PENINSULA PUBLISHING LLC

SHIFRA
Vol. **(2025)**, 2025, **pp**. 81–97
ISSN: 3078-3186

Research Article

Beyond Detection: Large Language Models and Next-Generation Cybersecurity

Aitizaz Ali[1, *], Mohamed C. Ghanem[2]

[1] *Network Security Forensic Group, School of Technology, Asia Pacifc University, Malaysia*

[2] *Cybersecurity Institute, University of Liverpool, Liverpool L69 7ZX, United Kingdom, UK*

ARTICLE INFO

Article History

Received 12 Nov 2024

Revised: 1 Jan 2025
Accepted 1 Feb 2025
Published 15 Feb 2025

Keywords

Large Language Models (LLMs),

Cybersecurity (CS),

Threat Detection (TD),

Autonomous Defense (AD),

Adversarial Simulation (AS)

ABSTRACT

The Integrating Contextual and Adaptive Cyber Defense Systems The accelerating evolution of cyber threats, seen both in terms of their complexity and frequency of attempts, makes it imperative to move away from traditional, reactive defense structures and towards intelligent, adaptive and proactive cyber defense strategies. Large Language Models (LLMs) (such as GPT this-will-be-a-zillionth-of-a-second-type-here or BERT derivatives) represent powerful new capabilities developed for understanding, analyzing, and generating human-like language with contextual depth enabling a new frontier in the land of cyber defense. This survey investigates the influence of LLMs on multiple key technical areas of cybersecurity including software and system security, network security, content moderation, hardware security, and blockchain security. It showcases applications of LLMs in tasks in vulnerability detection and prevention, phishing, malware detection and analysis, and intrusion detection; as well as smart contract auditing and adversarial attack simulation. The paper goes more in-depth on technical constructs like zero-shot and few-shot threat hunting, prompt engineering for scenario modeling, and retrieval-augmented generation for real-time intelligence. Task differences are captured in knowledge retrieval from external sources, enabling more contextual learning and task adaptability, inter-model interaction, and augmentation through domain-specific fine-tuning. While promising, LLMs pose open challenges, such as hallucination, adversarial misuse, generalization problems, and ethical issues around privacy and accountability. The implications of such comparative analysis across domains highlight the strengths as well as 'emerging risk' associated with LLM systems at mission critical environments. The survey ends with a highly visionary section on the future opportunities around future self-healing systems, autonomous cyber agents, and the future of LLMs that can not only detect threats but also predict, simulate, and respond automatically to threats. This work serves as a foundational roadmap for researchers and practitioners for the use of LLMs for the next generation of resilient cybersecurity tools.

1. INTRODUCTION

Large Language Models (LLMs) have been one of the most groundbreaking inventions of AI in recent years, with their applications ranging from natural language understanding and code generation, to information retrieval and human-machine interaction. Models like OpenAI's GPT family, Google's PaLM, Meta's LLaMA, and open-source projects such as Falcon and Mistral have showcased state-of-the-art performance on zero-shot or few-shot learning benchmarks, thus paving the way for automating advanced reasoning and decision-making pipelines with very little supervision [1- 4]. Simultaneously, the cybersecurity landscape is also changing rapidly, with adversaries using more advanced, machine learning-based techniques to find system weaknesses, conceal their presence, and take down digital networks [5,6]. Conventional security methods especially those relying on static rule sets or signature matching are proving inadequate in the face of advanced persistent threats, polymorphic malware, and real-time social engineering attacks. The increasing complexity and scale of cyber threats places a significant demand on intelligent, adaptive, and autonomous defense mechanisms [7].

Combining these LLMs with cybersecurity opens a new path to next-gen defense systems moving beyond passive detection. These models are capable of not only detecting known threats, but also generalizing on new attack signatures, mimicking the behavior of attackers and helping human analysts prepare a proactive approach [8,9]. Going "beyond detection" is a paradigm shift itself and imagines LLM as an agent to actively mitigate threats, repair code, analyze logs, etc. [10]. In this survey, we investigate the dual use of LLMs in cyber security domain and its applications in various topics

*Corresponding author email: aitizaz.ali@apu.edu.my
DOI: https://doi.org/10.70470/SHIFRA/2025/005

like Software security, Network security, Information security, Hardware security, and Blockchain security. Here we present the full taxonomy of cases of use, review contemporary s9235e46uues and limitations, and finally outline an agenda going forward to exploit the power of LLMs in the use of resilient, self-evolving cyber defense architectures in the future [11].

With training up to October 2023, you have to put LLMs in the context of what cybersecurity is today. Traditional cybersecurity methods rely heavily on rule-based systems, static signatures, and handcrafted heuristics that are often narrowly focused, reactive, and hit-or-miss. On the other hand, systems driven by LLM exhibit features like semantic comprehension, predictive analytics, and independent-response systems, representing a paradigm change in the industry. These models help systems handle massive amounts of structured and unstructured data from disparate sources, making them more responsive and proactive against changing threats. Table I Comparative overview of conventional and LLM-enhanced cybersecurity strategies. Transition from traditional to language-driven intelligence for a detailed defense across the threat landscape.

TABLE I. COMPARISON BETWEEN TRADITIONAL AND LLM-POWERED CYBERSECURITY APPROACHES

Traditional Cybersecurity	**LLM-Enhanced Cybersecurity**
Static Rules / Signatures	Adaptive Language Understanding
Reactive Detection	Predictive Analysis
Manual Incident Response	Automated Triage & Mitigation
Limited Natural Language Support	Full Context-Aware Text/Code Handling
Human-Centric Investigation	AI-Assisted & Autonomous Decision-Making

1.1 Terminology and Abbreviations

Abbreviation	**Full Term**
LLM	Large Language Model
AI	Artificial Intelligence
NLP	Natural Language Processing
IDS	Intrusion Detection System
NIDS	Network Intrusion Detection System
TI	Threat Intelligence
SOC	Security Operations Center
SIEM	Security Information and Event Management
SOAR	Security Orchestration, Automation, and Response
RAG	Retrieval-Augmented Generation
GPT	Generative Pretrained Transformer
BERT	Bidirectional Encoder Representations from Transformers
T5	Text-to-Text Transfer Transformer
CVE	Common Vulnerabilities and Exposures
MITRE	MITRE ATT&CK Framework
DeFi	Decentralized Finance
HDL	Hardware Description Language
GNN	Graph Neural Network
API	Application Programming Interface
IoT	Internet of Things
SoC	System on Chip
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
XAI	Explainable Artificial Intelligence
AD	Adversarial Detection
TD	Threat Detection

2. FOUNDATIONS OF LARGE LANGUAGE MODELS (LLMS)

LLMs (Large Language Models) have emerged as one of the most significant advancements in artificial intelligencee, equippingn machines the ability to comprehend, interpret and produce human-like language with unprecedented fluency. Such models are the technical backbone of promising new cybersecurity applications designed to offer more than just threat detection. Understanding their architectural design and training paradigms will be key to determining their future effectiveness in integrating with next-generation cybersecurity systems.

2.1 LLM Architectures

Most LLMs are based on the Transformer architecture, which introduced attention mechanisms and parallel processing that transformed the field of natural language processing. Depending on the intended use case and capabilities, LLMs are usually classified into three architectural paradigms: encoder-only, decoder-only, and encoder-decoder.

1. Language understanding models, also called encoder-only models, include BERT (Bidirectional Encoder Representations from Transformers). Contextual embeddings from input sequences are generated, which show great efficacy in log file analysis, policy classification, and anomaly detection. The capacity of these models to capture bidirectional context makes them capable of identifying subtle threats in structured security data [12].
2. Decoder-only models can be used for language generation (such as the GPT series). These models have proven themselves powerful at predictive and generative tasks, like predicting phishing and generating alerts narratives or threat behaviors in an incident report. Because they have an autoregressive structure, they are able to produce coherent, context-sensitive textual outputs [13].
3. Encoder-decoder models: models such as T5 (Text-to-Text Transfer Transformer) and BART (Bidirectional and Auto-Regressive Transformer) combining the good sides of both architectures. In these tasks, where understanding and generation perform together (examples may comprise malware code decompilation, multilingual threat description, and auto incident logging), these models have proven to be most effective. That versatility makes them useful components of security automation pipelines [14]. Table II provides a comparative summary of each of the various architecture articles that illustrate the distinctive features and cybersecurity relevance of each, respectively.

TABLE II. COMPARISON OF LLM ARCHITECTURES IN CYBERSECURITY CONTEXT

Architecture Type	Examples	Model Structure	Strengths	Cybersecurity Applications
Encoder-only	BERT, RoBERTa	Processes input to embeddings	Strong contextual understanding; suitable for classification	Log analysis, anomaly detection, access control rule parsing
Decoder-only	GPT-2, GPT-3	Generates output sequentially	Powerful generation capabilities; contextual prediction	Phishing simulation, alert generation, threat scenario modeling
Encoder-decoder	T5, BART	Combines encoder and decoder	Bidirectional understanding and generation in a unified model	Malware translation, incident report generation, multilingual summarization

To examine the different underlying structure of the various Large Language Models, we break down the three key network types Encoder-Only, Decoder-Only and Encoder-Decoder into their original structure as shown in Figure 1. These skills, used together, allow them to perform a specific set of jobs in cybersecurity including threat classification, threat classification, phishing simulation and generation of security reports. The diagram lays out the flow of input data through each architecture and what the nature of their outputs is to create a framework for examining their application in next-generation cybersecurity solutions.

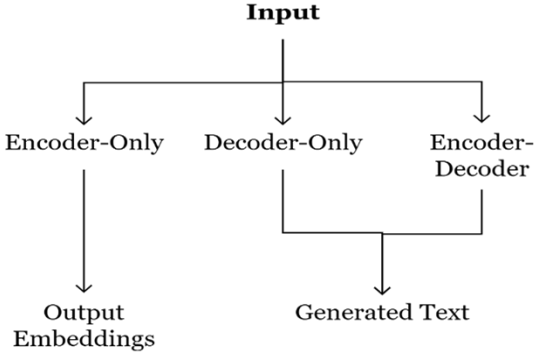

Fig. 1. Overview of Transformer-based LLM architectures showing Encoder-Only (e.g., BERT), Decoder-Only (e.g., GPT), and Encoder-Decoder (e.g., T5, BART) pathways from input to output.

2.2 Training Mechanisms

The trainer mechanisms of LLMs are primarily responsible for their exceptional capabilities. These mechanisms dictate how LLMs learn these language features and utilize them in specific areas such as cybersecurity. At a high level, three key strategies have been employed in the design and adaptation of LLMs: pretraining, fine-tuning, and few/zero-shot learning. Pretraining constitutes the initial stage wherein LLMs undergo training on extensive and varied text corpora through unsupervised or self-supervised learning objectives. Examples of such techniques are masked language modelling (as in BERT) and autoregressive prediction (as in GPT-based models). This phase allows the model to learn the syntactic, semantic, and contextual representations of language over a wide variety of topics and domains so that it gains general linguistic intelligence. Pretraining over such a wide corpus means that LLMs are capable of detecting patterns across logs, threat descriptions, technical documentation, and communication data [15].

Fine-tuning is a supervised learning method applied after pretraining. During this state, the pretrained model is fine-tuned on smaller, task-specific datasets that account for individual downstream tasks. Fine-tuning has been used in cybersecurity to improve the performance of LLMs on threat intelligence classification, phishing detection, malware code analysis, and vulnerability description generation. Moreover, the fine-tuned models exhibit stronger precision and recall on domain aware tasks which make them vital for practical cybersecurity deployment [16].

A more recent novelty concerning LLM utility involves few-shot and zero-shot learning. In few-shot learning, the model is exposed to only a few examples to carry out a new task whereas in zero-shot learning the model taps into its pretrained knowledge to accomplish tasks for which it sees no examples (this can refer to task-specific examples). These techniques are especially useful in cybersecurity situations with little and/or expensive-to-acquire tagged training data. An example of these types of attack models could be: a zero-shot model that detects suspicious behavior from textual patterns that the model has never been trained on and a few-shot models that can quickly learn new attack vectors from very few examples [17].

In summary, these training methodologies enable LLMs to engage seamlessly in highly dynamic and sensitive contexts, where contextual awareness and adaptability are paramount. Convolutional networks not only improve generalization from low-data, but also elevate them beyond the rigidity of traditional machine learning, leaving the doors open for their use in next generation cybersecurity suites. As shown in Table III, Comparison of Key Training Mechanisms for Large Language Models. It explains the differences in purpose, data needs, and cybersecurity applications for both approaches.

TABLE III. COMPARATIVE OVERVIEW OF LLM TRAINING MECHANISMS IN CYBERSECURITY

Training Mechanism	Description	Data Requirement	Use in Cybersecurity
Pretraining	Unsupervised learning on large corpora to build general linguistic understanding.	Massive general-domain text datasets	Foundational knowledge for analyzing logs, emails, and documentation.
Fine-tuning	Task-specific supervised learning applied to pretrained models.	Medium-sized labeled datasets	Adapting models for phishing detection, malware classification, vulnerability summarization.
Few-shot Learning	Learns from a few labeled examples at inference time.	Minimal labeled examples	Detecting novel threats with minimal training data.
Zero-shot Learning	Performs new tasks without direct task-specific training.	None (relies on pretraining)	Identifying anomalous behavior or rare attack types without prior examples.

2.3 Open-Source vs. Closed-Source Models in Security Contexts

The rubber meets the road between open-source vs closed-source LLMs Large Language Models (LLMs) can be broadly categorized into two types in terms of their availability and licensing open-source and closed-source. This distinction is directly relevant for their deployment in cybersecurity systems, especially those in need of transparency, data sovereignty or integration flexibility.

Such open-source models (still) provide the advantages of total transparency, customization (at a local level), and the ability to deploy in air-gapped or privacy-sensitive infrastructures. These models can be security-vulnerability audited, customized to certain domains such as intrusion detection, and operated on-premises in comprehensive to maintain data confidentiality. This community-driven approach enables rapid innovation and adaptation, which is appealing in an environment of research and government use cases. However, compared to commercial models, open-source models tend to be behind in raw performance, generalization, and multilingual or very complex tasks support.

On the other hand, proprietary models like OpenAI's GPT-4, Anthropic's Claude and Google's Gemini offer much better performance, access to massive pretraining datasets and integration with enterprise-level tools. They are often trained on huge amounts of data and fine-tuned on specific tasks such as text classification or named entity recognition, resulting in state-of-the-art performance for a variety of NLP tasks. Yet, their black-box nature raises significant concerns with respect to data confidentiality, control, vendor lock-in, and cost especially in security-critical domains such as defense, financial, and industrial systems. There is also a risk of sharing sensitive threat intelligence or private logs with proprietary third-party services [18], and organizations shall also consider that when making their decisions.

Awareness of the differences with open- and closed-source LLMs allows the executive team in cybersecurity to select models that fit operational, regulatory, and risk management needs. And as LLMs transition from merely being detection engines to becoming stakeholders in making cybersecurity decisions, these architectural and licensing decisions deciding their direction will matter more and more. To vividly highlight the differences between open-source and closed-source LLMs, Table IV summarizes their comparative features in cybersecurity domains.

TABLE IV. COMPARISON OF OPEN-SOURCE VS. CLOSED-SOURCE LLMS FOR CYBERSECURITY

Attribute	Open-Source LLMs	Closed-Source LLMs
Examples	LLaMA, Falcon, MPT, BLOOM	GPT-4 (OpenAI), Claude (Anthropic), Gemini (Google)
Transparency	Fully transparent, auditable	Opaque (black-box models)
Customization	Highly customizable	Limited to API-level configuration
Deployment	On-premise, edge-compatible	Cloud-based (some hybrid options emerging)
Performance	Moderate, domain-specific	High, general-purpose
Security & Privacy	Data stays local; lower privacy risk	Data sent to third parties; potential exposure
Cost	Free or low cost (compute-dependent)	Subscription or pay-per-use
Suitability	Government, research, high-trust environments	Enterprise automation, SaaS integration

3. CYBERSECURITY IN THE AGE OF LLMS

The influx of Large Language Models (LLMs) is sparking a revolution in cyber security, bringing with it a broader perspective on security, beyond conventional threat detection-based approaches to intelligent, proactive and adaptive cyber security. From predictive defense and automated analysis to real-time response to emerging threats, these sophisticated models are being leveraged across the cybersecurity landscape. This section breaks down core cyber security domains and discusses how LLMs are being utilized within each.

1. Software and System Security: LLMs are used to audit software codebases, identify vulnerabilities, and even automatically create patches. These tools have contextual understanding, allowing them to spot logical flaws and dangerous coding constructs that most static analysis tools would miss. Pretrained models adapted for code repositories (e.g., CodeBERT) enhance threat modeling, detect API misuse and automate the documentation of security weaknesses [19].
2. Network Security: In the case of Network security, LLMs can provide advanced threat modelling by modelling various traffic and being able to detect real-time abnormalities. In contrast to rule-based intrusion detection systems (IDS), LLM-powered IDS are able to identify zero-day attacks and sophisticated evasion techniques based on contextual interpretation of network logs and behavior [20]. Decoder-based models can also acknowledge attacks to simulate phishing and social engineering for defense training.
3. Information and Content Protection: LLMs are good at classifying, filtering, and getting summaries of sensitive content. They can redact sensitive information from documents, detect attempts to exfiltrate data, and monitor communication channels for insider attacks, all automatically. Due to their proficiency in human language, they are particularly well positioned to help detect malign influence embedded in natural language, such as spear-phishing emails or social media inauthentic activity [21].
4. Hardware Security: While not yet a mature application for LLMs, some early solutions are arising in the area of textual log analysis associated with firmware and BIOS systems helping to secure hardware. LLMs assist in identifying unauthorized device behavior or tampering by interpreting system error reports and sensor data. They are also being integrated into IoT platforms to assist in anomaly detection for low-resource embedded devices [22].
5. Blockchain and Smart Contract Security: In decentralized systems, LLMs can aid in the auditing of smart contracts, harnessing their potential understanding and verification of logic patterns present in Solidity code. These models further improve formal verification tools, as they are capable of identifying logical inconsistencies, reentrancy vulnerabilities, or gas optimization issues. Additionally, LLMs can describe attack vectors in simple terms; this can act as a translator between technical and non-technical stakeholders [23].

In order to understand the ever-changing dynamic of cybersecurity we first need to establish its main domains as well as how they are evolving into use cases utilizing Large Language Models (LLMs). Historically, cybersecurity has been largely dependent on rule-based systems and predefined signatures for threat detection and mitigation. However, these techniques are not well suited against zero-day attacks, advanced malware, and adaptive attacks. The model can identify trends and patterns from these streams of data which would make it possible to recognize novel attacks earlier and take countermeasures before the events of an incident actually happen. This is why they are excellent models for wrapping unstructured data as well as for learning from contextual patterns for real-time anomaly detection, adaptive access control, and automated incident response. Figure 2 presents a detailed taxonomy of the domains of cybersecurity and shows how the utilization LLMs impact each domain (spanning from software security and hardware security to blockchain and

network security). This transition allows for a more informed, adaptive approach to cyber defense, adhering to the demands of more intricate digital ecosystems.

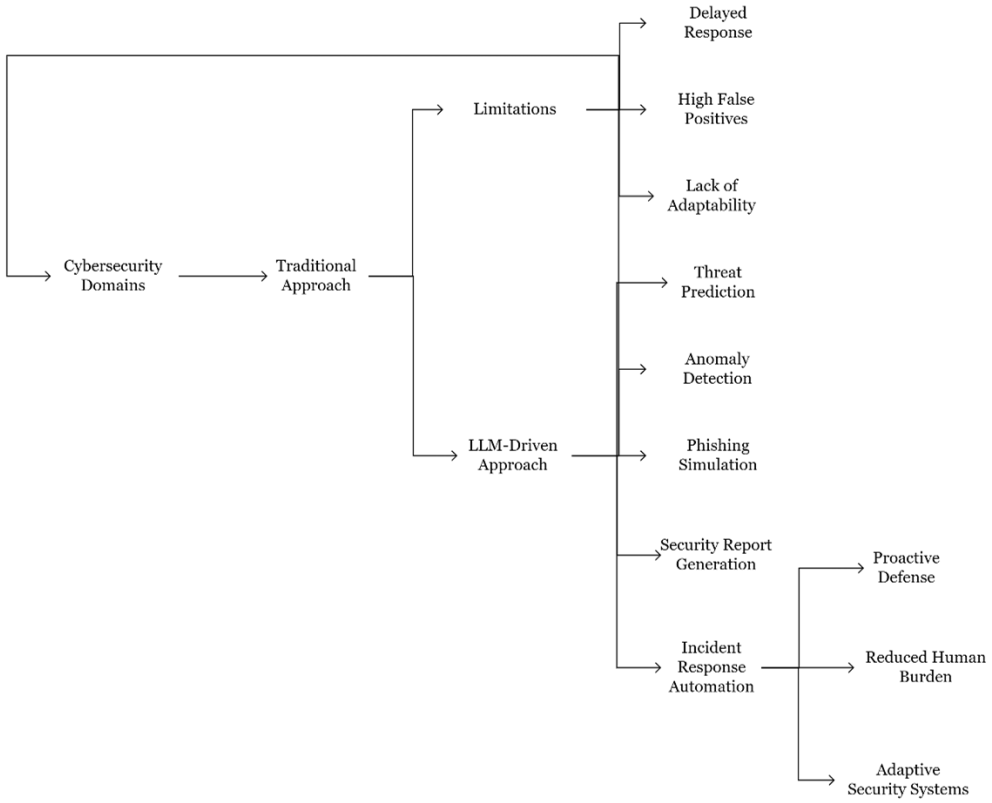

Fig. 2. Taxonomy and transformation of cybersecurity domains through the integration of LLMs, transitioning from traditional detection mechanisms to intelligent, adaptive, and automated security responses.

As LLMs infiltrate the core domains of cybersecurity, the transition from passive defense to intelligent, predictive security systems becomes increasingly tangible. Their capacity to autonomously learn, reason, and generate human-like language and code offers a foundation for real-time, context-aware security systems that go beyond detection.

4. LLMS ACROSS CYBERSECURITY DOMAINS

Large Language Models (LLMs) have begun transforming the cybersecurity landscape by extending their applicability across various specialized domains. Their ability to understand, generate, and contextualize human-like language allows for the automation of complex security tasks, enhancing both preventive and reactive defense mechanisms. This section explores key cybersecurity subdomains where LLMs demonstrate significant utility, illustrating their capabilities in both academic and practical contexts.

4.1 Software & System Security

Large Language Models (LLMs) are revolutionizing software and system security with their potential for intelligent automation in vulnerability detection, malware analysis, and program verification. LLMs with their deep contextual understandings are outpacing older static tools in finding security gaps and giving reasonable recommendations for remediation. This has been used most effectively [...] in finding and fixing vulnerabilities and bugs. Pretrained code-focus models, for example, CodeBERT, Codex, and GPT-Neo can analyze source code, detect known vulnerabilities patterns, and produce or even auto-generate secure code patches [24,25]. They help minimize the manual effort in secure software development lifecycles (SDLC) and allow for quick response to emerging threats. For malware analysis and log mining, LLMs can derive semantic meaning from the natural language description of binary behaviors or system logs. This makes it possible to cluster malware families by behavioral similarities, log parsing, and derive the incident narrative [26]. They are suitable for raw text conversion such as LLMs to turn massive-scale cybersecurity logs into meaningful threat intelligence since logs are a key aspect of cyber threat analysis.

LLMs are also being used in program fuzzing and binary analysis. They create a range of smart test inputs that maximize code coverage during fuzz testing, effectively revealing bugs. Moreover, they play a role in reverse engineering by providing summaries of disassembled binary code, annotating function names, and identifying undocumented or

suspicious behavior in code [27]. The different ways LLMs are used in software and system security are illustrated in Figure 3. By providing context-aware and automated solutions that ease the manual burden while expanding threat coverage, LLMs present several opportunities in this space, from code vulnerability detection to malware log interpretation and intelligent fuzz testing.

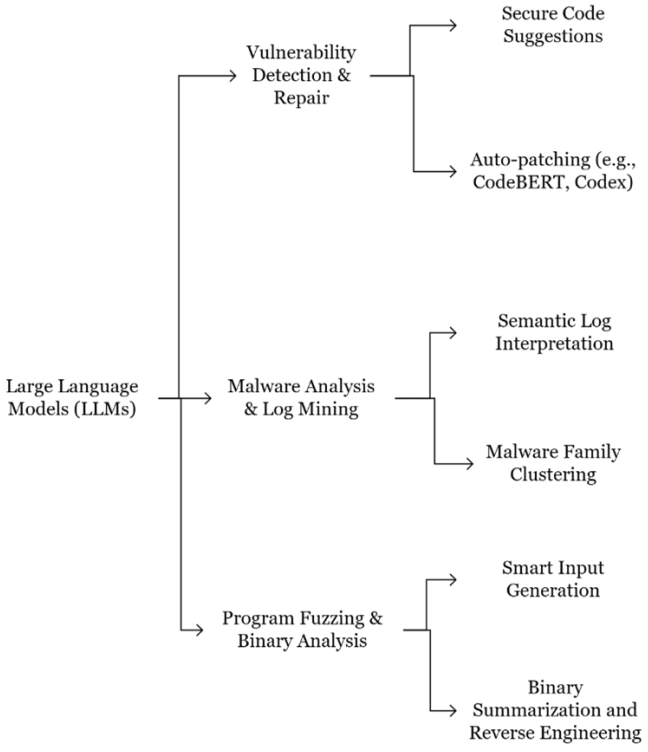

Fig. 3. Applications of Large Language Models in software and system security, illustrating how LLMs enhance code vulnerability repair, malware behavior analysis, and binary fuzzing tasks.

4.2 Network Security

Incorporating a degree of linguistic and contextual awareness never before found in automated threat defense systems, Large Language Models (LLM) are transforming network security. Through the analysis of both structured and unstructured data across multiple layers of the network stack, LLMs can predict and detect anomalies, understand new attack vectors over time, and support security teams in proactive defense initiatives.

1. IDS/NIDS (Intrusion Detection): LLMs have shown considerable improvements over existing signature-based and rule-based IDS systems. Conventional ID Systems usually depend on signatures that are defined in advance or static heuristics, rendering them powerless against zero-day attacks and polymorphic malware. On the other hand, LLMs like GPT or BERT-based classifiers learn traffic behaviors and semantic patterns over time windows, thus allowing them to identify subtle deviations such as the encrypted anomalies or protocol misuse observed in the network traffic [28]. This adaptability also allows them to be deployed in hybrid IDS/NIDS infrastructures, where LLMs can address eventual detection of suspicious behavior while traditional systems focus on known threats.
2. Threat Intelligence and Anomaly Detection: LLMs read and parse unstructured data from threat intelligence feeds, blogs, security forums, and even dark web content, working to automate IOC, TTP extraction. The output can be threat summaries (summarized in an alerting protocol), trends over adversarial behavior patterns, and be ingested by SIEM (Security Information and Event Management) platforms for correlation at real-time [29]. Because they are not easily fooled by ambiguity or multilingual data sources, they are useful for global cybersecurity monitoring and advanced early warning systems.
3. Automation of Penetration Testing: Penetration testing has long been a manual process executed by red teams, yet many of these tasks can now be automated at least to some extent with the use of large language models (LLMs). These models mimic tactics, techniques, and procedures (TTPs) of attackers by creating customized attack scripts, exploiting known vulnerabilities or generating phishing templates. Even more, they analyze results from pen-testing exercises

and provide recommendations on remediation strategies or priority patches significantly reducing time to redress security assessment [30].

This is interesting because LLMs can be leveraged to build a better mental model of networks enabling it to phase out towards a protection plan. Whereas existing systems rely on established (but often out-of-date) signatures or inflexible heuristics, LLMs have the ability to learn the behaviors of traffic adaptively in order to catch abnormal behavior ahead of time and determine whether or not to classify this as an emerging threat. And if that was not enough, they see application in other domains such as intrusion detection, generated threat intelligence, and even in red teaming simulations. As figure 4 suggests, LLMs build on the traditional architecture ensuring core network security elements through contextual interpretation, intelligent, automated and reusable learning capabilities.

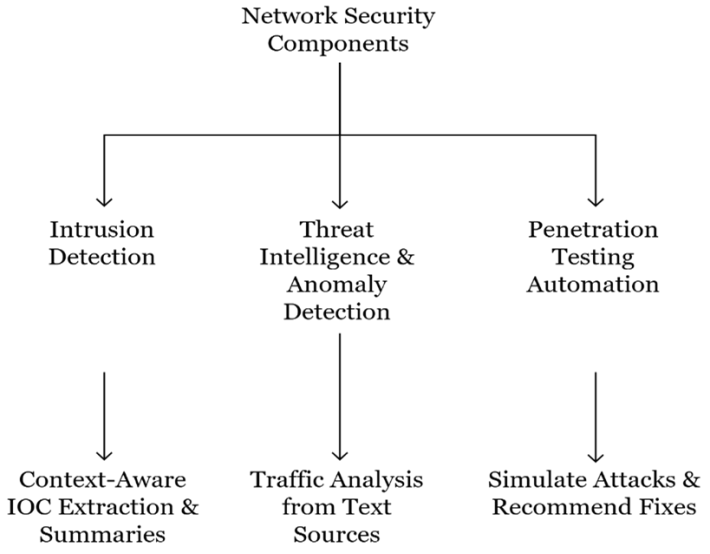

Fig. 4. LLMs in Network Security: Enhancing intrusion detection, threat intelligence, and penetration testing through contextual traffic analysis, automated IOC extraction, and intelligent simulation of attack behavior.

4.3 Information & Content Security

Given that the majority of cybersecurity challenges derive from human-centered communications channels, LLMs have become essential tools for improving the security of information and content. A solid command of language understanding and contextual processing also allows greater accuracy in spotting phishing attacks, moderating malicious content, and performing forensic analysis.

The prominent use case of LLMs is the detection of phishing and scams. Unlike traditional systems using static pattern-matching or blacklisted keywords, LLMs like GPT-4 or fine-tuned versions of BERT (like RoBERTa) can understand linguistic nuances in our writing, social engineering cues, or impersonation of corporate language. The ability of LLMs to perform holistic assessments of email headers, message body content, and sender behavior enables the identification of more advanced phishing emails that pass through traditional filters [31].

LLMs are revolutionizing the way organizations moderate harmful content in communication channels. These models are great at understanding context and can catch the subtlety of toxic language, hate speech or harassment that a simple keyword-based approach would miss. For enterprise, LLMs are being used more and more as a way to enforce the company's policies on communication and have become practically essential in highly regulated environments. This helps a lot in maintaining the integrity and compliance in workplaces as they can identify sarcasm from offensive statements [32]. LLMs also assist digital forensics and access control, providing safety analysts with capabilities for complex log navigation, event correlation, and abnormal user behavior detection [14]. They even parse natural language compliance documents, which makes dynamic policy-driven access decision enforcement a breeze. As a result, it allows real-time response to and semi-automated forensic workflows [33].

Collectively, these applications show the emerging use-cases for LLMs beyond standard security offerings. Bridging unstructured human words and structured cybersecurity reasoning, LLMs reveal threats based on deception, misinformation, or context manipulation. These key contributions span the gamut from phishing detection (which exploits social media functionality) to forensic analysis of phishing attacks (the social media self-service is its almost explosive success) and further illustrate the utility and power of the LLM platform within modern content-centric security paradigm (see Figure 5).

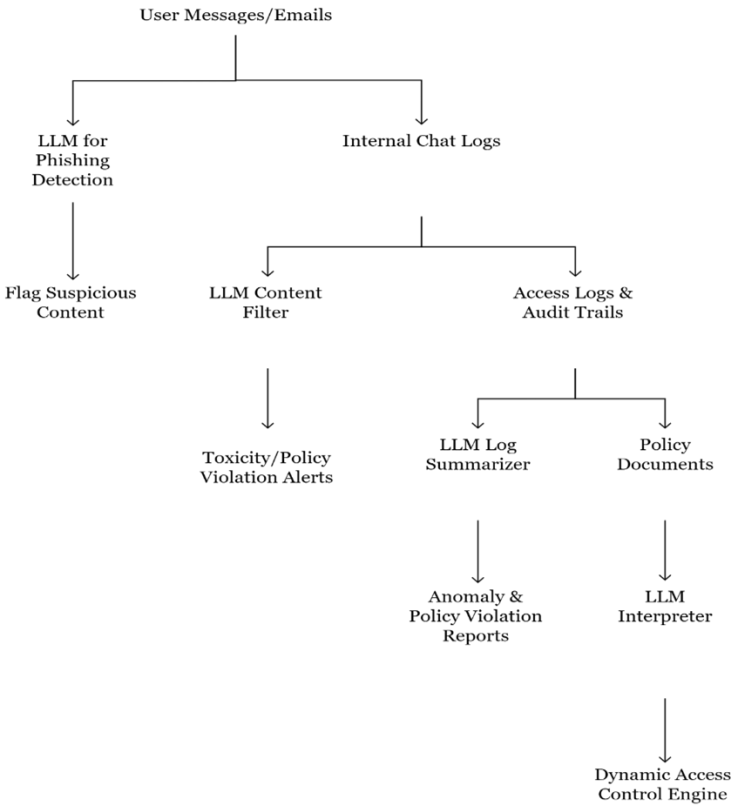

Fig. 5. Key applications of LLMs in information and content security, including phishing detection, harmful content moderation, and forensic analysis for adaptive access control.

4.4 Hardware Security

With the increasing exploitation of low-level systems, LLMs are stepping into the realm of hardware security. While still in its infancy, this nascent discipline has particular relevance to embedded systems and the IoT (Internet of Things), where conventional defenses may run into exactly limited processing power and memory. At the same time, LLMs introduce powerful code understanding and generation capabilities, which open new venues for detecting, mitigating, and even preventing hardware-level vulnerabilities.

One of the most promising applications is vulnerability detection and repair in the firmware and System-on-Chip (SoC) environments. Firmware, which runs below the level of the operating system, has traditionally been a challenge to analyze since it is often opaque and specific to platform. Properly trained LLMs, particularly when fine-tuned on assembly language, hardware logs, and binary structures, can learn to recognize firmware behavior and identify nonexistent instruction sequences or misconfigurations or rootkits. For example, an LLM can analyze bootloader code dumps [38] and flag potentially malicious instructions such as ones associated with persistence mechanisms, like Write Once Read Many (WORM) blocks, or memory remapping routines typically used in firmware-level attacks [34].

Similarly, LLMs can be applied to secure generation of hardware code, a topic of increasing significance in system design automation. LLMs, through prompt-based programming, can translate high-level specifications to syntactically correct and secure Hardware Description Language (HDL) such as Verilog or VHDL code. Hardware engineers can use so-called formal specification tools to check design logic with low human error and integrate security constraints (access control, fault tolerance, encryption modules) from the very first design step. Therefore, not only does LLM-assisted hardware development speed up prototype creation, but it also integrates proactive security principles into the very fabric of computing devices [35].

This evolution in LLM usage will have a tremendous impact on IoT ecosystems in which thousands of heterogeneous, and often insecure, devices operate in critical infrastructures, smart cities, and health care systems. As attackers increasingly targeting the firmware layer, the role of LLMs in defending that attack layer is becoming crucial. And with their capabilities in contextual code analysis & predictive repair, these LLMs can compress the firmware security audit lifecycle and offer rapid path to remediation. As depicted in Figure 6, LLMs currently have two roles in the hardware security landscape. On the one hand, they monitor firmware and SoC logs to look for suspicious instruction sequences and configuration anomalies, helping identify potential firmware-level threats in their early stages. On one hand, they help

foster secure HDL code generation from high-level requirements, thus ensuring that security constraints are considered and respected during the design stages. These features are particularly beneficial for embedded systems and resource-constrained IoT devices, which often have limited capabilities for traditional security monitoring.

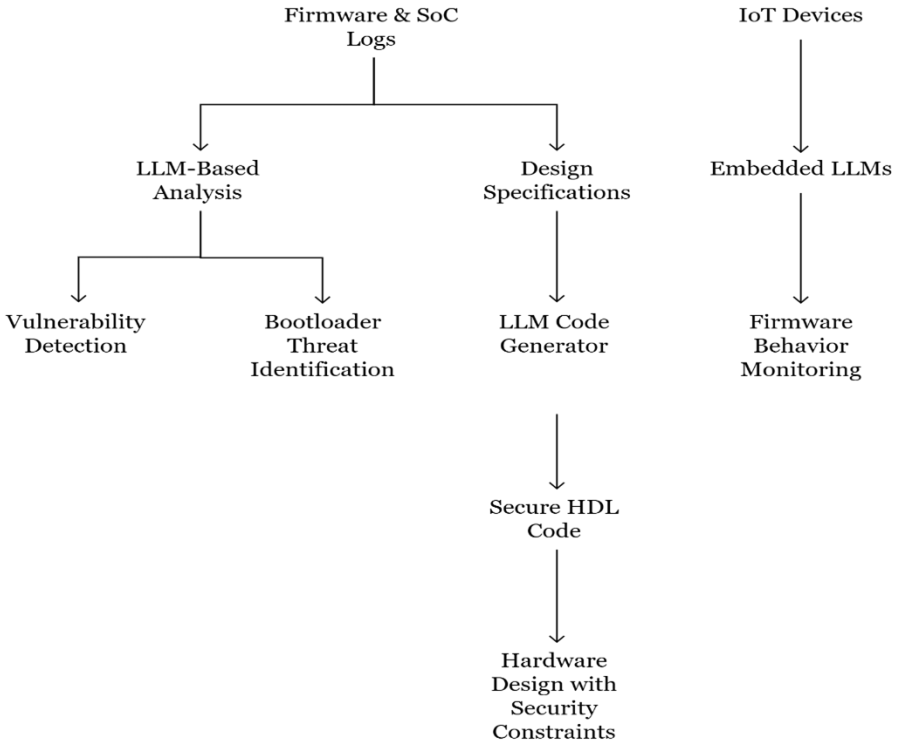

Fig. 6. Applications of LLMs in hardware security, including firmware vulnerability detection via SoC log analysis, and secure HDL code generation from design specifications to ensure safe hardware architecture in embedded and IoT systems.

4.5 Blockchain Security

While blockchain technologies have been heralded as revolutionizing digital trust and decentralization, they in fact also create new vectors of vulnerability, most notably smart contracts and transaction integrity. In this context, we introduce LLMs as powerful allies in the fight to secure decentralized systems. By being able to understand the semantics of the purpose behind the code and the unstructured data in the blockchain, they are able to automate the vulnerabilities analysis and fraud detection and making the auditing user friendly.

Detecting vulnerabilities in smart contracts is one of the most important fields in blockchain security. LLMs trained on Solidity, the dominant language for smart contracts, can recognize well-known security flaws, including reentrancy attacks, integer overflows/underflows, and access control violations. Traditional static analysis tools leverage known patterns in code to flag issues, but this results in a static set of flags and fixes; LLMs, on the other hand, excel at generalizing even risky logic from different codebases, enabling them to not only catch moments of weak logic, but to surface strong logic instead. They are also used for translating contract code to natural-language for human-readable audits or allowing non-technical stakeholders to verify the behavior of contracts [36].

Simultaneously, transaction modeling enhanced anomaly detection and has emerged as a hot topic in decentralized finance (DeFi). LLMs can also ingest and correlate on-chain metadata, including timestamps, gas fees, wallet histories, and token transfers, which can help uncover suspicious behaviors like front-running, wash trading, or sudden liquidity drains. LLMs help in producing contextual insights on new attack patterns that slide under the radar of rule-based systems [37] via unsupervised learning or fine-tuning. Such a feature is particularly useful for both compliance surveillance and real-time DeFi risk mitigation.

To more effectively envision how LLMs can contribute to securing the blockchain infrastructure, we provide a layered architecture of the encompassing features (Figure 7), from a high-level description that abstracts critical attributes that will be matched into possible roles for LLMs and experts. At the outermost layer, different analysis methods are employed on blockchain artifacts like transactions streams and smart contracts. At the heart of this system lies the LLM, which serves as a sophisticated engine to interpret, analyze and secure this data. The knowledge of interpreting smart contract logic, identifying aberrations, and articulating risk vectors leads to a more secure and transparent blockchain ecosystem.

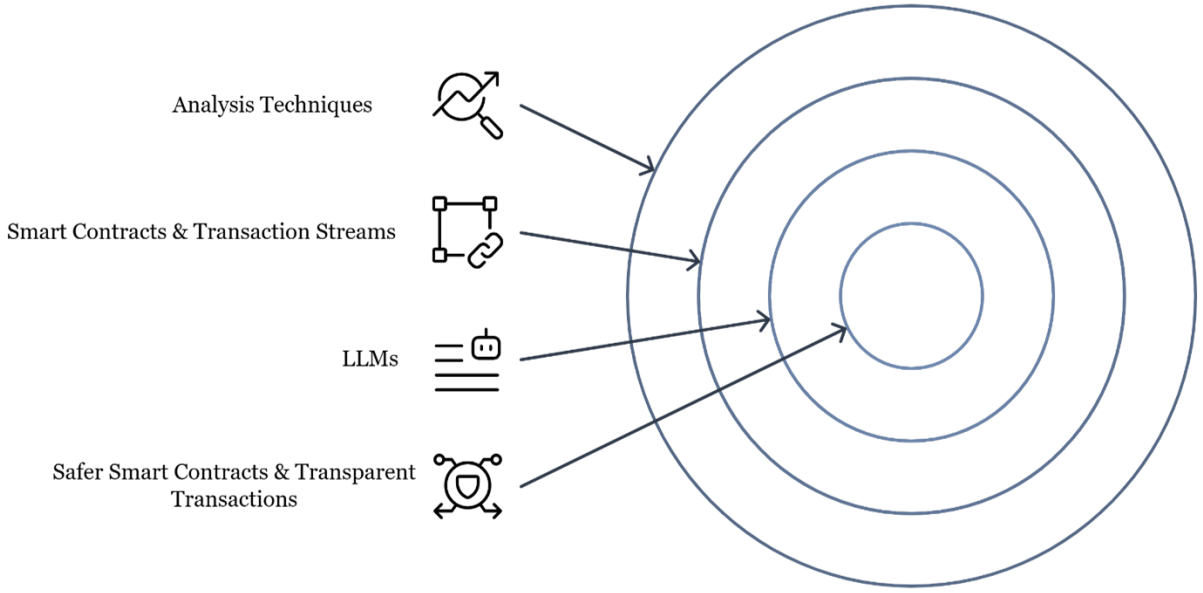

Fig. 7. Layered architecture showing how LLMs contribute to blockchain security through advanced analysis of smart contracts and transaction patterns, leading to improved safety and transparency in decentralized systems.

In the previous sections we dived into the extensive use of LLMs around prominent areas in the cybersecurity industry, we can see that LLMs are so much more than just a detection mechanism. Moreover, although current deployments are mainly focused on threat detection and assistance to analysts, the real transformative potential of the LLMs appears to be related to proactive defense, `live' decision-making and autonomous security. This lays the groundwork for the next generation of cybersecurity the one where LLMs not only serve as detection tools but as intelligent agents that autonomously predict, prevent and respond to cyber threats.

5. BEYOND DETECTION: TOWARD PROACTIVE AND AUTONOMOUS DEFENSE

With the continuous evolution of Large Language Models (LLMs), the use of Juggling Cybersecurity is changing from traditional detection and division functions. Now, however, increasingly those models are being uses as proactive agents that predict threats, run attack vectors simulations, understand patterns, and drive autonomous decision making. This is a major paradigm shift from passive signature-based systems to intelligent and self-adaptive cyber security systems that can learn, infer, and act in real time. Now, one of the most critical enablers of such a big change is prompt engineering, which helps in directing LLMs with good multiple queries to carry out complex tasks, such as threat modeling and red-team scenario generation. Analysts can use contextual prompts to identify possible breach paths, highlight system weaknesses, and simulate actions an adversary might take. These capabilities turn LLMs into cognitive assistants that can also do strategic reasoning. For instance, they can be prompts prompting an LLM to consider the security posture of the given network topology [38] or providing it with some known configurations and asking to suggest attack vectors against these configurations.

Moreover, few-shot and zero-shot learning approaches allow LLMs to adapt to new contexts using little or no training data. This is especially true in exception of disguised where tagged datasets are rare or history. Without such heavy retraining, LLMs can scan through user-behavior logs, recognize suspicious patterns on the command line, and detect underlying anomalies in network flows that are happening live. This enables the new potential for proactive threat hunting, especially for capturing emerging attack trends and zero-day exploits that evade traditional intrusion detection systems [39,40]. Besides passive analysis, LLMs can engage in adversarial simulations by mimicking attacker activity. They can also produce realistic phishing emails, malware, and SQL injection scripts that reflect real-world threats. Red teams can automate huge chunks of their penetration testing process with this capability, while defenders can utilize the same models to explore the generated attacks and improve their defensive modalities. This bi-directional simulation allows the mechanism for continuous adversarial training and cyber resilience enhancement [41].

As synthesized in Table V, LLMs contributes in various crucial aspects of proactive and autonomous cybersecurity strategies. These include hypothesis generation via prompting, few-shot anomaly detection and black-box simulation of adversarial behavior for robust defense testing. Collectively, these functionalities suggest that LLMs are moving from detection drivers to decision and response agents, laying the groundwork for an enlightened era of self-adaptive security ecosystems.

TABLE V. KEY ROLES OF LLMS IN PROACTIVE AND AUTONOMOUS CYBER DEFENSE

Capability	Description	LLM Role	Representative Use Cases
Prompt-Based Threat Modeling	Using structured prompts to identify vulnerabilities and simulate breach paths	Cognitive Reasoning Agent	Red teaming, secure architecture review
Zero-/Few-Shot Threat Hunting	Detecting anomalies with minimal labeled data	Generalization from Sparse Data	Insider threat detection, zero-day exploit recognition
Adversarial Simulation	Generating phishing emails, malware, and attack payloads for testing purposes	Adaptive Attacker/Defender	Penetration testing, social engineering scenario generation
Autonomous Feedback Loops	Self-learning through continuous simulation and policy refinement	Dual-role Interaction Model	Auto-hardening environments, dynamic security posture updating

6. ENHANCING LLMS FOR CYBERSECURITY TASKS

While LLMs offer remarkable natural language capabilities, their direct application to cybersecurity often requires targeted enhancements to meet the domain's complexity and specificity. This section explores practical strategies to adapt and extend LLM functionality through external augmentation and task-specific fine-tuning. These approaches help bridge the gap between general language understanding and the nuanced demands of cyber threat detection, analysis, and response.

6.1 External Augmentation Techniques

With LLMs capable of raw results generation, that may also not suit cyber needs, augmentation can help hone in the capabilities of LLMs to better suit the specialized needs in cyber. The external augmentation techniques enable LLMs to engage with structured knowledge repositories, adjust to security-centric queries, and work with other AI agents, providing meaningful responses to intricate threat environments. By doing this 1st Feature Augmentation helps to enrich model input by adding structured metadata, like IP logs, timestamps, protocol layers or user access patterns, in the natural language prompt. Such a hybrid input format allows LLMs to reason over linguistic as well as technical cues to achieve more accurate threat interpretation and action [41]. 2nd Knowledge Retrieval frameworks are frameworks that couple an LLM with an external security knowledge base (e.g., MITRE ATT&CK, CVE repositories), which allow models to pull real-time threat intelligence during inference. This is known as retrieval-augmented generation (RAG) which helps the models stay relevant without frequent fine-tuning [42]. 3rd: Inter-model Interaction LLMs are paired to symbolic reasoning engines, GNNs, or rule-based expert systems to check and refine outputs An LLM could propose a mitigation, for instance, and a formal verification engine could be used to assert that it is realizable in a given system architecture [43]. 4th Task-Adaptive TrainingFine-tuning LLMs on cybersecurity-specific corpora, including vulnerability databases, malware analysis reports, or intrusion detection logs. This adaptation allows the model to map general language understanding with technical specificity, improving overall classification and prediction performance in cybersecurity domains [44].

6.2 Challenges of Fine-Tuning for Security Tasks

While these advancements offer great promise, the adaptation of LLMs for cybersecurity also presents particular challenges. One of the greatest issues is overfitting, stemming from the small size and low diversity of labelled cybersecurity datasets. High-quality security data can often be scarce, sensitive, or proprietary, especially when it comes to attack traces, malicious payloads, or insider threat logs [45]. One major limitation of fine-tuning existing models is the challenge it presents for general readable language capabilities through the risk of catastrophic forgetting through splitting the training data sets into domain-specific narrow data. This issue becomes especially pronounced when maintaining the line between language fluency and technical correctness in threat interpretation [46].

And adversarial robustness is still a big problem. When provided with adversarial queries or poisoned samples, LLMs can generate fake-money or false outputs. In cybersecurity applications, such failures could lead to missed detections or erroneous and insecure recommendations [47]. The moral side is also not to be ignored. The approach does create privacy issues and compliance challenges, however, as training on real-world threat intelligence could expose sensitive information about an organization. Additionally, there are dual-use dilemmas involved with deploying LLMs that can simulate attacks or generate malware that need careful governance [48].

The augmentation methods beyond traditional fine tuning help the researchers and practitioners to achieve the best performance of the LLMs in the cybersecurity applications. These externally augmentations from structured feature placement to external knowledge retrieval enable LLMs to make more contextual and domain-specific decision. Additionally, recent work with collaborative architectures in which LLMs are embedded among other models or rules-based engines have shown potential to overcome limitations related to reliability and interpretability. Although task-adaptive training is critical to high-risk security environments, it is limited by dependent on the amount of available labeled data, which is often scarce. Table VI summarizes these augmentation strategies and relevant cybersecurity use cases, elucidating how they work synergistically to enhance the objectives of LLMs from generic language processors to domain-specific cybersecurity agents.

TABLE VI. KEY AUGMENTATION STRATEGIES FOR IMPROVING LLM PERFORMANCE IN CYBERSECURITY APPLICATIONS

Technique	Description	Cybersecurity Application
Feature Augmentation	Incorporating structured security metadata with natural language prompts	Enabling contextual log analysis and behavior correlation
Knowledge Retrieval	Accessing external sources like MITRE ATT&CK or CVE during inference	Real-time threat intelligence injection and contextual reasoning
Inter-model Interaction	Collaborating with rule-based engines or GNNs to validate LLM output	Refining LLM decisions for anomaly detection and access policies
Task-Adaptive Training	Fine-tuning on cybersecurity-specific datasets and technical corpora	Enhancing precision in malware detection, vulnerability analysis

7. OPPORTUNITIES AND FUTURE DIRECTIONS

As LLMs continue to mature, their ability to transform the cybersecurity landscape goes far beyond detection and response. So the future applications predict, more distributed, resilient and real-time context-aware systems, where the LLMs will be the backbone of all digital infrastructure security." LLM-powered analysis and decision making could lend itself to one such transformative direction: self-healing cybersecurity systems architectures that can autonomously monitor, analyze and remediate vulnerabilities. Such advanced systems utilize natural language reasoning to map symptoms to root causes, facilitating automatic execution of mitigation actions, without human intervention, and bringing threat response times down to the order of minutes in critical environments [49]

A more interesting frontier in this space is explainable and interpretable LLMs in the context of security operations. Powerful as they are, modern models are quite literally black boxes, a hindrance in high-risk areas where explainability matters. New initiatives seek to construct LLMs capable of justifying their predictions in natural language, offering step-by-step reasoning, and summarizing their decisions in human-readable forms. Such cheap availability of capabilities can increase trust among cybersecurity analysts while automatically aligning these capabilities with compliance requirements, which is particularly relevant in regulated sectors [50]. Moreover, LLMs are enabling autonomous agents and co-pilots in cyber intelligent assistants that can help defenders with a wide range of tasks. These agents could oversee threat feeds, triage alerts, propose mitigation others, and even mimic attack chains. Encompassing both cloud and on-premise environments, they promise the scaling of human expertise, serving as intelligent intermediaries between multiple sources of data and decision-makers [51].

The integration of LLMs with existing cybersecurity infrastructures (SIEM; SOAR; edge computing) will bring forth new capabilities. Embedding LLMs at the edge enables real-time decision-making in latency-sensitive environments, including industrial IoT or autonomous systems. Combining them with SIEM and SOAR platforms allows automating triage workflows, accelerating the incident response, and enriching alerts with contextual information derived from large volumes of unstructured corpora [52].

Taken together, these trends point toward an increasingly intelligent, adaptive, and self-governing cybersecurity ecosystem. LLMs have the potential to transform the very nature of security operationalization, shifting the paradigm from reactive defense to proactive resilience through components of natural language understanding, autonomous reasoning, and real-time interaction. Researchers and practitioners are examining multiple forward-looking applications to leverage the transformative potential of Large Language Models (LLMs) in cybersecurity. This involves building self-healing systems that can autonomously remediate security incidents, developing explainable models to address the issue of transparency, as well as integrating LLMs into real-time operational settings such as security information and event management (SIEM), security orchestration, automation, and response (SOAR), and edge computing environments. Emerging opportunities and their forecasted impact on cybersecurity operations are shown in Table VII.

TABLE VII. EMERGING OPPORTUNITIES AND FUTURE DIRECTIONS FOR LLMS IN CYBERSECURITY

Opportunity	Description	Impact in Cybersecurity
Self-Healing Systems	LLMs autonomously diagnose and remediate system faults or vulnerabilities	Enables faster recovery, reduces human workload, and increases system resilience
Explainable and Interpretable LLMs	Models provide reasoning and human-readable justifications for outputs	Enhances trust, improves compliance, and supports auditability in security operations
Autonomous Cyber Agents & Co-Pilots	Intelligent assistants that support security analysts in real-time	Accelerates threat analysis, prioritization, and response with contextual recommendations
Integration with SIEM, SOAR & Edge	LLMs embedded in operational security tools and edge devices	Boosts real-time alert enrichment, automation, and low-latency decision-making

8. CHALLENGES AND OPEN ISSUES

As LLMs proliferate within cybersecurity systems, many of these same technical and ethical challenges will need to be resolved before these models can realistically be trusted in real-time mission-critical defense strategies. While they do

provide remarkable capabilities in understanding natural language and generating intelligent responses, LLMs can be subject to domain overfitting, adversarial manipulation, and legal ambiguity. These issues run deeper than merely technical shortcomings; they touch on issues of trust, interpretability, accountability, and compliance. This section enlists the major open issues that need to be resolved by researchers, developers, and policymakers, to ensure a safe and effective deployment of LLMs in cybersecurity-related contexts.

1. Generalization and Robustness Across Domains: LLMs generally show diminished performance when faced with tasks that diverge from their training distribution. This is problematic for cybersecurity work as the building blocks of information in the form of system logs, network packets, firmware traces and all things binary do not resemble traditional natural language text. With improper domain adaptation techniques, models become insensitive to fine-grained anomalies and it also generates numerous false-positives, which makes detection performance less effective and less credible. Improving robustness in diverse operational configurations is still a major research area.
2. Trustworthiness, Hallucination, and Adversarial Usage: LLMs can hallucinate produce statistically plausible but incorrect or misleading outputs. Such errors in security-critical environments can result in defective diagnoses, missed real-world attacks, or spread of inaccurate threat intelligence. Furthermore, adversaries can construct malicious by-products (e.g., adversarial prompts) that will either drive model outputs, produce harmful contents, or avoid detection. This makes it crucial to verify the output, and improve prompt resilience before deploying to the real-world.
3. Ethics, legality and accountability As LLMs are more broadly employed for automated decision-making e.g. where to flag an account for abuse, deny access, or prioritize alerts it's unclear how accountability and compliance will work. Who is responsible if a model sends a false positive or wrongly identifies a threat? Furthermore, fine-tuning using sensitive threat intelligence data may violate privacy regulations (e.g., GDPR), especially if models learn to memorize or regenerate sensitive information. Will help address the dual-use challenge of LLMs their potential as both a protective or offensive tool for systems that is key for ethical AI governance.

In this context, Table 8 highlights the most significant limitations and ethical challenges in the real-world implementation of LLMs in cybersecurity and the potential impact on security operations as a result. This tabular view serves to enhance awareness of the multi-dimensional risks which can include technical deficiencies (e.g., hallucination), legal issues, and regulatory dilemmas to be aggressively mitigated to ensure the safe and responsible adoption of LLMs in security-critical environments.

TABLE VIII. SUMMARY OF CHALLENGES AND OPEN ISSUES IN LLM-DRIVEN CYBERSECURITY

Challenge	Description	Implications
Generalization & Robustness	Difficulty adapting to unseen or domain-specific security data	Lower detection accuracy, increased false negatives
Hallucination & Adversarial Prompts	Susceptibility to generating incorrect or manipulated outputs	Misinformation, attack vector exploitation
Trust & Interpretability	Opaque model reasoning and black-box decision-making	Reduced analyst confidence, lack of transparency
Ethical & Legal Risks	Privacy violations, consent issues, dual-use misuse	Regulatory non-compliance, liability, reputational and ethical concerns

9. COMPARATIVE ANALYSIS AND DISCUSSION

Large Language Models (LLMs) have seen various applications in cybersecurity across software and networks to content moderation, hardware systems, and blockchain platforms. While these models possess several strengths around language understanding, code understanding, and zero-shot learning, they also bring along many operational and security considerations. This section distills a comparative overview of LLM adoption across the major cybersecurity domains and considers the strengths and weaknesses of existing approaches.

LLMs excel at working with unstructured threat data, providing context to logs, behaviors, and attack patterns in ways that classical rule-based systems simply cannot. As an example, using models such as GPT and similar to BERT can detect code-level vulnerabilities [53], automate penetration testing [54] and summarize forensic logs [55]. However, there are still major drawbacks namely, hallucination of incorrect outputs, susceptibility to adversarial prompts, and lack of grounding in domain knowledge [56, 57]. To provide a sense of the landscape of LLM applications, Table IX includes a comparative summary of typical use cases, strengths, and remaining challenges across five core cybersecurity domains.

TABLE IX. COMPARATIVE OVERVIEW OF LLM APPLICATIONS ACROSS CYBERSECURITY DOMAINS

Domain	Key Applications	Strengths	Limitations
Software & System Security	Vulnerability detection, patch generation, malware log mining	Deep code comprehension, automation of secure fixes	Limited in binary and legacy code understanding
Network Security	Intrusion detection, anomaly spotting, attack simulation	Handles real-time flow, adapts to new threats	Prone to adversarial evasion; difficulty with encrypted traffic

Information & Content Security	Phishing detection, moderation, access pattern analysis	Linguistic precision, effective against social engineering	May misclassify nuanced or context-dependent text
Hardware Security	Firmware analysis, secure HDL generation	Enables secure-by-design development	Lacks pretrained knowledge on low-level hardware data
Blockchain Security	Smart contract auditing, DeFi anomaly detection	Understands Solidity, detects logic bugs and exploits	Struggles with large-scale transactional behavior analysis

Despite these limitations, the trajectory of LLM evolution points toward their increasing autonomy and integration with traditional security infrastructure. Hybrid systems combining LLMs with symbolic engines, knowledge graphs, or formal verification tools may offer enhanced trustworthiness and interpretability in mission-critical environments [58-60].

10. CONCLUSION

This survey focused on investigating the impact of Large Language Models (LLMs) on cybersecurity, enabling an evolution from traditional detection approaches to proactive, intelligent and autonomous cyber-defense strategies. LLMs have shown great promise in a number of areas including vulnerability discovery in source code, network intrusion detection, phishing detection and defense, malware behavior modeling, contract auditing, and adversarial simulation, due to their sophisticated language understanding and context modeling capabilities. Architectures such as encoder-only (BERT, etc.), decoder-only (GPT, etc.), and encoder-decoder (T5, etc.) have allowed for their integration into cyber technical solutions by way of analysis/generation tasks. The threat landscape is always in flux, but LLMs have exhibited great efficacy in zero-shot and few-shot learning, which enables security teams to act on novel attack vectors with minimal historical data. In addition, methods like prompt engineering, retrieval-augmented generation, inter-model collaboration, and task-adaptive training have augmented their capabilities in specialized security areas. Nevertheless, still challenges including domain generalization, adversarial manipulation, hallucination, and ethics namely data privacy, explainability, and dual-use concerns remain prominent hurdles to their widespread implementation. These limitations reinforce the need to construct LLM systems that are robust, interpretable, and accountable for working in high-stake environments in the field of cybersecurity. Going forward, the path highlights advancements towards self-healing essences, security-savvy autonomous agents, and a closer tether between LLMs and SIEM, SOAR, and edge compute frameworks. In an era where LLMs are rapidly evolving, treating them not just as intelligent assistants, but also as self-directed cyber defenders, will reimagine the very framework of security operations. In the end, the responsible and collaborative development of LLM-empowered security frameworks will be the foundation for adaptive and resilient cyber ecosystems that can counter the new ideals of sophistication being directed towards our systems.

Funding:

The study was conducted independently without any monetary support from third-party organizations or sponsors. The authors declare that no external funding was provided for this research.

Conflicts of Interest:

The authors declare no competing interests.

Acknowledgment:

The authors would like to thank their institutions for their moral support and provision of critical resources throughout the research process.

References

[1] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," *arXiv preprint arXiv:1312.6026*, 2013.
[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[3] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks," in *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, pp. 1597–1600, 2017.
[4] A. Vaswani *et al*., "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
[5] B. Yan *et al*., "On protecting the data privacy of large language models (LLMs): A survey," *arXiv preprint arXiv:2403.05156*, 2024.
[6] D. Myers *et al*., "Foundation and large language models: Fundamentals, challenges, opportunities, and social impacts," *Cluster Comput.*, vol. 27, no. 1, pp. 1–26, 2024.
[7] S. Tonmoy *et al*., "A comprehensive survey of hallucination mitigation techniques in large language models," *arXiv preprint arXiv:2401.01313*, 2024.
[8] M. A. Ferrag, M. Debbah, and M. Al-Hawawreh, "Generative AI for cyber threat-hunting in 6G-enabled IoT networks," in *Proc. IEEE/ACM 23rd Int. Symp. Cluster, Cloud and Internet Comput. Workshops (CCGridW)*, pp. 16–25, 2023.
[9] I. H. Sarker *et al*., "Multiaspect rule-based AI: Methods, taxonomy, challenges and directions toward automation, intelligence and transparent cybersecurity modeling for critical infrastructures," *Internet of Things*, p. 101110, 2024.
[10] Y. Yao *et al*., "A survey on large language model (LLM) security and privacy: The good, the bad, and the ugly," *High-Confidence Comput.*, p. 100211, 2024.

[11] Y. Yan, Y. Zhang, and K. Huang, "Depending on yourself when you should: Mentoring LLM with RL agents to become the master in cybersecurity games," *arXiv preprint arXiv:2403.17674*, 2024.
[12] M. Sladić *et al*., "LLM in the shell: Generative honeypots," *arXiv preprint arXiv:2309.00155*, 2023.
[13] W. Tann *et al*., "Using large language models for cybersecurity capture-the-flag challenges and certification questions," *arXiv preprint arXiv:2308.10443*, 2023.
[14] O. G. Lira, A. Marroquin, and M. A. To, "Harnessing the advanced capabilities of LLM for adaptive intrusion detection systems," in *Int. Conf. Adv. Inf. Netw. Appl.*, Springer, pp. 453–464, 2024.
[15] C. Ebert and M. Beck, "Artificial intelligence for cybersecurity," *IEEE Softw.*, vol. 40, no. 6, pp. 27–34, 2023.
[16] J. Wang *et al*., "Software testing with large language models: Survey, landscape, and vision," *IEEE Trans. Softw. Eng.*, 2024.
[17] E. Almazrouei *et al*., "The Falcon series of open language models," *arXiv preprint arXiv:2311.16867*, 2023.
[18] H. Zhou *et al*., "Large language model (LLM) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities," 2024.
[19] H. Lai and M. Nissim, "A survey on automatic generation of figurative language: From rule-based systems to large language models," *ACM Comput. Surv.*, 2024.
[20] M. A. Ferrag *et al*., "Revolutionizing cyber threat detection with large language models: A privacy-preserving BERT-based lightweight model for IoT/IIoT devices," *IEEE Access*, 2024.
[21] N. Tihanyi *et al*., "Dynamic intelligence assessment: Benchmarking LLMs on the road to AGI with a focus on model confidence," *arXiv preprint arXiv:2410.15490*, 2024.
[22] N. Tihanyi *et al*., "Cybermetric: A benchmark dataset based on retrieval-augmented generation for evaluating LLMs in cybersecurity knowledge," in *Proc. IEEE Int. Conf. Cyber Secur. Resilience (CSR)*, pp. 296–302, 2024.
[23] Z. Liu, "A review of advancements and applications of pre-trained language models in cybersecurity," in *Proc. 12th Int. Symp. Digit. Forensics Secur. (ISDFS)*, pp. 1–10, 2024.
[24] O. Friha *et al*., "LLM-based edge intelligence: A comprehensive survey on architectures, applications, security and trustworthiness," *IEEE Open J. Commun. Soc.*, 2024.
[25] S. Jamal, H. Wimmer, and I. H. Sarker, "An improved transformer-based model for detecting phishing, spam and ham emails: A large language model approach," *Security Privacy*, p. e402, 2024. [Online]. Available: https://doi.org/10.1002/spy2.402
[26] W. X. Zhao *et al*., "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.
[27] F. R. Alzaabi and A. Mehmood, "A review of recent advances, challenges, and opportunities in malicious insider threat detection using machine learning methods," *IEEE Access*, vol. 12, pp. 30907–30927, 2024.
[28] M. A. K. Raiaan *et al*., "A review on large language models: Architectures, applications, taxonomies, open issues and challenges," *IEEE Access*, vol. 12, pp. 26839–26874, 2024.
[29] R. Fang *et al*., "LLM agents can autonomously exploit one-day vulnerabilities," *arXiv preprint arXiv:2404.08144*, 2024.
[30] Y. Chang *et al*., "A survey on evaluation of large language models," *ACM Trans. Intell. Syst. Technol.*, 2023.
[31] D. Saha *et al*., "LLM for SoC security: A paradigm shift," *arXiv preprint arXiv:2310.06046*, 2023.
[32] B. Min *et al*., "Recent advances in natural language processing via large pre-trained language models: A survey," *ACM Comput. Surv.*, vol. 56, no. 2, pp. 1–40, 2023.
[33] S. Zhang *et al*., "Instruction tuning for large language models: A survey," *arXiv preprint arXiv:2308.10792*, 2023.
[34] A. Fan *et al*., "Large language models for software engineering: Survey and open problems," *arXiv preprint arXiv:2310.03533*, 2023.
[35] J. Wu *et al*., "Multimodal large language models: A survey," *arXiv preprint arXiv:2311.13165*, 2023.
[36] Y. Liu *et al*., "Trustworthy LLMs: A survey and guideline for evaluating large language models' alignment," *arXiv preprint arXiv:2308.05374*, 2023.
[37] L. Hu *et al*., "A survey of knowledge enhanced pre-trained language models," *IEEE Trans. Knowl. Data Eng.*, 2023.
[38] Z. A. Abbood, D. Ç. Atilla, and Ç. Aydin, "Intrusion detection system through deep learning in routing MANET networks," *Intell. Autom. Soft Comput.*, vol. 37, no. 1, pp. 269–281, 2023.
[39] Z. A. Abbood, N. A. F. Abbas, and B. Makki, "Spectrum sensing utilizing power threshold and artificial intelligence in cognitive radio," *Int. J. Robot. Control Syst.*, vol. 2, no. 4, pp. 628–637, 2022.
[40] Y. Yigit *et al*., "Critical infrastructure protection: Generative AI, challenges, and opportunities," *arXiv preprint arXiv:2405.04874*, 2024.
[41] J. Wang *et al*., "Software testing with large language models: Survey, landscape, and vision," *IEEE Trans. Softw. Eng.*, pp. 1–27, 2024.
[42] H. Xu *et al*., "Large language models for cyber security: A systematic literature review," *arXiv preprint arXiv:2405.04760*, 2024.
[43] Z. Han *et al*., "Parameter-efficient finetuning for large models: A comprehensive survey," *arXiv preprint arXiv:2403.14608*, 2024.
[44] J. Zhang *et al*., "When LLMs meet cybersecurity: A systematic literature review," *arXiv preprint arXiv:2405.03644*, 2024.
[45] C. Cui *et al*., "A survey on multimodal large language models for autonomous driving," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.* pp. 958–979, 2024,.
[46] G. Bai *et al*., "Beyond efficiency: A systematic survey of resource-efficient large language models," *arXiv preprint arXiv:2401.00625*, 2024.
[47] S. Tian *et al*., "Opportunities and challenges for ChatGPT and large language models in biomedicine and health," *Brief. Bioinform.*, vol. 25, no. 1, p. bbad493, 2024.
[48] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
[49] S. M. Kasongo, "A deep learning technique for intrusion detection system using a recurrent neural networks based framework," *Comput. Commun.*, vol. 199, pp. 113–125, 2023.
[50] S. M. Sohi, J.-P. Seifert, and F. Ganji, "RNNIDS: Enhancing network intrusion detection systems through deep learning," *Comput. Secur.*, vol. 102, p. 102151, 2021.
[51] K. Cho *et al*., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[52] H. Sedjelmaci *et al*., "Cyber security based on artificial intelligence for cyber-physical systems," *IEEE Netw.*, vol. 34, no. 3, pp. 6–7, 2020.
[53] P. Dixit and S. Silakari, "Deep learning algorithms for cybersecurity applications: A technological and status review," *Comput. Sci. Rev.*, vol. 39, p. 100317, 2021.
[54] S. Gaba *et al*., "A systematic analysis of enhancing cybersecurity using deep learning for cyber physical systems," *IEEE Access*, 2024.
[55] C. Yin *et al*., "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.
[56] D. Güera and E. J. Delp, "Deepfake video detection using recurrent neural networks," in *Proc. 15th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS*, pp. 1–6*)*, 2018.
[57] S. Althubiti *et al*., "Applying long short-term memory recurrent neural network for intrusion detection," in *Proc. SoutheastCon*, pp. 1–5, 2018.
[58] C. Xu *et al*., "An intrusion detection system using a deep neural network with gated recurrent units," *IEEE Access*, vol. 6, pp. 48697–48707, 2018.
[59] M. A. Ferrag and L. Maglaras, "DeepCoin: A novel deep learning and blockchain-based energy exchange framework for smart grids," *IEEE Trans. Eng. Manag.*, vol. 67, no. 4, pp. 1285–1297, 2019.
[60] A. Chawla *et al*., "Host based intrusion detection system with combined CNN/RNN model," in *ECML PKDD 2018 Workshops*, Springer, pp. 149–158, 2019.