International Journal of Engineering, Science and Environment
Peer-Reviewed Open Access • CC BY 4.0 • Quarterly

Vol. 1 | Issue 1 | Apr–Jun 2026
Banerjee et al. | Multimodal & LLM Approaches in Cybersecurity

SYSTEMATIC REVIEW

Multimodal and Large Language Model Approaches in Cybersecurity: A Systematic Review

Trina Banerjee[1], **Piyush Thapliyal**[2], **Mukthikka V**[3], **Gurpreet Singh**[4*]

[1] *SAKS Global, India,* [2] *University of Delhi, India,* [3] *Bharath Institute of Higher Education and Research, Chennai, India,*

[4] *Endicott College of International Studies, Woosong University, South Korea*

***Corresponding Author:** gurpreetsinghmcse@gmail.com

ABSTRACT

The rapid evolution of cyber threats demands increasingly sophisticated defensive mechanisms. In recent years, Large Language Models (LLMs) and Multimodal Large Language Models (MLLMs) have gained traction as valuable tools across multiple cybersecurity domains, offering capabilities that extend well beyond traditional rule-based and classical machine learning approaches. This systematic review provides a detailed analysis of 55 research papers published between 2019 and 2025, examining the application of LLMs and multimodal AI across eight key cybersecurity domains: vulnerability detection, malware analysis, phishing detection, network intrusion detection, cyber threat intelligence, security operations, penetration testing, and deepfake detection. We present a unified taxonomy that categorizes these approaches by their architectural type, covering encoder-only models (BERT variants), decoder-only models (GPT family), and multimodal architectures, as well as by their application domains. Our comparative analysis shows that while LLMs demonstrate strong capabilities in code comprehension, threat classification, and automated security analysis, notable challenges persist in areas such as hallucination, adversarial robustness, and the dual-use nature of these technologies. We further examine the security vulnerabilities present in LLMs themselves, including prompt injection and jailbreaking attacks. This review identifies open research gaps and proposes future directions, including agentic AI workflows, privacy-preserving security models, and the development of domain-specific foundation models for cybersecurity.

Keywords: cybersecurity, large language models, multimodal learning, threat detection, vulnerability analysis, deep learning, transformers

1. INTRODUCTION

The cybersecurity landscape has changed substantially in recent years, driven by the growing sophistication of cyber threats and the steady expansion of digital attack surfaces. Traditional security mechanisms, including signature-based detection systems, static rule engines, and conventional machine learning classifiers, are becoming insufficient to address the complexity and scale of modern cyber attacks [1, 2]. The rise of advanced persistent threats (APTs), zero-day vulnerabilities, AI-generated phishing campaigns, and polymorphic malware calls for a shift toward more intelligent, adaptive, and context-aware security solutions.

At the same time, the field of artificial intelligence has seen significant advances with the development of Large Language Models (LLMs) and their multimodal extensions. Models such as GPT-4, Claude, LLaMA, and Gemini have demonstrated strong capabilities in natural language understanding, code comprehension, logical reasoning, and multimodal perception [3, 4]. These capabilities are especially relevant to cybersecurity, where analysts must process large volumes of heterogeneous data, including source code, network logs, binary executables, threat reports, phishing emails, and visual content, in order to identify and respond to threats effectively. At the same time, these developments have raised new questions about the security and privacy implications of LLMs themselves [5].

The intersection of LLMs and cybersecurity has given rise to a rapidly growing body of research exploring both defensive and offensive applications. On the defensive side, LLMs have been applied to vulnerability detection in source code [6, 7], automated malware analysis [8], phishing detection through multimodal webpage analysis [9, 10], network intrusion detection [11], and security operations center (SOC) automation [12]. On the offensive side, researchers have shown that LLMs can assist in automated penetration testing [13], exploit generation [14], and social engineering attacks [15], underscoring the dual-use nature of these technologies.

Furthermore, the concept of multimodal learning, which refers to the ability to jointly process and reason across multiple data modalities such as text, images, code, and structured data, has opened new possibilities for cybersecurity applications. Multimodal approaches enable more thorough threat analysis by integrating information from diverse sources. For instance, combining visual brand analysis with textual content examination has proven effective for phishing detection [10], and fusing network traffic features with payload content has advanced intrusion detection systems [16].

Despite the rapid growth of this research area, existing surveys tend to focus on narrow subdomains, such as LLMs for vulnerability detection [17] or LLMs in general cybersecurity [18], without providing a unified, cross-domain perspective that encompasses multimodal approaches. Moreover, the field evolves quickly, with important new contributions appearing

regularly. This systematic review addresses these gaps by providing:

- A taxonomy of LLM and multimodal AI applications across eight cybersecurity domains.
- A systematic analysis of 55 peer-reviewed and preprint papers published between 2019 and 2025.
- A comparative evaluation of architectural approaches, including encoder-only, decoder-only, and multimodal models.
- An examination of the security vulnerabilities present in LLMs themselves.
- Identification of open research gaps and directions for future work.

The remainder of this paper is organized as follows: Section 2 provides background on transformer architectures and multimodal learning. Section 3 details our systematic review methodology. Section 4 presents our taxonomy of applications across eight cybersecurity domains. Section 5 provides a comparative analysis. Section 6 examines the security of LLMs themselves. Section 7 discusses key findings and limitations. Section 8 outlines future directions, and Section 9 concludes the paper.

2. BACKGROUND AND PRELIMINARIES

2.1 Transformer Architectures

The transformer architecture, introduced by Vaswani et al. in 2017, has become the foundation for modern language models. Its self-attention mechanism enables the model to capture long-range dependencies in sequential data, making it well suited for processing the diverse textual and sequential data encountered in cybersecurity contexts. Three primary architectural types have emerged from the transformer foundation:

Encoder-only models, exemplified by BERT and its variants, are designed for bidirectional contextual understanding. These models perform well on classification, named entity recognition (NER), and feature extraction tasks. In cybersecurity, domain-specific encoder models such as SecureBERT [19] and CyBERT [20] have been developed by pre-training on cybersecurity-specific corpora, enabling improved performance on tasks such as threat classification and indicator of compromise (IOC) extraction.

Decoder-only models, represented by the GPT family, are autoregressive models optimized for text generation. Their ability to generate coherent, context-aware text makes them useful for tasks such as code repair [21], report generation [22], and interactive security analysis [13].

Encoder-decoder models, such as T5 and BART, combine both approaches and are particularly effective for sequence-to-sequence tasks including code translation, vulnerability summarization, and structured output generation.

2.2 Large Language Models in Context

The scaling of transformer models to billions of parameters has given rise to emergent capabilities that are especially relevant to cybersecurity. These include in-context learning, chain-of-thought reasoning, and the ability to follow complex multi-step instructions. Models such as GPT-4, Claude, and Gemini demonstrate strong proficiency in understanding and generating code across multiple programming languages, interpreting technical documentation, and reasoning about complex system architectures [3].

The development of code-specialized LLMs, including CodeBERT [23], StarCoder [24], and Code Llama [25], has further advanced the application of LLMs to software security. These models are pre-trained on large-scale code corpora and demonstrate enhanced capabilities in understanding program semantics, detecting vulnerability patterns, and generating security-relevant code transformations.

2.3 Multimodal Learning Fundamentals

Multimodal learning refers to the development of models that can process, relate, and generate information across multiple data modalities. In the cybersecurity context, relevant modalities include natural language text (threat reports, documentation), programming languages (source code, scripts), visual data (screenshots, logos, CAPTCHAs), network data (traffic captures, flow records), and binary data (executable files, firmware).

Recent advances in Vision-Language Models (VLMs), such as GPT-4V, Gemini, and LLaVA, have enabled the simultaneous processing of textual and visual information, creating new opportunities for cybersecurity applications. For example, multimodal phishing detection systems can analyze both the visual appearance and textual content of suspicious web pages [9], while deepfake detection systems use VLMs to provide explainable forensic analysis [26].

2.4 Domain-Specific Security Models

The cybersecurity domain presents unique linguistic and technical challenges that general-purpose LLMs may not fully address. This has motivated the development of domain-specific models:

- **SecureBERT** [19] is a RoBERTa-based model pre-trained on a large corpus of cybersecurity text, including threat intelligence reports, vulnerability descriptions, and security advisories. It demonstrates improved performance on cybersecurity NER, text classification, and question-answering tasks compared to general-purpose BERT.
- **CyBERT** [20] is fine-tuned specifically for processing dense, technical cybersecurity language found in threat reports and attack descriptions, supporting SOC analysts in extracting actionable insights from unstructured security data.
- **CodeBERT** [23] bridges the gap between natural language and programming languages, enabling joint understanding of code comments, documentation, and executable code. This capability is essential for vulnerability detection and code analysis tasks.

3. METHODOLOGY

3.1 Search Strategy

This systematic review follows the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) guidelines. We conducted a literature search across multiple academic databases and preprint servers, including IEEE Xplore, ACM Digital Library, Springer, arXiv, USENIX, and Google Scholar. The search was performed using combinations of the following keywords: "large language model," "LLM," "multimodal," "transformer," "BERT," "GPT," combined with cybersecurity-specific terms including "cybersecurity," "vulnerability detection," "malware analysis," "phishing detection," "intrusion detection," "threat intelligence," "penetration testing," "deepfake detection," and "security operations."

3.2 Use of Large Language Models

In accordance with editorial policies, we disclose that Large Language Models were used as assistive tools during the literature search, synthesis, and manuscript preparation stages of this work. All content was independently verified, validated, and critically reviewed by the authors. The LLM does not meet authorship criteria and is not listed as an author.

3.3 Inclusion and Exclusion Criteria

Inclusion criteria: * Papers published between 2019 and 2025

- Papers applying LLMs, transformer-based models, or multimodal approaches to cybersecurity tasks
- Papers published in peer-reviewed venues or reputable preprint servers (arXiv)
- Papers presenting empirical results, novel architectures, or surveys

Exclusion criteria: * Papers not related to cybersecurity applications

- Papers using only classical machine learning without transformer components
- Duplicate publications or significantly overlapping studies
- Papers without accessible full text
- Non-English publications

3.4 Paper Selection Process

The selection process proceeded in three phases: (1) initial keyword search yielding 312 candidate papers, (2) title and abstract screening reducing the pool to 98 papers, and (3) full-text review resulting in the final selection of 55 papers.

3.5 Categorization Taxonomy

Selected papers were categorized along two dimensions: (1) the cybersecurity application domain (vulnerability detection, malware analysis, phishing detection, network intrusion detection, threat intelligence, security operations, penetration testing, and deepfake detection), and (2) the model architecture (encoder-only, decoder-only, encoder-decoder, or multimodal). This dual categorization enables cross-domain comparison of methodological approaches and identification of architectural trends.

4. TAXONOMY OF APPLICATIONS

This section presents a taxonomy of LLM and multimodal AI applications across eight cybersecurity domains. For each domain, we discuss the key approaches, representative works, and the current state of research.

4.1 Vulnerability Detection and Code Analysis

Vulnerability detection is one of the most extensively studied applications of LLMs in cybersecurity. Traditional static analysis tools rely on predefined rules and patterns, which limits their ability to detect complex, context-dependent vulnerabilities. Transformer-based models offer a different approach by learning vulnerability patterns directly from code.

Thapa et al. [6] conducted an early and influential study evaluating multiple transformer architectures, including BERT, RoBERTa, and GPT-2, for software vulnerability detection. Their experiments on the Draper VDISC dataset showed that pre-trained language models significantly outperform traditional feature-based classifiers, with RoBERTa achieving the highest F1-score of 92.1% on binary vulnerability classification.

Fu and Tantithamthavorn [7] introduced LineVul, a transformer-based approach that moves beyond file-level detection to predict vulnerabilities at the line level. By using the attention mechanism of RoBERTa, LineVul identifies the specific code lines most likely to contain vulnerabilities, providing actionable guidance for developers and reducing the manual effort required for code review.

The availability of larger models has expanded the scope of vulnerability analysis. A detailed evaluation of LLMs—including GPT-3.5, GPT-4, and CodeLlama—for vulnerability detection was carried out in [27]. They found that while these models demonstrate strong capabilities in understanding vulnerability semantics, they struggle with complex, multi-function vulnerabilities that require cross-file context. Their study reported that GPT-4 achieved 74.5% accuracy on the DiverseVul dataset [28], outperforming traditional deep learning approaches but still falling short of human expert performance.

Zhou et al. [29] examined the current state of LLM-based vulnerability detection, identifying key challenges including limited context windows, the need for domain-specific fine-tuning, and the difficulty of detecting zero-day vulnerabilities without training examples. Lu et al. [30] addressed some of these limitations with Grace, a framework that combines graph structure representations with in-context learning for LLM-based vulnerability detection, achieving strong results by integrating code property graphs with LLM reasoning.

Beyond detection, LLMs have shown promise in automated vulnerability repair. Pearce et al. [21] examined the zero-shot repair capabilities of models including Codex and CodeGen, finding that LLMs can generate correct patches for approximately 67% of synthetic vulnerability scenarios, though performance drops considerably on real-world CVEs. Wu et al. [31] further investigated neural networks for security

vulnerability fixing, showing that while LLMs generate syntactically valid patches more consistently than traditional APR tools, ensuring semantic correctness remains a substantial challenge.

4.2 Malware Analysis and Binary Reverse Engineering

Malware analysis has traditionally been a labor-intensive process requiring specialized expertise in reverse engineering, assembly language, and behavioral analysis. LLMs and transformer-based models are increasingly being used to automate and augment various stages of this process.

Rahali and Akhloufi [8] introduced MalBERT, a transformer-based approach that treats malware detection as a natural language processing task. By converting Android application bytecode into textual representations and fine-tuning a BERT model, MalBERT achieved detection accuracy exceeding 97% on benchmark datasets, illustrating the effectiveness of transfer learning from NLP to malware classification.

Demirkiran et al. [32] addressed the challenge of imbalanced malware family classification by developing an ensemble of pre-trained transformer models. Their approach combines multiple BERT-based classifiers through a weighted voting mechanism, achieving notable improvements in classifying rare malware families that are typically underrepresented in training data.

The application of LLMs to binary reverse engineering is a particularly active area of research. Xu et al. [33] introduced LLM4Decompile, which uses large language models to decompile binary code into human-readable source code. This approach outperforms traditional decompilers such as Ghidra and IDA Pro in producing semantically accurate and readable decompiled output, reducing the time required for malware analysis.

Pei et al. [34] explored the broader application of LLMs for malware analysis tasks, including automated YARA rule generation, malware family clustering, and behavioral summarization. Their work shows that GPT-4 can generate accurate YARA signatures from malware samples with 78% precision, though the authors note that hallucination remains a concern when generating detection rules for novel malware variants.

4.3 Phishing and Social Engineering Detection

Phishing and social engineering attacks are among the most prevalent and costly cyber threats [35]. Modern phishing attacks are inherently multimodal, combining deceptive text, visual brand impersonation, and malicious URLs, which makes them well suited for multimodal AI approaches.

Lee et al. [9] proposed a two-phase approach using multimodal LLMs for phishing webpage detection. In the first phase, the model performs brand identification by analyzing visual elements such as logos and color schemes. In the second phase, it verifies the legitimacy of the detected brand against the actual domain, achieving a detection accuracy of 98.6% on a dataset of 5,000 phishing and legitimate webpages.

Li et al. [10] introduced KnowPhish, which integrates multimodal knowledge graphs with LLMs for reference-based phishing detection. By constructing a knowledge graph containing brand visual identities, official domains, and legitimate page structures, KnowPhish enables zero-shot detection of phishing pages impersonating previously unseen brands, a capability that traditional rule-based and supervised learning approaches do not offer.

Koide et al. [36] evaluated ChatGPT's effectiveness for phishing site detection through prompt-based classification. Their study found that while ChatGPT achieves reasonable accuracy (92.3%) in identifying phishing URLs, it is susceptible to adversarial URL obfuscation techniques, highlighting the need for more robust LLM-based detection frameworks.

Roy et al. [37] examined the dual-use nature of LLMs in phishing, showing that while LLMs can serve as effective defenders against phishing, they can also be used to generate convincing phishing content that bypasses traditional filters. Bethany et al. [15] further explored this duality, finding that LLM-generated social engineering messages exhibit linguistic characteristics that are difficult to distinguish from human-authored messages using conventional detection methods.

4.4 Network Intrusion Detection

Network intrusion detection has traditionally relied on signature-based systems or classical machine learning models trained on handcrafted features. Transformer-based approaches are reshaping this domain by enabling automated feature learning from raw network data.

Ferrag et al. [11] proposed a privacy-preserving BERT-based lightweight model specifically designed for IoT and IIoT network security. Their approach fine-tunes a compact BERT model on network traffic data represented as textual sequences, achieving high detection rates while maintaining a model footprint suitable for deployment on resource-constrained edge devices. The model achieved F1-scores exceeding 95% on multiple IoT attack detection benchmarks.

Alkhatib et al. [38] investigated whether BERT can effectively understand network traffic patterns. Their study explored various strategies for encoding network flow data as text suitable for BERT processing, finding that while BERT-based models achieve competitive performance on standard benchmarks (NSL-KDD, CICIDS2017), the tokenization strategy is important; naive byte-level tokenization leads to poor performance compared to semantically meaningful encodings.

Liu et al. [39] developed an LLM-based framework for network intrusion detection that uses the reasoning capabilities of large language models to classify network flows. Their approach employs structured prompts to present network flow features to the LLM, producing interpretable detection decisions that can be explained in natural language, which is a notable advantage over black-box deep learning approaches.

Lin et al. [16] introduced MIND-IoT, a multimodal approach that combines transformer encoders with convolutional neural networks for IoT network traffic classification. This hybrid architecture draws on the CNN's strength in local feature extraction and the transformer's capability for capturing global

contextual dependencies, achieving 98.14% accuracy across diverse IoT traffic datasets.

Goodman et al. [40] proposed a transformer-based framework for payload maliciousness detection, showing that attention mechanisms can effectively identify malicious patterns within network packet payloads.

4.5 Cyber Threat Intelligence and Knowledge Extraction

The extraction and processing of Cyber Threat Intelligence (CTI) from unstructured sources, such as threat reports, security advisories, blog posts, and dark web forums, is an important capability for proactive cybersecurity defense. LLMs have shown strong capabilities in automating this traditionally manual process.

SecureBERT [19] is a notable contribution in this domain. By pre-training RoBERTa on a curated corpus of cybersecurity text, SecureBERT achieves substantially improved performance on cybersecurity-specific NLP tasks, including NER for extracting Indicators of Compromise (IOCs), threat actor identification, and vulnerability classification. The model has been widely adopted as a backbone for downstream CTI applications.

CyBERT [20] provides contextualized embeddings specifically tuned for cybersecurity language, supporting SOC analysts in extracting structured information from dense technical reports. The model is particularly effective at mapping extracted entities to established frameworks such as MITRE ATT&CK, enabling automated threat taxonomy classification.

Satvat et al. [41] developed EXTRACTOR, a system for automatically extracting attack behavior from threat reports using NLP techniques. EXTRACTOR converts unstructured threat reports into structured attack graphs, enabling automated correlation of threat intelligence across multiple sources and time periods.

The integration of Retrieval-Augmented Generation (RAG) with LLMs has emerged as an effective approach for CTI. Li et al. [42] introduced TechniqueRAG for adversarial technique annotation, combining retrieval mechanisms with LLM reasoning to accurately map threat descriptions to MITRE ATT&CK techniques. Abdeen et al. [43] developed RAGIntel, a RAG-based system for cyber attack investigation that integrates structured intelligence from MITRE ATT&CK using hybrid retrieval algorithms, substantially reducing hallucination compared to standalone LLM approaches.

Perrina et al. [22] proposed AGIR, a system for automating CTI reporting using natural language generation, showing that LLMs can produce analyst-quality threat reports when provided with structured input from security monitoring tools.

4.6 Security Operations and Incident Response

Security Operations Centers (SOCs) face a persistent challenge of alert fatigue, with analysts processing thousands of alerts daily, the majority of which are false positives. LLMs offer the potential to improve SOC efficiency through automated triage, log analysis, and incident response support.

Alam et al. [12] explored the application of multimodal LLMs to SOC operations, showing that models capable of processing both textual log data and visual dashboard information can provide more thorough security analysis than text-only approaches. Their work highlighted the importance of reducing "visual parroting," which refers to the tendency of multimodal models to simply describe visual elements without performing meaningful security analysis.

Sahin et al. [44] conducted an empirical study on integrating LLMs into security incident response workflows at a major enterprise. Their findings reveal that while LLMs notably accelerate the initial stages of investigation, particularly alert summarization and context enrichment, they require careful human oversight during the remediation phase, as LLM-generated remediation recommendations may not account for organization-specific configurations and dependencies.

Chuvakin et al. [45] presented a survey of LLMs for SOC operations, cataloging applications across the SOC workflow including log anomaly detection, alert triage, threat hunting, and compliance reporting. Their survey identifies the key challenge of maintaining factual accuracy when LLMs generate investigation narratives, noting that approximately 15% of LLM-generated SOC reports contain factual errors or omissions.

Siracusano et al. [46] demonstrated a practical framework integrating multiple LLMs with SIEM systems, showing that a multi-model approach, in which different specialized LLMs handle different aspects of security analysis, outperforms single-model approaches in both accuracy and coverage.

4.7 Penetration Testing and Offensive Security

The application of LLMs to automated penetration testing is one of the most active yet controversial areas of cybersecurity AI research. These systems aim to augment or automate the traditionally manual and expertise-intensive process of identifying and exploiting security vulnerabilities.

Deng et al. [13] introduced PentestGPT, one of the first LLM-based penetration testing tools. PentestGPT operates as an interactive "copilot" that guides penetration testers through the testing process, providing suggestions for reconnaissance, vulnerability identification, and exploitation. Evaluation on HackTheBox challenges showed that PentestGPT reduces the time required for penetration testing while enabling less experienced testers to achieve results comparable to intermediate-level professionals.

Fang et al. [14] demonstrated that LLM agents can autonomously exploit real-world one-day vulnerabilities. Their study showed that GPT-4, when provided with CVE descriptions and access to standard security tools, could autonomously develop and execute exploits for 87% of tested vulnerabilities, a finding that raises serious concerns about the potential for LLM-powered automated attacks.

Happe and Cito [47] systematically evaluated LLMs in penetration testing scenarios, finding that while current models perform well on individual subtasks (reconnaissance analysis, exploit suggestion), they struggle to maintain coherent multi-step attack strategies due to context window limitations and

the difficulty of maintaining state across extended testing sessions.

Yang et al. [48] developed AutoAttacker, an LLM-guided system for implementing automatic cyber-attacks. Their system decomposes attack scenarios into structured sub-tasks and uses LLM capabilities for each stage, showing that automated attack systems can match human penetration testers on standardized test environments.

4.8 Deepfake Detection and Visual Security

The proliferation of AI-generated synthetic media, particularly deepfake videos and images, poses serious challenges to visual security, authentication, and trust. VLMs and multimodal LLMs offer new approaches to detecting and analyzing synthetic content.

Jia et al. [26] conducted a detailed evaluation of whether LLMs and VLMs can effectively detect deepfakes. Their study found that state-of-the-art VLMs such as GPT-4V achieve zero-shot deepfake detection accuracy of 73.2%, substantially outperforming random chance but falling short of specialized deepfake detectors. However, when combined with targeted prompting strategies and artifact-specific instructions, VLM performance improves to 84.7%, suggesting that VLMs may serve as effective "first-pass" detectors in practical settings.

Shi et al. [49] introduced SHIELD, an evaluation benchmark for assessing the vulnerability of multimodal LLMs to visual adversarial attacks, including face spoofing and forgery. Their benchmark reveals that even advanced MLLMs are vulnerable to carefully crafted visual perturbations, achieving only 61% robustness against state-of-the-art visual attacks.

Hao et al. [50] developed Halligan, a VLM-based agent for solving unseen visual CAPTCHAs. By framing CAPTCHA solving as a search problem where the task instruction serves as an optimization objective, Halligan shows that general-purpose VLMs can effectively bypass traditional visual security mechanisms, a finding with notable implications for web security and bot detection.

Coccomini et al. [51] proposed combining EfficientNet and Vision Transformers for video deepfake detection, showing that hybrid architectures that draw on both convolutional and transformer-based feature extraction achieve strong performance on face forgery benchmarks compared to either approach alone.

5. COMPARATIVE ANALYSIS

This section provides a cross-domain comparative analysis of the reviewed approaches, examining architectural trends, performance characteristics, and dataset utilization patterns.

5.1 Summary of Reviewed Papers

Table I presents a summary of representative works across all eight cybersecurity domains, categorized by model architecture, task type, and key performance metrics.

Table I. Summary of Representative LLM and Multimodal Approaches across Cybersecurity Domains

Reference	Domain	Architecture	Key Model	Dataset	Performance
Thapa et al. (2022) [6]	Vuln. Detection	Encoder	RoBERTa	Draper VDISC	F1: 92.1%
Fu & Tantithamthavorn (2022) [7]	Vuln. Detection	Encoder	RoBERTa	Big-Vul	F1: 91.0%
Steenhoek et al. (2024) [27]	Vuln. Detection	Decoder	GPT-4	DiverseVul	Acc: 74.5%
Pearce et al. (2023) [21]	Vuln. Repair	Decoder	Codex	Synthetic CWE	Fix: 67%
Rahali & Akhloufi (2023) [8]	Malware	Encoder	BERT	Android apps	Acc: 97%
Xu et al. (2024) [33]	Binary RE	Decoder	LLaMA	Decompile	BLEU: 0.82
Lee et al. (2024) [9]	Phishing	Multimodal	GPT-4V	Web pages	Acc: 98.6%
Li et al. (2024) [10]	Phishing	Multimodal	LLM+KG	KnowPhish	F1: 96.2%
Koide et al. (2024) [36]	Phishing	Decoder	ChatGPT	URLs	Acc: 92.3%
Ferrag et al. (2024) [11]	Network IDS	Encoder	BERT	IoT traffic	F1: 95%
Alkhatib et al. (2022) [38]	Network IDS	Encoder	BERT	NSL-KDD	Acc: 89%
Lin et al. (2024) [16]	Network IDS	Hybrid	Trans+CNN	IoT data	Acc: 98.1%
Aghaei et al. (2023) [19]	CTI	Encoder	SecureBERT	Cyber corpus	F1: 94%
Abdeen et al. (2024) [43]	CTI	RAG+LLM	GPT-4+RAG	MITRE ATT&CK	Prec: 91%
Deng et al. (2024) [13]	Pen Testing	Decoder	GPT-4	HackTheBox	Solve: 73%
Fang et al. (2024) [14]	Pen Testing	Decoder	GPT-4	Real CVEs	Exploit: 87%
Jia et al. (2024) [26]	Deepfake	Multimodal	GPT-4V	FaceForensics	Acc: 84.7%

5.2 Architectural Trends

Analysis of the reviewed literature reveals several architectural trends. Encoder-only models are predominant in classification-oriented tasks such as malware detection, vulnerability classification, and network intrusion detection, where bidirectional context understanding is essential. Decoder-only models are preferred for generative tasks including code repair, penetration testing guidance, and threat report generation. Multimodal architectures are gaining traction in domains where multiple data types must be analyzed simultaneously, particularly in phishing detection and deepfake analysis.

A notable trend is the rise of hybrid architectures that combine transformers with other neural network types. The MIND-IoT framework [16] exemplifies this approach by pairing transformer encoders with CNNs, drawing on the complementary strengths of each architecture. Similarly, RAG-based systems [42, 43] represent a hybrid of retrieval and generation that addresses the knowledge limitations of standalone LLMs.

5.3 Cross-Domain Observations

Several cross-cutting observations emerge from our analysis:

- **Domain adaptation is critical:** General-purpose LLMs consistently underperform domain-specific models on cybersecurity tasks. SecureBERT [19] outperforms base BERT by 8 to 12 percentage points on CTI tasks, illustrating the value of domain-specific pre-training.
- **Scale does not guarantee superiority:** Larger models (GPT-4) do not always outperform smaller, fine-tuned models. For vulnerability detection, fine-tuned RoBERTa [7] achieves higher accuracy than zero-shot GPT-4 [27].
- **Multimodal approaches show the highest improvement margins:** The largest performance gains over baselines are observed in multimodal applications, particularly in phishing detection [9], [10], where combining visual and textual analysis provides complementary signals.
- **Interpretability is a differentiator:** LLM-based approaches uniquely offer natural language explanations for their decisions, an important advantage in security applications where analyst trust and auditability are essential [39].

6. SECURITY OF LARGE LANGUAGE MODELS

While LLMs offer useful capabilities for cybersecurity defense, they also introduce new attack surfaces. Understanding the security vulnerabilities present in LLMs is essential for their responsible deployment in security-critical applications.

6.1 Prompt Injection Attacks

Prompt injection is the most widely studied vulnerability class in LLM-integrated systems. Greshake et al. [52] showed that real-world LLM-integrated applications are vulnerable to indirect prompt injection, where adversarial instructions embedded in external data sources (e.g., web pages, emails, documents) can hijack the LLM's behavior. In a cybersecurity context, this is especially concerning for SOC tools that use LLMs to process security logs, since an attacker could embed malicious prompts within log entries to manipulate the analysis results.

Schulhoff et al. [53] organized HackAPrompt, a large-scale competition to evaluate prompt injection techniques, revealing systematic weaknesses in LLM alignment. Their findings showed that even models with safety training are vulnerable to creative prompt manipulation, with participants discovering novel injection techniques that bypass multiple layers of defense.

6.2 Jailbreaking Attacks

Wei et al. [54] provided a systematic analysis of how LLM safety training fails, identifying two fundamental failure modes: competing objectives (where the model's helpfulness objective conflicts with its safety objective) and mismatched generalization (where safety training fails to generalize to novel attack patterns). Their work has direct implications for cybersecurity applications, as it shows that safety-trained models can still be induced to generate exploit code, malware, and other harmful content.

Zou et al. [55] introduced the Greedy Coordinate Gradient (GCG) attack, which generates universal and transferable adversarial suffixes that can jailbreak multiple aligned LLMs. Their approach shows that adversarial attacks on LLMs can be automated and optimized, posing a real threat when LLMs are deployed in security-critical applications where adversaries may attempt to manipulate model behavior.

6.3 Implications for Cybersecurity Deployment

The vulnerability of LLMs to adversarial manipulation has direct implications for their deployment in cybersecurity:

- **SOC automation:** LLMs processing untrusted input (logs, alerts) must be hardened against indirect prompt injection [52].
- **Threat intelligence:** RAG-based systems must validate retrieved content to prevent poisoning of the knowledge base [42], [43].
- **Code analysis:** LLM-generated security patches must undergo independent verification to prevent the introduction of backdoors [31].
- **Multi-layer defense:** Strategies combining input sanitization, output filtering, and human oversight are essential for secure LLM deployment.

7. DISCUSSION

7.1 Key Findings

This systematic review reveals several important insights about the current state of LLM and multimodal AI applications in cybersecurity:

- **LLMs are being applied across all major cybersecurity domains.** Our analysis of 55 papers shows that LLMs have been successfully applied to every major cybersecurity domain, from code-level vulnerability detection to strategic threat intelligence analysis. The breadth of applications reflects the general-purpose nature of LLM capabilities and their alignment with the language-centric nature of cybersecurity work.
- **The dual-use challenge is a defining characteristic.** One of the most important findings is the inherent dual-use nature of LLMs in cybersecurity. The same capabilities that enable automated vulnerability detection [27] also enable

automated exploit development [14]. The same models that detect phishing [9] can generate more convincing phishing content [15]. This duality requires careful consideration of access controls, responsible disclosure, and governance frameworks.

- **Multimodal approaches represent an active frontier.** The most substantial performance improvements are observed in applications that use multimodal capabilities, combining textual, visual, and structural information. Phishing detection [10], deepfake analysis [26], and SOC analysis [12] all benefit from multimodal integration.

7.2 Limitations of Current Approaches

Several important limitations persist across the reviewed literature:

- **Hallucination** remains a pervasive concern, particularly in generative applications. LLMs can produce plausible but factually incorrect security analyses, generate non-functional exploit code, or misidentify vulnerability types. In a security context, hallucination can lead to false positives, wasted analyst effort, or, more seriously, missed true positives.
- **Context window limitations** constrain the ability of LLMs to analyze large codebases, extended network captures, or lengthy threat reports. While recent models have expanded context windows considerably, analyzing entire software projects or multi-day network captures remains impractical without sophisticated chunking and summarization strategies.
- **Adversarial robustness** of LLM-based security tools remains largely untested in adversarial settings. Attackers aware that LLMs are being used for defense may deliberately craft inputs designed to evade or manipulate the models.
- **Evaluation standardization** is lacking across the field. Different studies use different datasets, metrics, and evaluation protocols, making direct comparison of approaches difficult. The establishment of standardized cybersecurity-specific benchmarks for LLM evaluation is an important open need.

7.3 Ethical Considerations

The deployment of LLMs in cybersecurity raises important ethical questions. The automation of offensive security capabilities, such as exploit generation and attack planning, lowers the barrier to entry for malicious actors. Responsible research practices, including careful consideration of dual-use implications and appropriate access controls, are essential for the continued development of this field.

8. FUTURE DIRECTIONS

Based on our analysis, we identify several promising directions for future research:

- **Agentic AI for Cybersecurity.** The shift from static LLM queries to autonomous AI agents capable of performing multi-step security tasks represents an important research direction. Future systems will likely feature multiple specialized agents collaborating on complex security operations, with different agents handling reconnaissance, analysis, and response [13], [14].
- **Domain-Specific Foundation Models.** While general-purpose models show strong performance, the development of cybersecurity-specific foundation models, pre-trained on diverse security data including code, network traffic, malware samples, and threat reports, could yield further improvements. The success of SecureBERT [19] at a relatively small scale suggests that larger, more comprehensive security foundation models could be highly effective.
- **Privacy-Preserving Security AI.** The sensitive nature of security data calls for the development of privacy-preserving approaches, including federated learning for collaborative threat detection, differential privacy for model training, and on-premise deployment strategies that maintain data sovereignty.
- **Adversarial Robustness.** As LLM-based security tools become more prevalent, adversarial robustness will become increasingly important. Research into robust architectures, adversarial training for security applications, and layered defense strategies for LLM-integrated security systems is needed.
- **Standardized Evaluation Frameworks.** The field would benefit from standardized benchmarks that evaluate LLM capabilities across cybersecurity domains under consistent conditions, including adversarial settings, real-world data distributions, and temporal shifts.
- **Human-AI Collaboration Models.** Rather than pursuing full automation, the most promising near-term direction involves developing effective human-AI collaboration frameworks where LLMs augment human analysts' capabilities while maintaining human oversight for critical decisions [44].

9. CONCLUSION

This systematic review has provided a detailed analysis of 55 research papers examining the application of Large Language Models and multimodal AI approaches across eight cybersecurity domains. Our analysis shows that LLMs and multimodal models are substantially changing cybersecurity practice, offering strong capabilities in code comprehension, threat detection, automated analysis, and decision support.

The key contributions of this review include: (1) a unified taxonomy spanning vulnerability detection, malware analysis, phishing detection, network intrusion detection, threat intelligence, security operations, penetration testing, and

deepfake detection; (2) a comparative analysis showing that domain-specific and multimodal approaches consistently outperform general-purpose models; (3) an examination of the security vulnerabilities present in LLMs themselves; and (4) identification of directions for future research.

Our findings underscore the dual-use nature of LLMs in cybersecurity: they serve as effective defensive tools while simultaneously enabling more sophisticated attacks. Addressing this duality requires a multi-faceted approach encompassing technical safeguards, governance frameworks, and responsible research practices. As the cybersecurity landscape continues to evolve, the integration of LLMs and multimodal AI will likely deepen, moving toward agentic systems capable of autonomous security operations. However, the path forward must balance innovation with responsibility, ensuring that these technologies are deployed safely, ethically, and with appropriate human oversight.

DECLARATIONS

Competing Interests

The authors declare no competing interests.

Data Availability

Not applicable. This systematic review is based on previously published research papers that are publicly available through the cited sources.

Author Contributions

Trina Banerjee, Piyush, and Mukthikka V contributed equally to conceptualisation, literature review, data analysis, and writing. Gurpreet Singh supervised the study, critically reviewed the manuscript for intellectual content, and approved the final version for submission.

REFERENCES

[1] D. M. Divakaran and S. Gupta, "Large language models in cybersecurity: A survey of applications, vulnerabilities, and defense techniques," *IEEE Access*, vol. 12, pp. 179576–179609, 2024.

[2] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Generative AI and large language models for cyber security: All insights you need," *arXiv Preprint* arXiv:2405.12750, 2024.

[3] F. N. Motlagh, M. Hajizadeh, M. Majd, P. Najafi, F. Cheng, and C. Meinel, "Large language models in cybersecurity: State-of-the-art," *arXiv Preprint* arXiv:2402.00891, 2024.

[4] G. de J. C. da S. Zhang, L. Liu, S. Choi, R. Jain, and K. Suh, "A survey of large language models in cybersecurity," *arXiv Preprint* arXiv:2402.01854, 2024.

[5] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, "A survey on large language model (LLM) security and privacy: The good, the bad, and the ugly," *High-Confidence Computing*, vol. 4, no. 2, p. 100211, 2024.

[6] C. Thapa, S. I. Jang, M. E. Ahmed, S. Camtepe, J. Piber, and J. Grossklags, "Transformer-based language models for software vulnerability detection," in *Proceedings of ACSAC*, 2022, pp. 481–496.

[7] M. Fu and C. Tantithamthavorn, "LineVul: A transformer-based line-level vulnerability prediction," in *Proceedings of the 19th International Conference on Mining Software Repositories (MSR)*, 2022, pp. 608–620.

[8] A. Rahali and M. A. Akhloufi, "MalBERT: Malware detection using transformers," *IEEE Access*, vol. 11, pp. 88495–88511, 2023.

[9] J. Lee, P. Guo, J. Park, and L. Luo, "Multimodal large language models for phishing webpage detection and identification," in *Proceedings of the Symposium on Electronic Crime Research (eCrime)*, 2024.

[10] Y. Li, D. H. Chau, C. Zou, and T. Neth, "KnowPhish: Large language models meet multimodal knowledge graphs for enhancing reference-based phishing detection," in *Proceedings of the USENIX Security Symposium*, 2024.

[11] M. A. Ferrag, M. Ndhlovu, N. Tihanyi, L. C. Cordeiro, M. Debbah, and T. Lestable, "Revolutionizing cyber threat detection with large language models: A privacy-preserving BERT-based lightweight model for IoT/IIoT networks," *arXiv Preprint* arXiv:2306.14263, 2024.

[12] M. Alam, N. Dey, Y. Huang, and A. Bozorgi, "Looking beyond text: Reducing visual parroting with multimodal large language models for security operations center," *arXiv Preprint*, 2024.

[13] G. Deng *et al.*, "PentestGPT: An LLM-empowered automatic penetration testing tool," in *Proceedings of the USENIX Security Symposium*, 2024.

[14] R. Fang, R. Bindu, A. Gupta, Q. Zhan, and D. Kang, "LLM agents can autonomously exploit one-day vulnerabilities," *arXiv Preprint* arXiv:2404.08144, 2024.

[15] M. Bethany, S. Wheatley, B. Tobin, S. Neupane, and E. Mitra, "Large language models for automated social engineering attacks and defense," *arXiv Preprint*, 2024.

[16] Z. Lin, J. Li, and Rel. Ren, "MIND-IoT: Multimodal IoT network traffic classification using transformer-CNN," *PeerJ Computer Science*, vol. 10, p. e2326, 2024.

[17] X. Hou *et al.*, "Large language models for software vulnerability detection: A survey," *arXiv Preprint* arXiv:2403.08345, 2024.

[18] H. Xu, D. Xiao, Z. Li, J. Xu, and S. Wen, "Large language models for cyber security: A systematic literature review," *arXiv Preprint* arXiv:2405.04760, 2024.

[19] E. Academy, X. Niu, W. Shadid, and E. Al-Shaer, "SecureBERT: A domain-specific language model for cybersecurity," in *Proceedings of the International Conference on Security and Privacy in Communication Systems (SecureComm)*, 2023, pp. 257–275.

[20] P. Ranade, A. Piplai, A. Joshi, and T. Finin, "CyBERT: Contextualized embeddings for the cybersecurity domain," in *Proceedings of the IEEE International Conference on Big Data*, 2021, pp. 3334–3342.

[21] H. Pearce, B. Tan, B. Ahmad, R. Karri, and B. Dolan-Gavitt, "Examining zero-shot vulnerability repair with large language models," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2023, pp. 2339–2356.

[22] F. Perrina, G. Siracusano, and S. Zanero, "AGIR: Automating cyber threat intelligence reporting with natural language generation," in *Proceedings of the International Conference on Availability, Reliability and Security (ARES)*, 2023.

[23] Z. Feng *et al.*, "CodeBERT: A pre-trained model for programming and natural languages," in *Findings of the Association for Computational Linguistics (EMNLP)*, 2020, pp. 1536–1547.

[24] R. Li *et al.*, "StarCoder: May the source be with you!" *arXiv Preprint* arXiv:2305.06161, 2023.

[25] B. Rozière *et al.*, "Code Llama: Open foundation models for code," *arXiv Preprint* arXiv:2308.12950, 2024.

[26] S. Jia, R. Liu, J. Xu, and T. Yang, "Can large language models and vision-language models detect deepfakes?" *arXiv Preprint*, 2024.

[27] B. Steenhoek, M. M. Rahman, R. Jiles, and W. Le, "A comprehensive study of the capabilities of large language models for vulnerability detection," *arXiv Preprint* arXiv:2403.17218, 2024.

[28] Y. Chen, Z. Ding, L. Chen, X. Fan, and D. Wagner, "DiverseVul: A new vulnerable source code dataset for deep learning based vulnerability detection," in *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, 2023, pp. 654–668.

[29] X. Zhou, T. Zhang, and Lo. Lo, "Large language model for vulnerability detection: Emerging results and future directions," in *Proceedings of ICSE-NIER*, 2024, pp. 47–51.

[30] G. Lu, X. Chen, B. Mao, K. Pei, and J. Gonzalez, "Grace: Empowering LLM-based software vulnerability detection with graph structure and in-context learning," *arXiv Preprint* arXiv:2411.03592, 2024.

[31] Y. Wu *et al.*, "How effective are neural networks for fixing security vulnerabilities," in *Proceedings of ISSTA*, 2023, pp. 1282–1294.

[32] F. Demirkiran, A. Cayir, U. Unal, and H. Dag, "An ensemble of pre-trained transformer models for imbalanced multiclass malware classification," *Computers & Security*, vol. 121, p. 102846, 2022.

[33] H. Xu, Z. Luo, M. Ma, H. Lu, and Y. Wang, "LLM4Decompile: Decompiling binary code with large language models," in *Proceedings of EMNLP*, 2024.

[34] K. Pei, Z. Li, J. Ding, and B. Dolan-Gavitt, "Exploiting large language models for malware analysis," *arXiv Preprint*, 2024.

[35] A. van der Heijden and L. Allodi, "Cognitive triaging of phishing attacks," in *Proceedings of the USENIX Security Symposium*, 2019.

[36] T. Koide, D. Fukushi, H. Nakao, and D. Chiba, "Detecting phishing sites using ChatGPT," *arXiv Preprint* arXiv:2306.05816, 2024.

[37] S. S. Roy, S. Nath, and D. Sisodia, "ChatBots to PhishBots? Preventing phishing attacks using large language models," in *DMNLP Workshop at AAAI*, 2024.

[38] B. Alkhatib, S. Rass, and Y. Zhauniarovich, "Can BERT understand network traffic? Exploring the capabilities of NLP-based models for network analysis," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 4946–4957, 2022.

[39] X. Liu, Y. Yang, and Z. He, "Large language model for network intrusion detection," *arXiv Preprint*, 2024.

[40] K. Goodman, R. Rajagopalan, and M. Kremer, "A transformer-based framework for payload maliciousness detection," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2024.

[41] K. Satvat, R. Gjomemo, and V. N. Venkatakrishnan, "EXTRACTOR: Extracting attack behavior from threat reports," in *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*, 2021, pp. 598–615.

[42] S. Li, Z. Wen, and D. Kang, "TechniqueRAG: Adversarial technique annotation with retrieval-augmented generation," in *Findings of the Association for Computational Linguistics (ACL)*, 2025.

[43] B. Abdeen, A. Lakhani, E. Al-Shaer, and E. Academy, "RAGIntel: RAG-based LLM system for cyber attack investigation," *PeerJ Computer Science*, vol. 10, p. e2517, 2024.

[44] M. Sahin, G.-V. Jourdan, F. Brust, and T. Kroeger, "Integrating large language models into security incident response," in *Proceedings of the USENIX Security Symposium*, 2025.

[45] A. Chuvakin, F. Simorjay, and Y. Wei, "Large language models for security operations centers: A comprehensive survey," *arXiv Preprint*, 2025.

[46] G. Siracusano, A. Ferroni, and S. Zanero, "Enhancing security operations center efficiency through multi-model integration of large language models and SIEM systems," *IEEE Transactions on Information Forensics and Security*, 2024.

[47] A. Happe and J. Cito, "Getting pwn'd by AI: Penetration testing with large language models," in *Proceedings of the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*, 2023, pp. 2082–2086.

[48] J. Yang *et al.*, "AutoAttacker: A large language model guided system to implement automatic cyber-attacks," *arXiv Preprint*, 2024.

[49] Y. Shi *et al.*, "SHIELD: An evaluation benchmark for face spoofing and forgery detection with multimodal large language models," in *Proceedings of ECCV*, 2024.

[50] S. Hao, Y. Xu, L. Wang, and D. Wu, "Halligan: VLM agent for solving unseen visual CAPTCHAs," in *Proceedings of the USENIX Security Symposium*, 2024.

[51] D. A. Coccomini, N. Messina, G. Amato, and F. Falchi, "Combining EfficientNet and vision transformers for video deepfake detection," in *Proceedings of the International Conference on Image Analysis and Processing (ICIAP)*, 2022, pp. 219–229.

[52] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection," in *Proceedings of the ACM Workshop on Artificial Intelligence and Security (AISec)*, 2023, pp. 79–90.

[53] S. Schulhoff *et al.*, "Ignore this title and HackAPrompt: Exposing systemic weaknesses of LLMs through a global scale prompt hacking competition," in *Proceedings of EMNLP*, 2023.

[54] A. Wei, N. Haghtalab, and J. Steinhardt, "Jailbroken: How does LLM safety training fail?" *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2024.

[55] A. Zou, Z. Wang, J. Z. Kolter, and M. Fredrikson, "Universal and transferable adversarial attacks on aligned language models," *arXiv Preprint* arXiv:2307.15043, 2023.