injections	Assesses vulnerabilities of instruction-tuned Code LLMs using EvilInstructCoder.	High ASR@1 scores in adversarial attack scenarios	Significant vulnerability to adversarial attacks
[210]	ChatGPT, Google Bard, Microsoft Bing	Cisco certification exams, CTF challenge data	Investigates the effectiveness of LLMs in CTF exercises, highlighting jailbreak prompts.	Not specified	Ethical concerns and limitations in CTF applications
[172]	CyBERT	Cybersecurity corpus from CTI data	Focuses on recognizing specialized cybersecurity entities using a fine-tuned BERT model.	Outperforms base BERT model in domain-specific MLM evaluation	Not specified
[211]	Multiple BERT classifiers	Textual descriptions of security vulnerabilities	Automatically determines CVSS vectors and severity scores from textual vulnerability descriptions.	High accuracy in CVSS metric prediction	Requires extensive manual analysis for new vulnerabilities
[21]	Unspecified LLMs	SME case studies and LLM performance metrics in Australia	Explores the potential role of LLMs in enhancing cyber security policies for SMEs.	High relevance, accuracy, and applicability in cybersecurity settings	Gaps in completeness and clarity
[177]	OpenAI's GPT-3.5 and GPT-4	Spear phishing messages for British MPs	Examines the use of LLMs in spear phishing, focusing on message generation.	Cost-effective and realistic email generation	Potential misuse of LLMs in spear phishing
[212]	Google Gemini's generative AI	Corporate cybersecurity frameworks	Enhances detection, prevention, and response strategies against spear phishing attacks.	Improved accuracy and dynamic policy adjustments	Needs further exploration in broader AI applications

Key findings from the table include the superior performance of domain-specific models, such as CySecBERT and CyBERT, which excel in tasks requiring specialized knowledge. Additionally, general-purpose models like GPT-4 demonstrate adaptability in novel scenarios, such as handling borderline benign requests. However, the table also identifies significant limitations, including vulnerabilities to adversarial attacks, catastrophic forgetting, and ethical concerns in applying LLMs for potentially harmful scenarios like spear phishing. These insights emphasize the need for continued research to enhance the robustness, ethical use, and scalability of LLMs in cybersecurity applications.

The table further highlights critical gaps, such as the lack of specification in some studies regarding limitations and the need for more comprehensive benchmarks to evaluate LLMs effectively. These findings suggest that while LLMs offer promising solutions for cybersecurity, addressing their vulnerabilities and ethical implications is crucial to ensure their safe and effective deployment in real-world settings.

5.6. Technical Limitations

5.6.1. Interpretability

LLMs are often considered "black boxes", making it challenging to understand how they arrive at their decisions. This opacity inhibits the ability to diagnose errors, identify biases, and ensure the reliability of security operations. Without a clear understanding of how LLMs arrive at their decisions, cybersecurity professionals struggle to effectively troubleshoot errors, mitigate biases, and maintain trust in these systems. As a result, efforts to enhance interpretability through techniques such as visualization and explanation generation are crucial to improving the transparency and reliability of LLMs in cybersecurity applications.

5.6.2. Domain-Specific Knowledge

Another significant technical limitation of applying LLMs in cybersecurity is their potential lack of domain-specific knowledge, particularly in understanding intricate technical details or specific threat landscapes [213]. This deficiency could lead to inaccuracies or inefficiencies in threat detection and response processes. Without a deep understanding of cybersecurity concepts, LLMs may struggle to accurately interpret and contextualize security-related information, resulting in suboptimal performance in identifying and mitigating cyber threats. Addressing this limitation requires efforts to enhance LLMs' domain-specific knowledge through specialized training datasets, fine-tuning techniques, and collaboration with cybersecurity experts to ensure the models effectively capture and utilize relevant security information [41].

5.6.3. Scalability

The process of training and deploying these models at scale demands substantial computational resources. This requirement can pose challenges for organizations constrained by limited infrastructure or budgetary constraints. The resource-intensive nature of LLMs necessitates access to high-performance computing infrastructure, extensive storage capacities, and skilled personnel for model development and maintenance [214]. Additionally, the associated costs of hardware, software licenses, and energy consumption further compound the scalability challenges, potentially impeding the widespread adoption of LLM-based cybersecurity solutions [215]. To address this limitation, organizations may explore cloud-based solutions, distributed computing approaches, or model optimization techniques to mitigate the computational burdens and enhance the scalability of LLM deployments in cybersecurity contexts.

5.6.4. Adversarial Attacks

The susceptibility of exposing vulnerabilities where malicious actors manipulate input data to deceive the model and generate incorrect outputs compromises the effectiveness of LLM-based security systems, as attackers can exploit weaknesses in the model's decision-making process to evade detection, bypass security measures, circumvent security measures, or use the results for malicious purposes [216]. Adversarial attacks can manifest in a variety of ways, including poisoning attacks, stealth attacks, and model reversal attacks. To increase the resilience of LLMs and strengthen their defenses against strategic manipulation, it is necessary to develop complex security measures such as adversary train-

ing, intrusion purity, and anomaly detection methods [217]. Figure 2 shows the lifecycle of LLMs from data collection to monitoring and highlights where vulnerabilities can be introduced and exploited by cyberattacks. Each step includes potential attack vectors that need to be mitigated to ensure the secure and ethical deployment of LLMs.

5.7. Ethical Limitations

5.7.1. Bias and Fairness

LLMs trained on biased datasets have the propensity to perpetuate or exacerbate existing biases, potentially resulting in unfair or discriminatory outcomes in cybersecurity tasks. Addressing bias and ensuring fairness in LLMs is imperative to uphold equity and integrity in security operations. Failure to mitigate biases can lead to discriminatory practices, unequal treatment, and compromised trust in LLM-based security systems. Efforts to address this ethical limitation involve employing techniques such as bias detection, data preprocessing, fairness-aware training, and diverse dataset curation to mitigate bias and promote fairness in LLMs utilized for cybersecurity purposes [218]. By proactively addressing bias and promoting fairness, organizations can enhance the ethical integrity and societal impact of LLM-based cybersecurity solutions [219].

5.7.2. Privacy Concerns

The development of LLMs presents serious questions about the critical concerns regarding the privacy and confidentiality of sensitive information, including private messages and personal information [220]. Organizations using LLMs in cybersecurity initiatives must prioritize protecting people's rights and adhering to privacy requirements. Researchers and practitioners are investigating novel ways to guarantee privacy and secrecy in LLM-based cybersecurity applications in order to allay these worries [221]. Access restrictions, encryption, anonymization, and data minimization are some of the strategies used to safeguard sensitive data at every stage of their lifetime, from processing and collecting to storage and analysis. To further examine and improve the efficacy of privacy protection mechanisms, regulatory compliance, routine audits, and privacy impact evaluations are essential. Organizations can fully utilize LLMs in cybersecurity while protecting people's right to privacy and preserving confidence in data handling procedures by emphasizing privacy preservation and implementing strong privacy measures. In order to promote the responsible and reliable deployment of LLMs in cybersecurity areas, it is critical to close the gap between technological innovation and ethical considerations, as this multidisciplinary endeavor highlights [222].

5.7.3. Misuse and Manipulation

LLMs can be exploited by malicious actors to generate deceptive content, such as phishing emails or fake news, for nefarious purposes [221]. This misuse raises ethical concerns regarding the responsible use of LLMs and the potential harm inflicted on individuals and organizations. The ability of LLMs to generate convincing yet fabricated content underscores the importance of implementing safeguards and ethical guidelines to prevent their exploitation for malicious intents [223]. Addressing this limitations promote responsible use, implement content verification, and raise awareness about risks. Collaboration among policymakers, researchers, and industry experts is crucial for developing strategies to mitigate misuse and manipulation [224]. Proactively tackling these ethical concerns ensures responsible LLM deployment and upholds ethical standards in cybersecurity.

5.7.4. Accountability and Transparency

The use of LLMs in cybersecurity requires clear accountability and transparency mechanisms to ensure responsible decision-making and mitigate potential risks. Organizations

must be transparent about how LLMs are trained, deployed, and evaluated to maintain trust and accountability [225]. Although LLMs offer significant advantages in a variety of applications, their susceptibility to cyberattacks poses serious challenges. Ongoing research and development is required to improve the robustness and safety of these models, ensuring that they can be safely used in critical applications and this is shown in Figure 3.

6. LLMs for Cybersecurity Tools

Recent advancements in cybersecurity research and operations have made LLMs an extremely powerful tool, revolutionizing the way security activities are handled and carried out. These models' capacity to understand, generate, and manipulate natural language text has found diverse applications in reversing engineering, network analysis, cloud security, and even in conceptualizing proofs of concept for both defensive mechanisms and potential cyber threats.

6.1. Reverse Engineering Applications

In the domain of reverse engineering, tools like G-3PO, ai for Pwndbg, and ai for GEF exemplify the integration of LLMs to facilitate code analysis and debugging. G-3PO, developed by Olivia Lucca Fraser at Tenable, is designed to work within the Ghidra software framework, offering an AI-powered assistant that leverages OpenAI and Anthropic's models to annotate and provide insights on decompiled code [226]. Similarly, Fraser's development of ai for Pwndbg and ai for GEF introduces AI capabilities into the debugging process, enhancing the efficiency and depth of analysis for security professionals [227]. Gepetto and GPT-WPRE further extend the application of LLMs in reverse engineering, offering tools for IDA Pro and Ghidra, respectively, to generate explanatory comments and summarize binary analyses, thereby making complex code more accessible and understandable [157,228].

6.2. Network Analysis

Network security also benefits from LLM innovations, as seen with BurpGPT. This BurpSuite plugin, developed by Yossi Nisani at Tenable, employs GPT models to analyze HTTP requests and responses, highlighting potential security vulnerabilities and offering insights that can guide further investigation and remediation efforts [229].

6.3. Cloud Security

Cloud security, particularly concerning identity access and management (IAM) policies, has seen the introduction of tools like EscalateGPT. This tool utilizes GPT to identify and explore privilege escalation vulnerabilities in AWS IAM configurations, showcasing the potential of LLMs to navigate and secure complex cloud environments.

6.4. Proofs of Concept

The versatility of LLMs extends into the development of proofs of concept that both demonstrate potential cybersecurity threats and explore innovative defenses. Examples include Indirect Prompt Injections and LLMorphism, which reveal new vectors for cyberattacks and malware development. These applications underscore the dual-edged nature of LLM capabilities, serving both to enhance security postures and to highlight novel vulnerabilities (Figures 8 and 9).

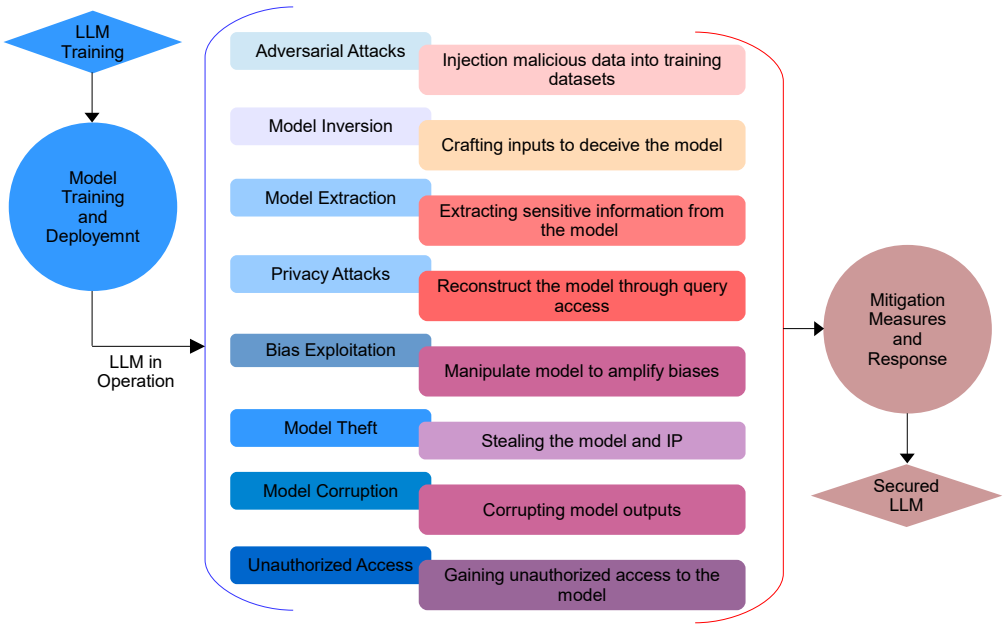

Figure 8. The processes and vulnerabilities of LLMs to cyberattacks.

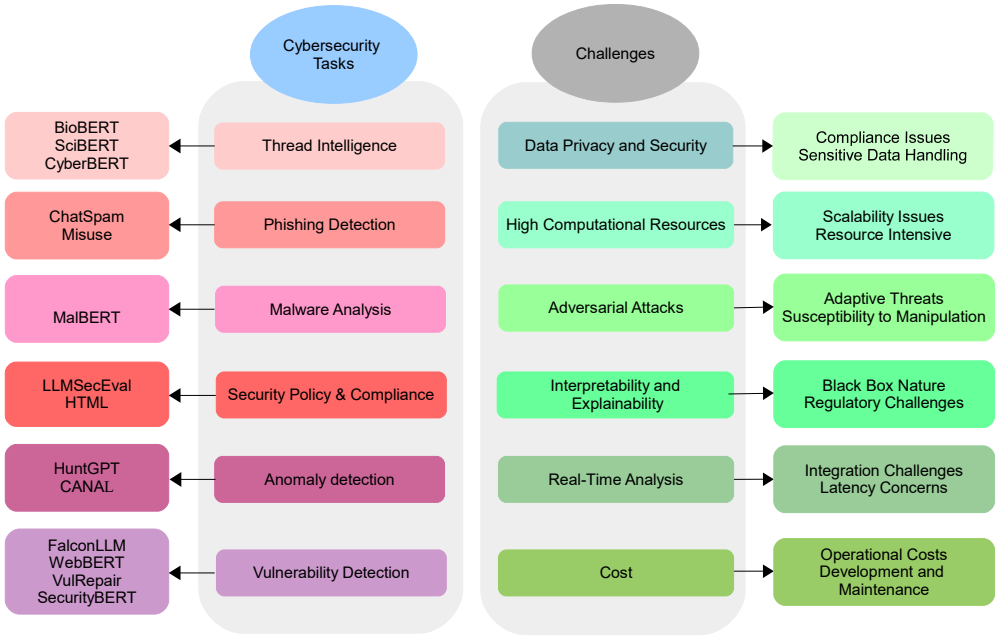

Figure 9. Cybersecuritiy for LLMs and thier limitations.

7. Challenges of Deploying LLMs in Cybersecurity

7.1. Data Privacy and Security

LLMs can unintentionally memorize and output sensitive data, such as personal information or proprietary data, during their operations. This issue arises because LLMs are often trained on vast datasets containing personal or confidential data, leading to potential risks of violating privacy regulations such as the GDPR (General Data Protection Regulation) or HIPAA (Health Insurance Portability and Accountability Act). For instance, when users interact with LLM-based cybersecurity tools, the model might expose sensitive information stored in its training data [230]. To address this, mechanisms like differential privacy can be implemented to minimize the risk of such data being memorized or leaked.

However, there is still ongoing research to make LLMs more privacy-preserving while maintaining performance.

7.2. Integration with Existing Systems

Integrating LLMs into existing cybersecurity infrastructures can be challenging. Many enterprises have well-established security infrastructures that rely on specific tools, workflows, and protocols, which can be difficult to harmonize with LLM-based solutions [231]. This includes the complexity of integrating LLMs with existing Security Information and Event Management (SIEM) systems, intrusion detection systems (IDSs), and automated response tools [232]. Achieving seamless integration often requires significant customization, API development, and adjustments to infrastructure, making the deployment process slow and costly. Compatibility issues with legacy systems further complicate this integration. LLMs also require real-time access to relevant data, which may be constrained by organizational data silos or privacy regulations (ar5iv) [233].

Moreover, ensuring that LLMs operate within the strict security and compliance frameworks necessary in cybersecurity environments is critical. LLMs must align with specific governance policies, creating an additional layer of complexity in deployment [234]. Handling large-scale threat intelligence data efficiently, with minimal latency, is another obstacle when integrating with existing systems [235]. In 2023, recent studies explored hybrid models and modular architectures to address these challenges, including research on using microservices and containerized LLMs to simplify integration. By isolating core functionalities of LLMs, it is easier to plug them into existing cybersecurity tools, reducing overhead and improving interoperability. However, a lot of work still needs to be performed in optimizing these processes for seamless, secure integration at scale [59].

7.3. Robustness and Security

LLMs are susceptible to adversarial attacks, where small changes in input can manipulate the model's behavior, leading to incorrect predictions or malicious outputs. In cybersecurity applications, such vulnerabilities can be catastrophic [236]. For example, attackers can introduce adversarial samples that mislead an LLM-powered threat detection system into classifying a malicious action as benign. Furthermore, there are risks from prompt injection attacks, where manipulating input prompts can alter the model's responses to generate incorrect or harmful outputs. Enhancing the robustness of LLMs against such attacks requires careful fine-tuning and the integration of adversarial training methods to mitigate these threats.

7.4. Contextual Accuracy

In cybersecurity, accurate context interpretation is critical for detecting sophisticated threats. LLMs, while powerful, may sometimes generate inaccurate or non-faithful explanations, especially when tasked with multi-step reasoning or analyzing complex log files. This problem, known as the "faithfulness" issue, affects LLMs when their generated explanations do not accurately represent the underlying data or analysis. For cybersecurity applications like insider threat detection, where logs are vast and involve complex behaviors, this can lead to false positives or missed threats [144]. Techniques like multi-agent collaboration or evidence-based debates (such as Audit-LLM) have been proposed to enhance the reasoning capabilities of LLMs, improving their reliability in such tasks.

7.5. Scalability and Performance

Real-time cybersecurity applications, such as intrusion detection systems, require fast processing of massive amounts of data. Deploying LLMs in such environments poses scalability challenges, as LLMs are computationally intensive and require significant

resources, such as memory and processing power. This is especially problematic for real-time threat detection, where delays can lead to missed alerts or slow response times [41]. Optimizing LLMs for scalability without compromising their accuracy is an ongoing research area. Techniques such as model distillation, edge computing, and optimized inference can help scale LLM deployments in cybersecurity settings.

7.6. Continual Learning and Adaptation

LLMs must continuously adapt to new cyber threats, tools, and evolving attack methods, making it vital for them to undergo ongoing training without forgetting previously acquired knowledge [237]. Traditional models often suffer from "catastrophic forgetting" when fine-tuned on new data, which can degrade their performance in previously learned tasks. To overcome this, continual learning techniques are being explored to help LLMs integrate new knowledge while retaining existing capabilities [238]. For instance, methods like Continual Pre-training (CPT) aim to incrementally update models with facts, domains, or tasks relevant to evolving cybersecurity needs [239]. However, ensuring that these updates happen efficiently, without bloating the models or causing ethical misalignment, is a key technical hurdle [240].

8. Future Directions

This article explores several potential future directions for LLMs in cybersecurity, including the detection of social engineering attempts, automation of incident responses, disruption of phishing campaigns, and tailoring security education programs. By examining these key areas of research and innovation, organizations can gain valuable insights into how LLMs can be leveraged to bolster cybersecurity defenses and mitigate emerging threats effectively.

8.1. Detection of Social Engineering Attempts

Future studies might concentrate on using LLMs to identify and counteract social engineering activities, such pretexting and phishing scams. LLMs can assist in spotting questionable communications and shield users from social engineering techniques by examining linguistic patterns and contextual clues [241]. Using a variety of datasets of social engineering scenarios, this method trains LLMs to better identify manipulation tactics and fraudulent language. Additionally, in order to automatically identify and address possible social engineering threats in real time, researchers may investigate the integration of LLMs with cybersecurity systems. Developments in this field could greatly strengthen an organization's defenses against social engineering assaults and lower the likelihood that private data would be compromised [242].

8.2. Automation of Incident Responses

In order to automate incident response procedures, future studies may investigate the integration of LLMs with security orchestration, automation, and response (SOAR) platforms. Organizations can improve their cybersecurity posture and resilience by streamlining incident detection, analysis, and remediation by utilizing LLMs' natural language processing capabilities [243]. Using this method entails creating workflows and algorithms based on LLM that can automatically evaluate security alerts, correlate threat intelligence, and plan response actions according to pre-established playbooks. Researchers may also look into the use of LLMs for dialogue management and natural language comprehension to provide more complex and context-aware incident response interactions with stakeholders and security analysts. Developments in this field could lessen the need for manual intervention, increase the effectiveness of incident response, and lessen the negative effects of cybersecurity events on an organization's operations and data assets [244].

8.3. Disruption of Phishing Campaigns

Researchers may explore novel approaches for leveraging LLMs to disrupt phishing campaigns and thwart malicious actors' efforts to deceive users. By harnessing LLMs' natural language generation capabilities, organizations can generate deceptive content or craft targeted responses to phishing emails, thereby undermining the effectiveness of phishing attacks and protecting sensitive information [245]. This approach involves training LLMs on a diverse range of phishing scenarios and techniques to enable them to recognize and counteract common phishing tactics, such as spoofed emails, fake websites, and social engineering ploys [63]. Additionally, researchers may investigate the use of LLMs for proactive threat hunting and deception operations, where LLMs generate decoy data or bait content to lure and expose phishing attackers. Advancements in this area have the potential to disrupt phishing campaigns, reduce the success rate of phishing attacks, and enhance organizations' resilience against social engineering threats [246].

8.4. Tailoring Security Education Programs

Future research could explore the potential of leveraging LLM-generated insights and analytics to tailor security education programs for individuals and organizations. By analyzing user behavior, identifying common misconceptions, and understanding prevalent vulnerabilities, LLMs can provide valuable insights into the specific cybersecurity knowledge gaps and learning needs of different user groups [247]. This approach involves developing LLM-based algorithms and models that can analyze user interactions, identify areas of weakness, and recommend personalized training modules or educational materials to address specific cybersecurity risks and threats [248]. Additionally, researchers may investigate the use of LLMs for simulating realistic cyberattack scenarios and interactive training exercises to enhance user engagement and effectiveness. Advancements in this area have the potential to improve the overall cybersecurity awareness and resilience of individuals and organizations, ultimately reducing the likelihood of successful cyberattacks and data breaches [152].

9. Conclusions

The integration of LLMs into cybersecurity frameworks marks a paradigm shift in how organizations address and mitigate digital threats. By leveraging advanced natural language processing capabilities, LLMs have shown remarkable potential to revolutionize several facets of cybersecurity, from real-time threat detection and malware analysis to phishing prevention and incident response. These models, characterized by their capacity to process and generate human-like text, enable security systems to parse unstructured data, identify emerging risks, and automate responses with unprecedented efficiency and precision.

A critical contribution of LLMs lies in their ability to detect and analyze sophisticated cyber threats. Their capacity to sift through vast and diverse datasets, ranging from security logs to open-source intelligence, empowers organizations to stay ahead of evolving attack vectors. For instance, LLMs enhance phishing detection by recognizing subtle linguistic cues, while their application in malware analysis allows for accurate classification and behavior prediction, even for novel threats. Moreover, these models significantly contribute to workforce training by simulating realistic cyber scenarios, enabling organizations to improve preparedness and resilience.

However, the deployment of LLMs in cybersecurity also raises significant challenges. One of the foremost concerns is interpretability. The "black-box" nature of LLMs makes it difficult for security analysts to understand their decision-making processes, leading to potential issues in diagnosing errors or addressing biases. Moreover, the resource-intensive

nature of LLMs, requiring substantial computational power and infrastructure, limits their scalability and accessibility for smaller organizations. Ethical concerns, such as the potential misuse of LLMs for generating malicious content like phishing emails or malware, further underscore the need for stringent governance and accountability mechanisms.

Privacy considerations are another critical limitation. The risk of exposing sensitive or proprietary data during training or inference processes necessitates robust privacy-preserving techniques. Strategies such as differential privacy, data anonymization, and encrypted computations are essential to mitigate these risks and ensure compliance with regulatory frameworks like the GDPR and HIPAA. Additionally, adversarial attacks, including prompt injections and model manipulations, pose a significant threat to the reliability of LLM-based systems, highlighting the need for ongoing advancements in adversarial defense mechanisms.

Despite these challenges, the future of LLMs in cybersecurity is promising. Emerging research focuses on enhancing the robustness and ethical alignment of these models, with developments in areas such as continual learning, automated attack detection, and collaborative multi-agent systems. Furthermore, integrating LLMs with existing security frameworks, such as Security Information and Event Management (SIEM) systems and security orchestration, automation, and response (SOAR) platforms, can significantly enhance their operational impact. Tailored security education programs, powered by LLM analytics, offer a proactive approach to mitigating user vulnerabilities and fostering a culture of cybersecurity awareness.

All in all, while LLMs are not a panacea for all cybersecurity challenges, they represent a powerful tool for augmenting traditional security measures and addressing complex threats. Their ability to analyze, predict, and respond to cyber risks positions them as invaluable assets in the fight against an increasingly sophisticated threat landscape. However, to fully realize their potential, it is imperative to address their limitations through interdisciplinary collaboration, regulatory oversight, and continuous innovation. By doing so, LLMs can pave the way for a more secure and resilient digital future.

Author Contributions: W.K.: Conceptualization, methodology, data curation, formal analysis, visualization, writing—original draft. Y.H.: Conceptualization, methodology, data curation, formal analysis, visualization, writing—original draft, writing—review & editing, project administration, supervision. H.A.A.: Methodology, formal analysis, visualization, writing—review & editing, supervision. S.T.: Methodology, formal analysis, visualization, writing—review & editing, supervision. S.A.: Methodology, formal analysis, visualization, writing—review & editing, supervision. W.M.: Methodology, formal analysis, visualization, writing—review & editing, project administration. H.A.-A.: Methodology, formal analysis, visualization, writing—review & editing, project administration. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data will be shared upon request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

AI	Artificial Intelligence
CSPM	Cloud Security Posture Management
LLM	Large Language Model
GPT	Generative Pre-trained Transformer
APTs	Advanced Persistent Threats

ML	Machine Learning
T5	Text-to-Text Transfer Transformer
PPFLE	Privacy-Preserving Fixed-Length Encoding
CTF	Capture the Flag
XAI	Explainable Artificial Intelligence
TPR	True Positive Rate
IPSDM	Improved Phishing and Spam Detection Model
URLTran	URL Transformer
GRU	Gated Recurrent Unit
CFG	Control-Flow Graph
CAN	Controller Area Network
NF	NetFlow Dataset
IVN-IDS	In-Vehicle Network Intrusion Detection System
NID	Network Intrusion Detection
PLLM-CS	Privacy-aware Large Language Model for Cybersecurity
VulnLLMEval	Vulnerability Large Language Model Evaluation Framework
LLMPATCH	Large Language Model-Based Automated Patching System
NER	Named-Entity Recognition
KGV	Knowledge Graph Verifier
Xpert	Expert Query Recommendation Framework
AuditGPT	Audit Generative Pre-trained Transformer
LEGILM	Legal and Regulatory Compliance Framework using LLMs
DrSec	Endpoint Detection and Response Security System
FalconLLM	Secure Fine-Tuned Large Language Model
RA-LLM	Robustly Aligned Large Language Model
ChatSpam	Chat-based Spam Detection Framework
CipherChat	Framework for Evaluating LLM Safety Against Ciphers
G-3PO	Ghidra AI Assistant for Code Analysis
EscalateGPT	GPT for Privilege Escalation in Cloud IAM Policies
SecureBERT	Security-Focused Bidirectional Encoder Representation
Multilingual Self-Defense	Framework for Generating Multilingual Safety Data
IoT	Internet of Things
RCA	Root Cause Analysis
NVD	National Vulnerability Database
BERT	Bidirectional Encoder Representations from Transformers
DL	Deep Learning
NLP	Natural Language Processing
RAG	Retrieval-Augmented Generation
BBPE	Byte-Level Byte Pair Encoding
CWE	Common Weakness Enumeration
ANN	Artificial Neural Network
FPR	False Positive Rate
RBPD	Reference-Based Phishing Detector
KPD	KnowPhish Detector
LSTM	Long Short-Term Memory
GIN	Graph Isomorphism Network
IDS	Intrusion Detection System
CGAN	Conditional Generative Adversarial Network
BoT-IoT	Botnet Internet of Things
TON_IoT	ToN IoT Dataset
AMG	Adversarial Malware Generation
VulDetectBench	Vulnerability Detection Benchmark
CTI	Cyber Threat Intelligence
RE	Relation Extraction

WILEE	Weighted Interactive Learning Environment for Exploration
IcM BRAIN	Incident Management Brain
Audit-LLM	Multi-Agent Framework for Insider Threat Detection
BIM	Building Information Modeling
CodeAttack	Adversarial Framework for Testing LLM Safety in Code Generation
LLM-TIKG	LLM for Threat Intelligence Knowledge Graph Construction
AutoDefense	Multi-Agent Defense Framework for LLMs
JailbreakGPT	Attack on LLMs to Elicit Unintended Outputs
AutoDAN	Automated Defense Against Adversarial Networks
BurpGPT	BurpSuite Plugin for HTTP Analysis with GPT
EvilInstructCoder	Framework for Evaluating Adversarial Vulnerabilities in Code LLMs
CySecBERT	Domain-Specific BERT for Cybersecurity Tasks
Cipher Safety Alignment	Extension of Safety Measures to Non-Natural Languages

References

1. Saeed, S.; Altamimi, S.A.; Alkayyal, N.A.; Alshehri, E.; Alabbad, D.A. Digital transformation and cybersecurity challenges for businesses resilience: Issues and recommendations. *Sensors* **2023**, *23*, 6666. [CrossRef] [PubMed]
2. Sharma, A.; Gupta, B.B.; Singh, A.K.; Saraswat, V. Advanced persistent threats (APT): Evolution, anatomy, attribution and countermeasures. *J. Ambient Intell. Human. Comput.* **2023**, *14*, 9355–9381. [CrossRef]
3. Areeb, Q.M.; Nadeem, M.; Sohail, S.S.; Imam, R.; Doctor, F.; Himeur, Y.; Hussain, A.; Amira, A. Filter bubbles in recommender systems: Fact or fallacy—A systematic review. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2023**, *13*, e1512. [CrossRef]
4. Himeur, Y.; Sohail, S.S.; Bensaali, F.; Amira, A.; Alazab, M. Latest trends of security and privacy in recommender systems: A comprehensive review and future perspectives. *Comput. Secur.* **2022**, *118*, 102746. [CrossRef]
5. Salem, A.H.; Azzam, S.M.; Emam, O.; Abohany, A.A. Advancing cybersecurity: A comprehensive review of AI-driven detection techniques. *J. Big Data* **2024**, *11*, 105. [CrossRef]
6. Himeur, Y.; Boukabou, A. A robust and secure key-frames based video watermarking system using chaotic encryption. *Multimed. Tools Appl.* **2018**, *77*, 8603–8627. [CrossRef]
7. Pruemmer, J.; van Steen, T.; van den Berg, B. A systematic review of current cybersecurity training methods. *Comput. Secur.* **2024**, *136*, 103585. [CrossRef]
8. Du, M.; He, F.; Zou, N.; Tao, D.; Hu, X. Shortcut learning of large language models in natural language understanding: A survey. *arXiv* **2022**, arXiv:2208.11857. [CrossRef]
9. Sohail, S.S.; Farhat, F.; Himeur, Y.; Nadeem, M.; Madsen, D.Ø.; Singh, Y.; Atalla, S.; Mansoor, W. The future of gpt: A taxonomy of existing chatgpt research, current challenges, and possible future directions. *Curr. Chall. Possible Future Dir.* **2023**. [CrossRef]
10. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
11. Khennouche, F.; Elmir, Y.; Himeur, Y.; Djebari, N.; Amira, A. Revolutionizing generative pre-traineds: Insights and challenges in deploying ChatGPT and generative chatbots for FAQs. *Expert Syst. Appl.* **2024**, *246*, 123224. [CrossRef]
12. Farhat, F.; Silva, E.S.; Hassani, H.; Madsen, D.Ø.; Sohail, S.S.; Himeur, Y.; Alam, M.A.; Zafar, A. The scholarly footprint of ChatGPT: A bibliometric analysis of the early outbreak phase. *Front. Artif. Intell.* **2024**, *6*, 1270749. [CrossRef] [PubMed]
13. Sohail, S.S.; Madsen, D.Ø.; Himeur, Y.; Ashraf, M. Using ChatGPT to navigate ambivalent and contradictory research findings on artificial intelligence. *Front. Artif. Intell.* **2023**, *6*, 1195797. [CrossRef]
14. Papageorgiou, E.; Chronis, C.; Varlamis, I.; Himeur, Y. A survey on the use of large language models (llms) in fake news. *Future Internet* **2024**, *16*, 298. [CrossRef]
15. 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]
16. Safitra, M.F.; Lubis, M.; Fakhrurroja, H. Counterattacking cyber threats: A framework for the future of cybersecurity. *Sustainability* **2023**, *15*, 13369. [CrossRef]
17. Arabo, A. Cyber security challenges within the connected home ecosystem futures. *Procedia Comput. Sci.* **2015**, *61*, 227–232. [CrossRef]
18. Humayun, M.; Niazi, M.; Jhanjhi, N.; Alshayeb, M.; Mahmood, S. Cybersecurity threats and vulnerabilities: A systematic mapping study. *Arab. J. Sci. Eng.* **2020**, *45*, 3171–3189. [CrossRef]
19. Kaur, J.; Ramkumar, K. The recent trends in cybersecurity: A review. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 5766–5781.

20. Aghaei, E.; Niu, X.; Shadid, W.; Al-Shaer, E. SecureBERT: A Domain-Specific Language Model for Cybersecurity. In Proceedings of the International Conference on Security and Privacy in Communication Systems, Kansas City, MO, USA, 17–19 October 2022; Springer Nature: Cham, Switzerland, 2022; pp. 39–56.
21. Kereopa-Yorke, B. Building resilient SMEs: Harnessing large language models for cyber security in Australia. *J. AI Robot. Workplace Autom.* **2024**, *3*, 15–27. [CrossRef]
22. Fu, M.; Wang, P.; Liu, M.; Zhang, Z.; Zhou, X. IoV-BERT-IDS: Hybrid Network Intrusion Detection System in IoV Using Large Language Models. *IEEE Trans. Vehicul. Technol.* **2024**. [CrossRef]
23. Zhang, A.K.; Perry, N.; Dulepet, R.; Ji, J.; Lin, J.W.; Jones, E.; Menders, C.; Hussein, G.; Liu, S.; Jasper, D.; et al. Cybench: A Framework for Evaluating Cybersecurity Capabilities and Risks of Language Models. *arXiv* **2024**, arXiv:2408.08926.
24. Zaboli, A.; Choi, S.L.; Song, T.J.; Hong, J. Chatgpt and other large language models for cybersecurity of smart grid applications. In Proceedings of the 2024 IEEE Power & Energy Society General Meeting (PESGM), Seattle, WA, USA, 21–25 July 2024; pp. 1–5.
25. Xu, J.; Stokes, J.W.; McDonald, G.; Bai, X.; Marshall, D.; Wang, S.; Swaminathan, A.; Li, Z. Autoattacker: A large language model guided system to implement automatic cyber-attacks. *arXiv* **2024**, arXiv:2403.01038.
26. Bayer, M.; Kuehn, P.; Shanehsaz, R.; Reuter, C. Cysecbert: A domain-adapted language model for the cybersecurity domain. *ACM Trans. Priv. Secur.* **2024**, *27*, 1–20. [CrossRef]
27. McIntosh, T.R.; Susnjak, T.; Liu, T.; Watters, P.; Xu, D.; Liu, D.; Nowrozy, R.; Halgamuge, M.N. From COBIT to ISO 42001: Evaluating cybersecurity frameworks for opportunities, risks, and regulatory compliance in commercializing large language models. *Comput. Secur.* **2024**, *144*, 103964. [CrossRef]
28. Liang, J. Machine Learning in Cybersecurity Training. *Cyber Train. Rev.* **2021**.
29. Chopra, S.; Ahmad, H.; Goel, D.; Szabo, C. ChatNVD: Advancing Cybersecurity Vulnerability Assessment with Large Language Models. *arXiv* **2024**, arXiv:2412.04756.
30. Lipton, Z.C. The Mythos of Model Interpretability. *Commun. ACM* **2018**, *16*, 31–57.
31. Hodge, R. AI-Driven Cybersecurity Ecosystems. *AI Soc.* **2021**.
32. Zoph, B.; Ghiasi, G.; Lin, T.-Y.; Cui,Y.; Liu, H.; Cubuk, E.D.; Le, Q.V. Rethinking Pretraining and Fine-tuning in AI. *AI Adv.* **2020**.
33. Hassanin, M.; Keshk, M.; Salim, S.; Alsubaie, M.; Sharma, D. PLLM-CS: Pre-trained Large Language Model (LLM) for cyber threat detection in satellite networks. *Ad Hoc Netw.* **2025**, *166*, 103645. [CrossRef]
34. Chen, M.; Zhu, K.; Lu, B.; Li, D.; Yuan, Q.; Zhu, Y. AECR: Automatic attack technique intelligence extraction based on fine-tuned large language model. *Comput. Secur.* **2025**, *150*, 104213. [CrossRef]
35. Zhang, Y.; Du, T.; Ma, Y.; Wang, X.; Xie, Y.; Yang, G.; Lu, Y.; Chang, E.C. AttacKG+: Boosting attack graph construction with Large Language Models. *Comput. Secur.* **2025**, *150*, 104220. [CrossRef]
36. Tu, N.; Nam, S.; Hong, J.W.K. Intent-Based Network Configuration Using Large Language Models. *Int. J. Netw. Manag.* **2025**, *35*, e2313. [CrossRef]
37. Tihanyi, N.; Bisztray, T.; Ferrag, M.A.; Jain, R.; Cordeiro, L.C. How secure is AI-generated code: A large-scale comparison of large language models. *Empir. Softw. Eng.* **2025**, *30*, 1–42. [CrossRef]
38. Dharmendra, H.; Raghunandan, G.; Sindhu, A.; Samanvitha, C.; Nethravathi, N.; Elango, D. Human Evaluation in Large Language Model Testing: Assessing the Quality of AI Model Output. In *Advancements in Intelligent Process Automation*; IGI Global: Hershey, PA, USA, 2025; pp. 553–574.
39. Zhong, H.; Zhang, Q.; Li, W.; Lin, R.; Tang, Y. KPLLM-STE: Knowledge-enhanced and prompt-aware large language models for short-text expansion. *World Wide Web* **2025**, *28*, 1–25. [CrossRef]
40. Luo, J.; Chen, Z.; Chen, W.; Lu, H.; Lyu, F. A study on the application of the T5 large language model in encrypted traffic classification. *Peer-Netw. Appl.* **2025**, *18*, 1–13. [CrossRef]
41. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; 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.
42. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
43. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S.; et al. PaLM: Scaling language models with pathways. *arXiv* **2022**, arXiv:2204.02311.
44. Technology Innovation Institute. Falcon 180B: A State-of-the-Art Open-Source LLM. Technical Report. 2023. Available online: https://www.tii.ae/news/technology-innovation-institute-introduces-worlds-most-powerful-open-llm-falcon-180b (accessed on 21 May 2024).
45. Community, S. Vicuna 13-B: A Fine-Tuned LLM for Conversational AI. Project Documentation. 2023. Available online: https://lmsys.org/blog/2023-03-30-vicuna/ (accessed on 21 May 2024).
46. Salesforce. XGen-7B: A Robust Model for Long-Context Text Generation. Technical Report. Available online: https://clarifai.com/salesforce/xgen/models/xgen-7b-8k-instruct (accessed on 21 May 2024).

47. Scao, T.L.; Fan, A.; Wolf, T.; Akiki, C.; Pavlick, E.; Ilić, S.; Hesslow, D.; Castagné, R.; Luccioni, A.S.; Yvon, F.; et al. BLOOM: A multilingual open-source LLM. *arXiv* **2022**, arXiv:2211.05100.
48. AI, M. Mistral 7B: Optimized for NLP and Generative Tasks. Technical Report. Available online: https://mistral.ai/news/announcing-mistral-7b/ (accessed on 21 May 2024).
49. EleutherAI. GPT-NeoX: Open-Source Large-Scale Language Modeling. Project Documentation. 2022. Available online: https://github.com/EleutherAI/gpt-neox (accessed on 21 May 2024).
50. Research, M.A. LLaMA 2: Open Foundation and Fine-Tuned Models for Research Use. Technical Report. 2023. Available online: https://ai.meta.com/research/publications/llama-2-open-foundation-and-fine-tuned-chat-models/ (accessed on 21 May 2024).
51. Microsoft. Turing-NLG: Microsoft's Generative Pre-Trained Language Model. Technical Report. 2020. Available online: https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/ (accessed on 21 May 2024).
52. Tihanyi, N.; Ferrag, M.A.; Jain, R.; Debbah, M.O. CyberMetric: A Benchmark Dataset for Evaluating Large Language Models in Cybersecurity Knowledge. *arXiv* **2024**, arXiv:2402.07688.
53. Liu, Z. SecQA: Question-Answering Benchmark Dataset for Evaluating LLMs in Computer Security. *arXiv* **2023**, arXiv:2312.15838.
54. Shao, M.; Jancheska, S.; Udeshi, M.; Dolan-Gavitt, B.; Xi, H.; Milner, K.; Chen, B.; Yin, M.; Garg, S.; Krishnamurthy, P.; et al. NYU CTF Dataset: Evaluating LLMs on Capture the Flag Challenges for Offensive Security Tasks. *arXiv* **2024**, arXiv:2406.05590.
55. Rydén, A.; Näslund, E.; Schiller, E.M.; Almgren, M. LLMSecCode: Evaluating large language models for secure coding. In *International Symposium on Cyber Security, Cryptology, and Machine Learning*; Springer Nature: Cham, Switzerland, 2024; pp. 100–118.
56. Silvestri, S.; Islam, S.; Papastergiou, S.; Tzagkarakis, C.; Ciampi, M. A machine learning approach for the NLP-based analysis of cyber threats and vulnerabilities of the healthcare ecosystem. *Sensors* **2023**, *23*, 651. [CrossRef] [PubMed]
57. Thapa, C.; Jang, S.I.; Ahmed, M.E.; Camtepe, S.; Pieprzyk, J.; Nepal, S. Transformer-based language models for software vulnerability detection. In Proceedings of the 38th Annual Computer Security Applications Conference, Austin, TX, USA, 5–9 December 2022; pp. 481–496.
58. Rahali, A.; Akhloufi, M.A. MalBERT: Malware detection using bidirectional encoder representations from transformers. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), Melbourne, Australia, 17–20 October 2021; pp. 3226–3231.
59. Ferrag, M.A.; Ndhlovu, M.; Tihanyi, N.; Cordeiro, L.C.; Debbah, M.; Lestable, T.; Thandi, N.S. Revolutionizing cyber threat detection with large language models: A privacy-preserving bert-based lightweight model for iot/iiot devices. *IEEE Access* **2024**, *12*, 23733–23750. [CrossRef]
60. Shen, Y.; Heacock, L.; Elias, J.; Hentel, K.D.; Reig, B.; Shih, G.; Moy, L. ChatGPT and other large language models are double-edged swords. *Radiology* **2023**, *307*, e230163. [CrossRef] [PubMed]
61. Ullah, F.; Naeem, H.; Jabbar, S.; Khalid, S.; Latif, M.A.; Al-Turjman, F.; Mostarda, L. Cyber security threats detection in internet of things using deep learning approach. *IEEE Access* **2019**, *7*, 124379–124389. [CrossRef]
62. Trad, F.; Chehab, A. Prompt engineering or fine-tuning? a case study on phishing detection with large language models. *Mach. Learn. Knowl. Extr.* **2024**, *6*, 367–384. [CrossRef]
63. Koide, T.; Fukushi, N.; Nakano, H.; Chiba, D. Chatspamdetector: Leveraging large language models for effective phishing email detection. *arXiv* **2024**, arXiv:2402.18093.
64. Chataut, R.; Gyawali, P.K.; Usman, Y. Can ai keep you safe? a study of large language models for phishing detection. In Proceedings of the 2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 8–10 January 2024; pp. 0548–0554.
65. Lee, J.; Lim, P.; Hooi, B.; Divakaran, D.M. Multimodal Large Language Models for Phishing Webpage Detection and Identification. *arXiv* **2024**, arXiv:2408.05941.
66. Patel, H.; Rehman, U.; Iqbal, F. Large Language Models Spot Phishing Emails with Surprising Accuracy: A Comparative Analysis of Performance. *arXiv* **2024**, arXiv:2404.15485.
67. Roy, S.S.; Nilizadeh, S. Utilizing Large Language Models to Optimize the Detection and Explainability of Phishing Websites. *arXiv* **2024**, arXiv:2408.05667.
68. Bethany, M.; Galiopoulos, A.; Bethany, E.; Karkevandi, M.B.; Vishwamitra, N.; Najafirad, P. Large language model lateral spear phishing: A comparative study in large-scale organizational settings. *arXiv* **2024**, arXiv:2401.09727.
69. Mahendru, S.; Pandit, T. SecureNet: A Comparative Study of DeBERTa and Large Language Models for Phishing Detection. *arXiv* **2024**, arXiv:2406.06663.
70. Heiding, F.; Schneier, B.; Vishwanath, A.; Bernstein, J.; Park, P.S. Devising and detecting phishing emails using large language models. *IEEE Access* **2024**, *12*, 42131–42146. [CrossRef]
71. Uddin, M.A.; Sarker, I.H. An Explainable Transformer-based Model for Phishing Email Detection: A Large Language Model Approach. *arXiv* **2024**, arXiv:2402.13871.

72. Benavides-Astudillo, E.; Fuertes, W.; Sanchez-Gordon, S.; Nuñez-Agurto, D.; Rodríguez-Galán, G. A phishing-attack-detection model using natural language processing and deep learning. *Appl. Sci.* **2023**, *13*, 5275. [CrossRef]
73. Li, Y.; Huang, C.; Deng, S.; Lock, M.L.; Cao, T.; Oo, N.; Hooi, B.; Lim, H.W. KnowPhish: Large Language Models Meet Multimodal Knowledge Graphs for Enhancing Reference-Based Phishing Detection. *arXiv* **2024**, arXiv:2403.02253.
74. Haynes, K.; Shirazi, H.; Ray, I. Lightweight URL-based phishing detection using natural language processing transformers for mobile devices. *Procedia Comput. Sci.* **2021**, *191*, 127–134. [CrossRef]
75. Wang, H.; Hooi, B. Automated Phishing Detection Using URLs and Webpages. *arXiv* **2024**, arXiv:2408.01667.
76. Maneriker, P.; Stokes, J.W.; Lazo, E.G.; Carutasu, D.; Tajaddodianfar, F.; Gururajan, A. Urltran: Improving phishing url detection using transformers. In Proceedings of the MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM), San Diego, CA, USA, 29 November–2 December 2021; pp. 197–204.
77. Jamal, S.; Wimmer, H.; Sarker, I.H. An improved transformer-based model for detecting phishing, spam and ham emails: A large language model approach. *Secur. Priv.* **2024**, *7*, e402. [CrossRef]
78. Nguyen, Q.H.; Wu, T.; Nguyen, V.; Yuan, X.; Xue, J.; Rudolph, C. Utilizing Large Language Models with Human Feedback Integration for Generating Dedicated Warning for Phishing Emails. In Proceedings of the 2nd ACM Workshop on Secure and Trustworthy Deep Learning Systems, Singapore, 2–20 July 2024; pp. 35–46.
79. Roy, S.S.; Thota, P.; Naragam, K.V.; Nilizadeh, S. From Chatbots to Phishbots?: Phishing Scam Generation in Commercial Large Language Models. In Proceedings of the 2024 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2024; p. 221.
80. Abbas, N.N.; Ahmed, T.; Shah, S.H.U.; Omar, M.; Park, H.W. Investigating the applications of artificial intelligence in cyber security. *Scientometrics* **2019**, *121*, 1189–1211. [CrossRef]
81. Sajid, M.S.I.; Wei, J.; Alam, M.R.; Aghaei, E.; Al-Shaer, E. Dodgetron: Towards autonomous cyber deception using dynamic hybrid analysis of malware. In Proceedings of the 2020 IEEE Conference on Communications and Network Security (CNS), Avignon, France, 29 June–1 July 2020; pp. 1–9.
82. Hu, J.L.; Ebrahimi, M.; Chen, H. Single-shot black-box adversarial attacks against malware detectors: A causal language model approach. In Proceedings of the 2021 IEEE International Conference on Intelligence and Security Informatics (ISI), San Antonio, TX, USA, 2–3 November 2021; pp. 1–6.
83. Sánchez, P.M.S.; Celdrán, A.H.; Bovet, G.; Pérez, G.M. Transfer Learning in Pre-Trained Large Language Models for Malware Detection Based on System Calls. *arXiv* **2024**, arXiv:2405.09318.
84. Demırcı, D.; Acarturk, C.; şirlancis, M.; şahın, N. Static malware detection using stacked bilstm and gpt-2. *IEEE Access* **2022**, *10*, 58488–58502. [CrossRef]
85. Gao, Y.; Hasegawa, H.; Yamaguchi, Y.; Shimada, H. Malware detection using attributed CFG generated by pre-trained language model with graph isomorphism network. In Proceedings of the 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC), Los Alamitos, CA, USA, 27 June–1 July 2022; pp. 1495–1501.
86. Zahan, N.; Burckhardt, P.; Lysenko, M.; Aboukhadijeh, F.; Williams, L. Shifting the Lens: Detecting Malware in npm Ecosystem with Large Language Models. *arXiv* **2024**, arXiv:2403.12196.
87. Madani, P. Metamorphic malware evolution: The potential and peril of large language models. In Proceedings of the 2023 5th IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), Atlanta, GA, USA, 1–4 November 2023; pp. 74–81.
88. Khan, I.; Kwon, Y.W. A Structural-Semantic Approach Integrating Graph-Based and Large Language Models Representation to Detect Android Malware. In Proceedings of the IFIP International Conference on ICT Systems Security and Privacy Protection, Edinburgh, UK, 12–14 June 2024; Springer: Cham, Switzerland, 2024; pp. 279–293.
89. Fang, C.; Miao, N.; Srivastav, S.; Liu, J.; Zhang, R.; Fang, R.; Tsang, R.; Nazari, N.; Wang, H.; Homayoun, H.; et al. Large Language Models for Code Analysis: Do {LLMs} Really Do Their Job? In Proceedings of the 33rd USENIX Security Symposium (USENIX Security 24), Philadelphia, PA, USA, 14–16 August 2024; pp. 829–846.
90. Houssel, P.R.; Singh, P.; Layeghy, S.; Portmann, M. Towards Explainable Network Intrusion Detection Using Large Language Models. *arXiv* **2024**, arXiv:2408.04342.
91. Zhang, H.; Sediq, A.B.; Afana, A.; Erol-Kantarci, M. Large Language Models in Wireless Application Design: In-Context Learning-enhanced Automatic Network Intrusion Detection. *arXiv* **2024**, arXiv:2405.11002.
92. Alkhatib, N.; Mushtaq, M.; Ghauch, H.; Danger, J.L. Can-bert do it? controller area network intrusion detection system based on bert language model. In Proceedings of the 2022 IEEE/ACS 19th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, 5–8 December 2022; pp. 1–8.
93. Lai, H. Intrusion Detection Technology Based on Large Language Models. In Proceedings of the 2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT), Bengaluru, India, 20–21 October 2023; pp. 1–5.
94. Li, F.; Shen, H.; Mai, J.; Wang, T.; Dai, Y.; Miao, X. Pre-trained language model-enhanced conditional generative adversarial networks for intrusion detection. *Peer- Netw. Appl.* **2024**, *17*, 227–245. [CrossRef]

95. Lin, J.; Guo, Y.; Chen, H. Intrusion Detection at Scale with the Assistance of a Command-line Language Model. *arXiv* **2024**, arXiv:2404.13402.
96. Wang, Z.; Li, J.; Yang, S.; Luo, X.; Li, D.; Mahmoodi, S. A lightweight IoT intrusion detection model based on improved BERT-of-Theseus. *Expert Syst. Appl.* **2024**, *238*, 122045. [CrossRef]
97. Ziems, N.; Liu, G.; Flanagan, J.; Jiang, M. Explaining tree model decisions in natural language for network intrusion detection. *arXiv* **2023**, arXiv:2310.19658.
98. Nguyen, L.G.; Watabe, K. Flow-based network intrusion detection based on BERT masked language model. In Proceedings of the 3rd International CoNEXT Student Workshop, Rome, Italy, 9 December 2022; pp. 7–8.
99. Tran, N.; Chen, H.; Bhuyan, J.; Ding, J. Data curation and quality evaluation for machine learning-based cyber intrusion detection. *IEEE Access* **2022**, *10*, 121900–121923. [CrossRef]
100. Nam, M.; Park, S.; Kim, D.S. Intrusion detection method using bi-directional GPT for in-vehicle controller area networks. *IEEE Access* **2021**, *9*, 124931–124944. [CrossRef]
101. Hu, Y.; Zou, F.; Han, J.; Sun, X.; Wang, Y. Llm-tikg: Threat intelligence knowledge graph construction utilizing large language model. *Comput. Secur.* **2024**, *145*, 103999. [CrossRef]
102. Manocchio, L.D.; Layeghy, S.; Lo, W.W.; Kulatilleke, G.K.; Sarhan, M.; Portmann, M. Flowtransformer: A transformer framework for flow-based network intrusion detection systems. *Expert Syst. Appl.* **2024**, *241*, 122564. [CrossRef]
103. Farrukh, Y.A.; Wali, S.; Khan, I.; Bastian, N.D. XG-NID: Dual-Modality Network Intrusion Detection using a Heterogeneous Graph Neural Network and Large Language Model. *arXiv* **2024**, arXiv:2408.16021.
104. Zhou, X.; Cao, S.; Sun, X.; Lo, D. Large Language Model for Vulnerability Detection and Repair: Literature Review and the Road Ahead. *J. Cybersecur.* **2024**. [CrossRef]
105. Zibaeirad, A.; Vieira, M. VulnLLMEval: A Framework for Evaluating Large Language Models in Software Vulnerability Detection and Patching. *arXiv* **2024**, arXiv:2409.10756.
106. Zhang, L.; Zou, Q.; Singhal, A.; Sun, X.; Liu, P. Evaluating Large Language Models for Real-World Vulnerability Repair in C/C++ Code. In Proceedings of the IWSPA '24: Proceedings of the 10th ACM International Workshop on Security and Privacy Analytics, Porto, Portugal, 12 June 2024; pp. 49–58. [CrossRef]
107. Steenhoek, B.; Rahman, M.M.; Roy, M.K.; Alam, M.S.; Barr, E.T.; Le, W. A Comprehensive Study of the Capabilities of Large Language Models for Vulnerability Detection. *arXiv* **2024**, arXiv:2403.17218.
108. Boi, B.; Esposito, C.; Lee, S. Smart Contract Vulnerability Detection: The Role of Large Language Model (LLM). *ACM SIGAPP Appl. Comput. Rev.* **2024**, *24*, 19–29. [CrossRef]
109. Jensen, R.I.T.; Tawosi, V.; Alamir, S. Software Vulnerability and Functionality Assessment using Large Language Models. In Proceedings of the Third ACM/IEEE International Workshop on NL-Based Software Engineering (NLBSE '24), Lisbon, Portugal, 20 April 2024; pp. 25–28. [CrossRef]
110. Mathews, N.S.; Brus, Y.; Aafer, Y.; Nagappan, M.; McIntosh, S. LLbezpeky: Leveraging Large Language Models for Vulnerability Detection. *arXiv* **2024**, arXiv:2401.01269.
111. Liu, Y.; Gao, L.; Yang, M.; Xie, Y.; Chen, P.; Zhang, X.; Chen, W. VulDetectBench: Evaluating the Deep Capability of Vulnerability Detection with Large Language Models. *arXiv* **2024**, arXiv:2406.07595.
112. Nong, Y.; Yang, H.; Cheng, L.; Hu, H.; Cai, H. Automated Software Vulnerability Patching using Large Language Models. *arXiv* **2024**, arXiv:2408.13597.
113. Zhou, X.; Zhang, T.; Lo, D. Large Language Model for Vulnerability Detection: Emerging Results and Future Directions. In Proceedings of the 2024 ACM/IEEE 44th International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER'24), Lisbon, Portugal, 14–20 April 2024; pp. 47–51. [CrossRef]
114. Nana, S.R.; Bassole, D.; Guel, D.; Sié, O. *Deep Learning and Web Applications Vulnerabilities Detection: An Approach Based on Large Language Models*; Laboratoire de Mathématiques et d'Informatique, Université Joseph KI-ZERBO: Ouagadougou, Burkina Faso, 2023.
115. Hasan, S.M.; Alotaibi, A.M.; Talukder, S.; Shahid, A.R. Distributed Threat Intelligence at the Edge Devices: A Large Language Model-Driven Approach. *arXiv* **2024**, arXiv:2405.08755.
116. 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**, arXiv:2408.03354.
117. Wu, Z.; Tang, F.; Zhao, M.; Li, Y. KGV: Integrating Large Language Models with Knowledge Graphs for Cyber Threat Intelligence Credibility Assessment. *arXiv* **2024**, arXiv:2408.08088.
118. Jo, H.; Lee, Y.; Shin, S. Vulcan: Automatic extraction and analysis of cyber threat intelligence from unstructured text. *Comput. Secur.* **2022**, *120*, 102763. [CrossRef]
119. Liu, J.; Zhan, J. Constructing Knowledge Graph from Cyber Threat Intelligence Using Large Language Model. In Proceedings of the 2023 IEEE International Conference on Big Data (BigData), Sorrento, Italy, 15–18 December 2024.

120. Karuna, P.; Hemberg, E.; O'Reilly, U.M.; Rutar, N. Automating cyber threat hunting using NLP, automated query generation, and genetic perturbation. *arXiv* **2021**, arXiv:2104.11576.
121. Tanksale, V. Cyber Threat Hunting Using Large Language Models. In Proceedings of the International Congress on Information and Communication Technology, London, UK, 19–22 February 2024; Springer: Singapore, 2024; pp. 629–641.
122. Jiang, Y.; Zhang, C.; He, S.; Yang, Z.; Ma, M.; Qin, S.; Kang, Y.; Dang, Y.; Rajmohan, S.; Lin, Q.; et al. Xpert: Empowering Incident Management with Query Recommendations via Large Language Models. In Proceedings of the 2023 International Conference on Incident Management, Lisbon, Portugal, 14–20 April 2023; pp. 1–13.
123. Grigorev, A.; Saleh, K.; Ou, Y.; Mihaita, A.S. Enhancing Traffic Incident Management with Large Language Models: A Hybrid Machine Learning Approach for Severity Classification. In Proceedings of the Proceedings of the 2024 International Conference on Intelligent Transportation Systems (ITSC), Edmonton, AB, Canada, 24–27 September 2024.
124. Chen, Z.; Kang, Y.; Li, L.; Zhang, X.; Zhang, H.; Xu, H.; Zhou, Y.; Yang, L.; Sun, J.; Xu, Z.; et al. Towards Intelligent Incident Management: Why We Need It and How We Make It. In Proceedings of the ACM/IEEE International Conference on Software Engineering (ICSE), Melbourne, VIC, Australia, 14–20 May 2023.
125. Tharayil, S.M.; Alotaibi, N.M.; Idris, M.A.; Aldhalaan, B.H. Combining NLP and Generative Models for Predicting Incident Category and Incident Routing in Incidents Management Systems. In *Information Systems Engineering and Management (ISEM, Volume 5)*; Springer: Cham, Switzerland, 2024.
126. Grigorev, A.; Saleh, A.S.M.K.; Ou, Y. IncidentResponseGPT: Generating traffic incident response plans with generative artificial intelligence. *arXiv* **2024**, arXiv:2404.18550.
127. Tulechki, N. Natural Language Processing of Incident and Accident Reports: Application to Risk Management in Civil Aviation. Ph.D. Thesis, Université Toulouse le Mirail-Toulouse II, Toulouse, France, 2024.
128. Sufi, F. An innovative GPT-based open-source intelligence using historical cyber incident reports. *Nat. Lang. Process. J.* **2024**, *7*, 100074. [CrossRef]
129. Wu, X.; Duan, R.; Ni, J. Unveiling Security, Privacy, and Ethical Concerns of ChatGPT. *J. Inf. Intell.* **2024**, *2*, 102–115. [CrossRef]
130. Kibriya, H.; Khan, W.Z.; Siddiqa, A.; Khan, M.K. Privacy Issues in Large Language Models: A Survey. *J. Inf. Intell.* **2024**, *2*, 102–115. [CrossRef]
131. Plant, R.; Giuffrida, V.; Gkatzia, D. You Are What You Write: Preserving Privacy in the Era of Large Language Models. *arXiv* **2022**, arXiv:2204.09391.
132. Brown, H.; Lee, K.; Mireshghallah, F.; Shokri, R.; Tramèr, F. What Does it Mean for a Language Model to Preserve Privacy? In Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT '22), Seoul, Republic of Korea, 21–24 June 2022; pp. 2280–2292. [CrossRef]
133. Peris, C.; Dupuy, C.; Majmudar, J.; Parikh, R.; Smaili, S.; Zemel, R.; Gupta, R. Privacy in the Time of Language Models. In Proceedings of the WSDM '23: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, Singapore, 27 February–3 March 2023; pp. 1–10.
134. Rodriguez, D.; Yang, I.; Del Alamo, J.M.; Sadeh, N. Large language models: A new approach for privacy policy analysis at scale. *J. Comput.* **2024**, 123–145. [CrossRef]
135. Alsalemi, A.; Al-Kababji, A.; Himeur, Y.; Bensaali, F.; Amira, A. Cloud energy micro-moment data classification: A platform study. In Proceedings of the 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC), Leicester, UK, 7–10 December 2020; pp. 420–425.
136. Long, Z.; Yan, H.; Shen, G.; Zhang, X.; He, H.; Cheng, L. A Transformer-based network intrusion detection approach for cloud security. *J. Cloud Comput.* **2024**, *13*, 5. [CrossRef]
137. Bulut, M.F.; Hwang, J. NL2Vul: Natural Language to Standard Vulnerability Score for Cloud Security Posture Management. In Proceedings of the 2021 IEEE 14th International Conference on Cloud Computing (CLOUD), Chicago, IL, USA, 5–10 September 2021.
138. Chen, Y.; Xie, H.; Ma, M.; Kang, Y.; Gao, X.; Shi, L.; Cao, Y.; Gao, X.; Fan, H.; Wen, M. Automatic Root Cause Analysis via Large Language Models for Cloud Incidents. In Proceedings of the Nineteenth European Conference on Computer Systems (EuroSys '24), Athens, Greece, 22–25 April 2024; pp. 674–688. [CrossRef]
139. Kilhoffer, Z.; Bashir, M. Cloud Privacy Beyond Legal Compliance: An NLP Analysis of Certifiable Privacy and Security Standards. In Proceedings of the 2024 IEEE Cloud Summit, Washington, DC, USA, 27–28 June 2024.
140. Mouratidis, H.; Shei, S.; Delaney, A. A Security Requirements Modelling Language for Cloud Computing Environments. *Softw. Syst. Model.* **2020**, *19*, 271–295. [CrossRef]
141. Baghdasaryan, A.; Bunarjyan, T.; Poghosyan, A.; Harutyunyan, A.; El-Zein, J. Knowledge Retrieval and Diagnostics in Cloud Services with Large Language Models. *Expert Syst. Appl.* **2024**, *255 Pt D*, 124736. [CrossRef]
142. Cao, D.; Jun, W. LLM-CloudSec: Large Language Model Empowered Automatic and Deep Vulnerability Analysis for Intelligent Clouds. In Proceedings of the IEEE INFOCOM 2024—IEEE Conference on Computer Communications, Vancouver, BC, Canada, 20–23 May 2024.

143. Stutz, D.; de Assis, J.T.; Laghari, A.A.; Khan, A.A.; Andreopoulos, N.; Terziev, A.; Deshpande, A.; Kulkarni, D.; Grata, E.G. Enhancing Security in Cloud Computing Using Artificial Intelligence (AI). In *Advancements in Artificial Intelligence and Cloud Computing*; Mahajan, S., Khurana, M., Estrela, V.V., Eds.; Wiley: Hoboken, NJ, USA, 2024; Chapter 11, pp. 179–220. [CrossRef]
144. Song, C.; Ma, L.; Zheng, J.; Liao, J.; Kuang, H.; Yang, L. Audit-LLM: Multi-Agent Collaboration for Log-based Insider Threat Detection. *arXiv* **2024**, arXiv:2408.08902.
145. Xia, S.; Shao, S.; He, M.; Yu, T.; Song, L.; Zhang, Y. AuditGPT: Auditing Smart Contracts with ChatGPT. *arXiv* **2024**, arXiv:2404.12345.
146. Fotoh, L.E.; Mugwira, T. Exploring Large Language Models in External Audits: Implications and Ethical Considerations. *SSRN Preprint* **2024**. [CrossRef]
147. Cartwright, O.; Dunbar, H.; Radcliffe, T. Evaluating Privacy Compliance in Commercial Large Language Models—ChatGPT, Claude, and Gemini. *Preprint* **2024**. [CrossRef]
148. Hassani, S. Enhancing Legal Compliance and Regulation Analysis with Large Language Models. *arXiv* **2024**, arXiv:2404.17522.
149. Zhu, L.; Yang, L.; Li, C.; Hu, S.; Liu, L.; Yin, B. LEGILM: A Fine-Tuned Legal Language Model for Data Compliance. *arXiv* **2024**, arXiv:2409.13721.
150. Chard, S.; Johnson, B.; Lewis, D. Auditing Large Language Models for Privacy Compliance with Specially Crafted Prompts. *OSF Preprint* **2024**. [CrossRef]
151. Sharif, M.; Datta, P.; Riddle, A.; Westfall, K.; Bates, A.; Ganti, V.; Lentzk, M.; Ott, D. DrSec: Flexible Distributed Representations for Efficient Endpoint Security. In Proceedings of the 2024 IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 20–22 May 2024.
152. Motlagh, F.N.; Hajizadeh, M.; Majd, M.; Najafi, P.; Cheng, F.; Meinel, C. Large Language Models in Cybersecurity: State-of-the-Art. *arXiv* **2024**, arXiv:2401.00001.
153. Black, G.S.; Rimal, B.P.; Vaidyan, V.M. Balancing Security and Correctness in Code Generation: An Empirical Study on Commercial Large Language Models. *IEEE Transa. Emerg. Top. Comput.* **2024**, 1–12. [CrossRef]
154. Shao, M.; Chen, B.; Jancheska, S.; Dolan-Gavitt, B.; Garg, S.; Karri, R.; Shafique, M. An Empirical Evaluation of LLMs for Solving Offensive Security Challenges. *arXiv* **2024**, arXiv:2402.11814.
155. Ren, Q.; Gao, C.; Shao, J.; Yan, J.; Tan, X.; Lam, W.; Ma, L. CodeAttack: Revealing Safety Generalization Challenges of Large Language Models via Code Completion. *arXiv* **2024**, arXiv:2403.00002.
156. Cui, T.; Wang, Y.; Fu, C.; Xiao, Y.; Li, S.; Deng, X.; Liu, Y.; Zhang, Q.; Qiu, Z.; Li, P.; et al. Risk Taxonomy, Mitigation, and Assessment Benchmarks of Large Language Model Systems. *arXiv* **2024**, arXiv:2401.00003.
157. Szabó, Z.; Bilicki, V. A New Approach to Web Application Security: Utilizing GPT Language Models for Source Code Inspection. *Future Internet* **2023**, *15*, 326. [CrossRef]
158. Achitouv, I.; Gorduza, D.; Jacquier, A. Natural Language Processing for Financial Regulation. *IEEE Access* **2020**, *8*, 238837–238849. [CrossRef]
159. Alashri, S.; Karbab, E.; Binsalleeh, H.; Debbabi, M. CyberBERT: A Pre-trained Language Model for Cybersecurity. *IEEE Access* **2021**, *9*, 23892–23903.
160. Corporation, T.M. MITRE ATT&CK Framework: A Knowledge Base for Adversary Tactics and Techniques. MITRE Technical Reports. 2020. Available online: https://attack.mitre.org/ (accessed on 21 May 2024).
161. Parliament and of the European Union. *General Data Protection Regulation (GDPR)*; Parliament and of the European Union: Brussels, Belgium, 2016.
162. US Department of Health and Human Services. *Health Insurance Portability and Accountability Act of 1996 (HIPAA)*; US Department of Health and Human Services: Washington, DC, USA, 1996.
163. Wang, W.; Sadjadi, S.M.; Rishe, N. A Survey of Major Cybersecurity Compliance Frameworks. In Proceedings of the IEEE 10th Conference on Big Data Security on Cloud (BigDataSecurity), New York, NY, USA, 10–12 May 2024; pp. 23–34.
164. Yao, X.; Wu, X.; Li, X.; Xu, H.; Li, C.; Huang, P.; Li, S.; Ma, X.; Shan, J. Smart Audit System Empowered by LLM. *arXiv* **2024**, arXiv:2410.07677.
165. Sarker, I.H. Generative AI and Large Language Modeling in Cybersecurity. In *AI-Driven Cybersecurity and Threat Intelligence: Cyber Automation, Intelligent Decision-Making and Explainability*; Springer Nature: Cham, Switzerland, 2024; pp. 79–99.
166. Saha, D.; Tarek, S.; Yahyaei, K.; Saha, S.K.; Zhou, J.; Tehranipoor, M.; Farahmandi, F. LLM for soc security: A paradigm shift. *IEEE Access* **2024**, *12*, 155498–155521. [CrossRef]
167. Jones, A.J. Justo Integration of LLMs with Traditional Security Tools. In *Application of Large Language Models (LLMs) for Software Vulnerability Detection*; IGI Global: Hershey, PA, USA, 2024; pp. 295–328.
168. Johnson, O.; Mozes, M.; He, X.; Kleinberg, B.; Griffin, L.D. Use of llms for illicit purposes: Threats, prevention measures, and vulnerabilities. *arXiv* **2023**, arXiv:2308.12833.

169. Ayoobi, N.; Shahriar, S.; Mukherjee, A. The looming threat of fake and llm-generated linkedin profiles: Challenges and opportunities for detection and prevention. In Proceedings of the 34th ACM Conference on Hypertext and Social Media, Rome, Italy, 4–8 September 2023; pp. 1–10.
170. White, C. Interpretable Insights for Cybersecurity Stakeholders Using LLMs. *AI Soc.* **2021**, *30*, 1205–1220.
171. Hildebrandt, C.; Woodlief, T.; Elbaum, S. ODD-diLLMma: Driving Automation System ODD Compliance Checking using LLMs. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, United Arab Emirates, 13–18 October 2024; pp. 13809–13816.
172. Ranade, P.; Piplai, A.; Joshi, A.; Finin, T. Cybert: Contextualized embeddings for the cybersecurity domain. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), Virtual, 15–18 December 2021; pp. 3334–3342.
173. Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C.H.; Kang, J. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* **2020**, *36*, 1234–1240. [CrossRef] [PubMed]
174. Beltagy, I.; Lo, K.; Cohan, A. SciBERT: A pretrained language model for scientific text. *arXiv* **2019**, arXiv:1903.10676.
175. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv* **2019**, arXiv:1909.11942.
176. Tony, C.; Mutas, M.; Ferreyra, N.E.D.; Scandariato, R. Llmseceval: A dataset of natural language prompts for security evaluations. In Proceedings of the 2023 IEEE/ACM 20th International Conference on Mining Software Repositories (MSR), Melbourne, Australia, 15–16 May 2023; pp. 588–592.
177. Hazell, J. Spear Phishing with Large Language Models. *arXiv* **2023**, arXiv:2305.06972.
178. Fayyazi, R.; Yang, S.J. On the Uses of Large Language Models to Interpret Ambiguous Cyberattack Descriptions. *arXiv* **2023**, arXiv:2306.14062.
179. Seyyar, Y.E.; Yavuz, A.G.; Ünver, H.M. An attack detection framework based on BERT and deep learning. *IEEE Access* **2022**, *10*, 68633–68644. [CrossRef]
180. 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, 20–27 February 2024.
181. Fu, M.; Tantithamthavorn, C.; Le, T.; Nguyen, V.; Phung, D. VulRepair: A T5-Based Automated Software Vulnerability Repair. In Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Singapore, 14–18 November 2022; pp. 935–947.
182. Ferrag, M.A.; Battah, A.; Tihanyi, N.; Debbah, M.; Lestable, T.; Cordeiro, L.C. Securefalcon: The next cyber reasoning system for cyber security. *arXiv* **2023**, arXiv:2307.06616.
183. Koide, T.; Fukushi, N.; Nakano, H.; Chiba, D. Detecting phishing sites using chatgpt. *arXiv* **2023**, arXiv:2306.05816.
184. Gur, I.; Nachum, O.; Miao, Y.; Safdari, M.; Huang, A.; Chowdhery, A.; Narang, S.; Fiedel, N.; Faust, A. Understanding HTML with Large Language Models. *arXiv* **2023**, arXiv:2210.03945.
185. Ali, T.; Kostakos, P. HuntGPT: Integrating Machine Learning-Based Anomaly Detection and Explainable AI with Large Language Models (LLMs). *arXiv* **2023**, arXiv:2309.16021.
186. Greshake, K.; Abdelnabi, S.; Mishra, S.; Endres, C.; Holz, T.; Fritz, M. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security, Copenhagen, Denmark, 26–30 November 2023; pp. 79–90.
187. Zhang, Y.; Ippolito, D. Prompts should not be seen as secrets: Systematically measuring prompt extraction attack success. *arXiv* **2023**, arXiv:2307.06865.
188. Yan, J.; Gupta, V.; Ren, X. Bite: Textual backdoor attacks with iterative trigger injection. *arXiv* **2022**, arXiv:2205.12700.
189. Qi, X.; Huang, K.; Panda, A.; Wang, M.; Mittal, P. Visual adversarial examples jailbreak large language models. *arXiv* **2023**, arXiv:2306.13213. [CrossRef]
190. Bagdasaryan, E.; Hsieh, T.Y.; Nassi, B.; Shmatikov, V. (Ab) using Images and Sounds for Indirect Instruction Injection in Multi-Modal LLMs. *arXiv* **2023**, arXiv:2307.10490.
191. Bailey, L.; Ong, E.; Russell, S.; Emmons, S. Image hijacking: Adversarial images can control generative models at runtime. *arXiv* **2023**, arXiv:2309.00236.
192. Wei, A.; Haghtalab, N.; Steinhardt, J. Jailbroken: How does llm safety training fail? *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 80079–80110.
193. Qiu, H.; Zhang, S.; Li, A.; He, H.; Lan, Z. Latent jailbreak: A benchmark for evaluating text safety and output robustness of large language models. *arXiv* **2023**, arXiv:2307.08487.
194. Shayegani, E.; Mamun, M.A.A.; Fu, Y.; Zaree, P.; Dong, Y.; Abu-Ghazaleh, N. Survey of vulnerabilities in large language models revealed by adversarial attacks. *arXiv* **2023**, arXiv:2310.10844.
195. Carlini, N.; Nasr, M.; Choquette-Choo, C.A.; Jagielski, M.; Gao, I.; Koh, P.W.W.; Ippolito, D.; Tramer, F.; Schmidt, L. Are aligned neural networks adversarially aligned? *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 61478–61500.

196. Zou, A.; Wang, Z.; Kolter, J.Z.; Fredrikson, M. Universal and transferable adversarial attacks on aligned language models. *arXiv* **2023**, arXiv:2307.15043.
197. Cao, B.; Cao, Y.; Lin, L.; Chen, J. Defending against alignment-breaking attacks via robustly aligned llm. *arXiv* **2023**, arXiv:2309.14348.
198. Yu, J.; Lin, X.; Xing, X. Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts. *arXiv* **2023**, arXiv:2309.10253.
199. Liu, X.; Xu, N.; Chen, M.; Xiao, C. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *arXiv* **2023**, arXiv:2310.04451.
200. Zeng, Y.; Wu, Y.; Zhang, X.; Wang, H.; Wu, Q. Autodefense: Multi-agent llm defense against jailbreak attacks. *arXiv* **2024**, arXiv:2403.04783.
201. Helbling, A.; Phute, M.; Hull, M.; Chau, D.H. Llm self defense: By self examination, llms know they are being tricked. *arXiv* **2023**, arXiv:2308.07308.
202. Alon, G.; Kamfonas, M. Detecting language model attacks with perplexity. *arXiv* **2023**, arXiv:2308.14132.
203. Jain, N.; Schwarzschild, A.; Wen, Y.; Somepalli, G.; Kirchenbauer, J.; Chiang, P.y.; Goldblum, M.; Saha, A.; Geiping, J.; Goldstein, T. Baseline defenses for adversarial attacks against aligned language models. *arXiv* **2023**, arXiv:2309.00614.
204. Zhu, S.; Zhang, R.; An, B.; Wu, G.; Barrow, J.; Wang, Z.; Huang, F.; Nenkova, A.; Sun, T. Autodan: Automatic and interpretable adversarial attacks on large language models. *arXiv* **2023**, arXiv:2310.15140.
205. Li, H.; Guo, D.; Fan, W.; Xu, M.; Huang, J.; Meng, F.; Song, Y. Multi-step jailbreaking privacy attacks on chatgpt. *arXiv* **2023**, arXiv:2304.05197.
206. Yuan, Y.; Jiao, W.; Wang, W.; Huang, J.t.; He, P.; Shi, S.; Tu, Z. Gpt-4 is too smart to be safe: Stealthy chat with llms via cipher. *arXiv* **2023**, arXiv:2308.06463.
207. Deng, Y.; Zhang, W.; Pan, S.J.; Bing, L. Multilingual jailbreak challenges in large language models. *arXiv* **2023**, arXiv:2310.06474.
208. Bhatt, M.; Chennabasappa, S.; Li, Y.; Nikolaidis, C.; Song, D.; Wan, S.; Ahmad, F.; Aschermann, C.; Chen, Y.; Kapil, D.; et al. Cyberseceval 2: A wide-ranging cybersecurity evaluation suite for large language models. *arXiv* **2024**, arXiv:2404.13161.
209. Hossen, M.I.; Zhang, J.; Cao, Y.; Hei, X. Assessing Cybersecurity Vulnerabilities in Code Large Language Models. *arXiv* **2024**, arXiv:2404.18567.
210. Tann, W.; Liu, Y.; Sim, J.H.; Seah, C.M.; Chang, E.C. Using large language models for cybersecurity capture-the-flag challenges and certification questions. *arXiv* **2023**, arXiv:2308.10443.
211. Shahid, M.R.; Debar, H. Cvss-bert: Explainable natural language processing to determine the severity of a computer security vulnerability from its description. In Proceedings of the 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), Pasadena, CA, USA, 13–16 December 2021; pp. 1600–1607.
212. Quinn, T.; Thompson, O. Applying Large Language Model (LLM) for Developing Cybersecurity Policies to Counteract Spear Phishing Attacks on Senior Corporate Managers. *Res. Sq.* **2024**. [CrossRef]
213. Kaddour, J.; Harris, J.; Mozes, M.; Bradley, H.; Raileanu, R.; McHardy, R. Challenges and Applications of Large Language Models. *arXiv* **2023**, arXiv:2307.10169v1.
214. Das, B.C.; Amini, M.H. Security and Privacy Challenges of Large Language Models: A Survey. *arXiv* **2024**, arXiv:2402.00888. [CrossRef]
215. Yang, J.; Jin, H.; Tang, R.; Han, X.; Feng, Q.; Jiang, H.; Yin, B.; Hu, X. Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. *arXiv* **2023**, arXiv:2304.13712. [CrossRef]
216. Abdali, S.; Anarfi, R.; Barberan, C.; He, J. Securing Large Language Models: Threats, Vulnerabilities and Responsible Practices. *arXiv* **2024**, arXiv:2403.12503.
217. Kumar, P. Adversarial attacks and defenses for large language models (LLMs): Methods, frameworks & challenges. *Int. J. Multimed. Inf. Retr.* **2024**, *13*, 26.
218. Weidinger, L.; Mellor, J.; Rauh, M.; Griffin, C.; Uesato, J.; Huang, P.-S.; Cheng, M.; Glaese, M.; Balle, B.; Kasirzadeh, A.; et al. Ethical and social risks of harm from language models. *arXiv* **2021**, arXiv:2112.04359.
219. Lee, H.; Hong, S.J.; Park, J.; Kim, T.; Kim, G.; Ha, J.H. KoSBI: A Dataset for Mitigating Social Bias Risks Towards Safer Large Language Model Applications. *arXiv* **2023**, arXiv:2305.17701.
220. ChatGPT Banned in Italy Over Privacy Concerns. 2023. Available online: https://www.bbc.com/news/technology-65139406 (accessed on 1 May 2023).
221. Glukhov, D.; Shumailov, I.; Gal, Y.; Papernot, N.; Papyan, V. LLM Censorship: A Machine Learning Challenge or a Computer Security Problem? *arXiv* **2023**, arXiv:2307.10719.
222. Chen, C.; Feng, X.; Zhou, J.; Yin, J.; Zheng, X. Federated Large Language Model: A Position Paper. *arXiv* **2023**, arXiv:2307.08925.
223. Pan, Y.; Pan, L.; Chen, W.; Nakov, P.; Kan, M.Y.; Wang, W.Y. *On the Risk of Misinformation Pollution with Large Language Models*; National University of Singapore: Singapore, 2023.
224. Kshetri, N. Cybercrime and Privacy Threats of Large Language Models. *IT Prof.* **2023**, *25*, 9–13. [CrossRef]

225. Wang, Y.; Pan, Y.; Yan, M.; Su, Z.; Luan, T.H. A Survey on ChatGPT: AI-Generated Contents, Challenges, and Solutions. *IEEE Trans. Cybern.* **2023**, *4*, 280–302. [CrossRef]
226. Al-kfairy, M.; Mustafa, D.; Kshetri, N.; Insiew, M.; Alfandi, O. Ethical challenges and solutions of generative AI: An interdisciplinary perspective. *Informatics* **2024**, *11*, 58. [CrossRef]
227. Feretzakis, G.; Papaspyridis, K.; Gkoulalas-Divanis, A.; Verykios, V.S. Privacy-Preserving Techniques in Generative AI and Large Language Models: A Narrative Review. *Information* **2024**, *15*, 697. [CrossRef]
228. Coello, C.E.A.; Alimam, M.N.; Kouatly, R. Effectiveness of ChatGPT in coding: A comparative analysis of popular large language models. *Digital* **2024**, *4*, 114–125. [CrossRef]
229. Huang, K.; Li, Y.; Thaine, P. Use GenAI Tools to Boost Your Security Posture. In *Generative AI Security: Theories and Practices*; Springer: Cham, Switzerland, 2024; pp. 305–338.
230. Carlini, N.; Tramer, F.; Wallace, E.; Jagielski, M.; Herbert-Voss, A.; Lee, K.; Roberts, A.; Brown, T.B.; Song, D.; Erlingsson, Ú.; et al. Extracting Training Data from Large Language Models. *arXiv* **2020**, arXiv:2012.07805.
231. Xi, Z.; Du, T.; Li, C.; Pang, R.; Ji, S.; Chen, J.; Ma, F.; Wang, T. Defending pre-trained language models as few-shot learners against backdoor attacks. *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 32748–32764.
232. Guo, P.; Liu, F.; Lin, X.; Zhao, Q.; Zhang, Q. L-autoda: Leveraging large language models for automated decision-based adversarial attacks. *arXiv* **2024**, arXiv:2401.15335.
233. Xue, J.; Zheng, M.; Hua, T.; Shen, Y.; Liu, Y.; Bölöni, L.; Lou, Q. Trojllm: A black-box trojan prompt attack on large language models. *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 65665–65677.
234. Jin, M.; Zhu, S.; Wang, B.; Zhou, Z.; Zhang, C.; Zhang, Y. Attackeval: How to evaluate the effectiveness of jailbreak attacking on large language models. *arXiv* **2024**, arXiv:2401.09002.
235. Yang, H.; Xiang, K.; Ge, M.; Li, H.; Lu, R.; Yu, S. A comprehensive overview of backdoor attacks in large language models within communication networks. *IEEE Netw.* **2024**, *38*, 211–218. [CrossRef]
236. Wang, Y.; Dong, X.; Caverlee, J.; Yu, P.S. DA3: A Distribution-Aware Adversarial Attack against Language Models. *arXiv* **2024**, arXiv:2311.08598.
237. Cui, J.; Xu, Y.; Huang, Z.; Zhou, S.; Jiao, J.; Zhang, J. Recent advances in attack and defense approaches of large language models. *arXiv* **2024**, arXiv:2409.03274.
238. Zhao, W.; Li, Z.; Li, Y.; Zhang, Y.; Sun, J. Defending Large Language Models Against Jailbreak Attacks via Layer-specific Editing. *arXiv* **2024**, arXiv:2405.18166.
239. Ashcroft, C.; Whitaker, K. Evaluation of domain-specific prompt engineering attacks on large language models. *ESS Open Arch. Eprints* **2024**, *362*, 36267312.
240. Wu, T.; Luo, L.; Li, Y.F.; Pan, S.; Vu, T.T.; Haffari, G. Continual Learning for Large Language Models: A Survey. *arXiv* **2023**, arXiv:2301.07082.
241. Ai, L.; Kumarage, T.; Bhattacharjee, A.; Liu, Z.; Hui, Z.; Davinroy, M.; Cook, J.; Cassani, L.; Trapeznikov, K.; Kirchner, M.; et al. Defending Against Social Engineering Attacks in the Age of LLMs. *arXiv* **2024**, arXiv:2406.12263.
242. Schmitt, M.; Flechais, I. Digital Deception: Generative artificial intelligence in social engineering and phishing. *Artif. Intell. Rev.* **2024**, *57*, 1–23. [CrossRef]
243. Hassanin, M.; Moustafa, N. A Comprehensive Overview of Large Language Models (LLMs) for Cyber Defences: Opportunities and Directions. *arXiv* **2024**, arXiv:2405.14487.
244. Waelchli, S.; Walter, Y. Reducing the risk of social engineering attacks using SOAR measures in a real world environment: A case study. *Comput. Secur.* **2025**, *148*, 104137. [CrossRef]
245. Karlzen, H.; Sommestad, T. Automatic incident response solutions: A review of proposed solutions' input and output. In Proceedings of the 18th International Conference on Availability, Reliability and Security, Benevento, Italy, 29 August–1 September 2023; pp. 1–9.
246. Kyaw, P.H.; Gutierrez, J.; Ghobakhlou, A. A Systematic Review of Deep Learning Techniques for Phishing Email Detection. *Electronics* **2024**, *13*, 3823. [CrossRef]
247. Yao, Y.; Duan, J.; Xu, K.; Cai, Y.; Sun, Z.; Zhang, Y. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confid. Comput.* **2024**, *4*, 100211. [CrossRef]
248. Xu, H.; Wang, S.; Li, N.; Wang, K.; Zhao, Y.; Chen, K.; Yu, T.; Liu, Y.; Wang, H. Large language models for cyber security: A systematic literature review. *arXiv* **2024**, arXiv:2405.04760.