SJCS Scientific Journal of Computer Science

futuristech PT. Teknologi Futuristik Indonesia

Received: May 7, 2026; Revised: July 20, 2026; Accepted: August 1, 2026; Published: August 2, 2026; Current version: August 2, 2026. e-ISSN: 3110-3170
Digital Object Identifier 10.64539/sjcs.v2i2.2026.506

Article

D2ANN-RL: Defense-in-Depth ANN-Reinforcement Learning Framework for LLM Chatbot Code Injection Mitigation

Victor Omoboye Oluwasegun[1], Oluwatosin Samuel Falebita[2], Nabeela Temitayo Adebola[3], Victor Aduragbemi Adekunle[4], Divine Chukwuemeka Uzodinma[5], Toluhi Michael Lanre[6], David Oyewumi Oyekunle[7], Chima-Duru Goodness Goziechukwu[8], Ugochukwu Okwudili Matthew[9,*]

[1] Data Science Department, University of Roehampton, London, SW15 5PJ, United Kingdom.
[2] Computer Science, Colorado State University, Fort Collins, 80521, United States.
[3] Cyber Security Threat Intelligence Forensic, University of Salford, Manchester, M5 4WT, United Kingdom.
[4] Department of Mathematics Modelling, University of Ilorin, Ilorin, 240003, Nigeria.
[5] Cyblack Cybersecurity, Manchester, M20 2QW, United Kingdom.
[6] Data Science, University of Salford, Manchester, M5 4WT, United Kingdom.
[7] Project Management, University of Salford, Manchester, M5 4WT, United Kingdom.
[8] Computer Engineering, Caritas University, Enugu, 400001, Nigeria.
[9] Computer Science Department, Federal University of Lavras, Minas Gerais, 37203-202, Brazil;
e-mail: okwudili.ugochukwu@estudante.ufla.br (U. O. Matthew).
* Correspondence

The authors received no financial support for the research, authorship, and/or publication of this article.

Abstract: The growing cybersecurity vulnerabilities in artificial intelligence (AI) service models, particularly Large Language Models (LLMs), highlight code injection as a critical threat to chatbot reliability and safe deployment. On the account that LLMs process inputs as undifferentiated token sequences, they cannot reliably distinguish trusted system prompts from untrusted user inputs. This architectural limitation enables attackers to exploit direct and indirect prompt injection channels, resulting in insecure code generation, altered execution flows, and potential data exfiltration or remote code execution. In mission critical environments such as cloud platforms, IoT ecosystems, and defense systems, these risks escalate into unauthorized access and operational compromise. To address this challenge, the present study introduced a D2ANN-RL framework that integrates input/output sanitization, context isolation, sandboxing, and secure prompt engineering, supported by hybridization of Artificial Neural Network (ANN)–Reinforcement Learning (RL) detection model. The ANN component ensures robust feature extraction, while RL dynamically adapts defense strategies to evolving adversarial vectors. Computational evaluation demonstrates the framework's effectiveness, achieving 96.95% detection accuracy, precision of 96.9%, recall of 97%, and F-Score of 96.95%. The Defense Performance Index (DPI) reached 84.9%, validating model resilience, scalability, and balanced classification integrity. These findings highlight the broader implications of deploying transparent, adaptive, and generalizable safeguards for LLM based chatbot systems, advancing secure AI integration and mitigating systemic vulnerabilities in mission critical operations.

Keywords: Large Language Models (LLMs); LLM-chatbots; Code injection mitigation; Artificial Neural Networks (ANN); Reinforcement Learning (RL); Adversarial Manipulation; Chatbots; AI Conversational Agents.

1. Introduction

The potential of artificial intelligence (AI)-powered conversational agents has been dramatically transformed by the emergence of Large Language Models (LLMs), fundamentally changing how humans interact with Generative Artificial Intelligence (GenAI) across diverse domains and customer-oriented services [1], [2]. These advances have significantly enhanced natural language

understanding and generation, enabling conversational agents to perform increasingly sophisticated tasks with human-like fluency. These models, which enable natural human-machine interactions, have completely changed the technological landscape and constitute a major breakthrough in AI and robotics innovation [3], [4]. LLMs excel at natural language processing (NLP) tasks, including summarization, translation, question answering, and automatic coding support [5]. LLM-powered conversational agents have established a new benchmark for NLP systems by demonstrating human-like reasoning, contextual awareness, and advanced dialogue generation across applications ranging from customer service automation to personalized virtual assistants [6]. Their capabilities stem from breakthroughs in deep learning, particularly transformer-based architectures such as Generative Pre-trained Transformer (GPT) and Bidirectional Encoder Representations from Transformers (BERT) [7]. These models, trained on vast datasets, enable chatbots to deliver coherent, logical, and contextually appropriate responses that adapt dynamically to user intent. The progression reflects not only technical sophistication but also a fundamental shift in human-machine interaction, as these systems increasingly provide seamless and intuitive communication experiences that closely resemble human conversation [1].

Despite these remarkable capabilities, the rapid adoption of LLM-powered conversational agents has also introduced significant cybersecurity challenges. LLMs are increasingly integrated into machine learning design, AI systems engineering, and software security, while their incorporation into conversational agents continues to reshape these disciplines. Although this integration accelerates software development and enhances user interaction, it simultaneously introduces new cybersecurity vulnerabilities, particularly code injection attacks, requiring stronger safeguards to ensure secure, reliable, and ethical deployment in modern software systems and Internet of Things (IoT) platforms [8], [9]. Among these emerging threats, code injection has become one of the most critical security concerns because it can manipulate chatbot behavior, compromise sensitive information, and potentially lead to unauthorized system access. Addressing these vulnerabilities is therefore essential to guarantee trust, resilience, and the secure deployment of LLM-powered applications while protecting real-world systems from data exfiltration, remote code execution, and other malicious activities.

The underlying cause of these vulnerabilities lies in the architectural characteristics of LLMs. Unlike conventional software that follows explicitly programmed logic, LLMs interpret all inputs as undifferentiated token sequences and generate outputs based on probabilistic predictions derived from learned patterns [10]. This inherent ambiguity makes them particularly susceptible to adversarial manipulation, most notably through prompt injection attacks. The Open Worldwide Application Security Project (OWASP) has identified prompt injection as the leading security vulnerability affecting LLM applications [11]. These attacks include direct prompt injection (jailbreaking), in which users deliberately override model safeguards, and indirect prompt injection, where malicious instructions are embedded within external resources such as websites, documents, or files.

The consequences of these attacks extend well beyond manipulated conversations. In software engineering contexts, these weaknesses can escalate into insecure code generation, where models reproduce unsafe programming patterns such as inadequate input validation, and data poisoning, which corrupts training datasets or knowledge bases to induce systematic errors. These risks become even more severe when conversational agents are granted excessive autonomy, enabling unrestricted interaction with Application Programming Interfaces (APIs), databases, or operating system commands without sufficient human oversight. Consequently, adversarial manipulation can substantially increase the likelihood of catastrophic system compromise and undermine the trustworthiness of LLM-powered applications.

Although LLMs have transformed conversational AI by enabling context-aware dialogue across diverse application domains, their architectural ambiguity—where trusted instructions and untrusted inputs are processed uniformly—creates security vulnerabilities that adversaries can exploit through prompt and code injection attacks. Conventional defense mechanisms, such as prompt sanitization, context isolation, and rule-based filtering, have been proposed to mitigate these risks. However, these approaches generally rely on static protection strategies that are difficult to adapt to increasingly sophisticated and evolving adversarial techniques. This challenge remains unresolved because conventional intrusion detection datasets fail to capture LLM-specific attack vectors, while single-layer defense mechanisms lack the adaptability required to counter emerging threats effectively [12].

To address these limitations, this study proposes the D2ANN-RL Framework, which integrates hierarchical feature extraction using Artificial Neural Networks (ANNs) with adaptive Reinforcement Learning (RL). The proposed framework combines semantic features, including token distribution and dialogue flow, with network-level attributes to improve the detection of code injection attacks against LLM-powered conversational agents. Furthermore, the framework incorporates Gradient-weighted Class Activation Mapping (Grad-CAM) to enhance model transparency by providing interpretable visual explanations for security-related decisions, thereby improving both detection performance and explainability.

The primary contribution of this study is the development of a layered intelligent cybersecurity framework that

combines hierarchical feature learning, adaptive policy optimization, and explainable artificial intelligence to strengthen the resilience of LLM-powered conversational agents against adversarial code injection attacks. By integrating adaptive learning with interpretable decision-making, the proposed framework provides a scalable approach for securing LLM-based conversational systems operating in increasingly complex and dynamic environments.

The remainder of this paper is organized as follows. Section 2 and Section 3 present the study objectives and theoretical framework. Section 4 reviews the related literature. Section 5 describes the research methodology. Section 6 presents the experimental results and discusses the research findings, and Section 7 concludes the study with recommendations for future research.

2. Study Objective

This study aims to investigate the cybersecurity challenges associated with the rapid evolution of LLM-powered conversational agents, with particular emphasis on code injection attacks resulting from their increasing autonomy and integration into modern software systems. While advances such as fine-tuning, multimodal integration, and Retrieval-Augmented Generation (RAG) have significantly expanded the capabilities of conversational agents, they have also broadened the attack surface, particularly when LLMs interact with Application Programming Interfaces (APIs), databases, and operating system commands. To mitigate these security risks, this study proposes the D2ANN-RL Framework, which combines hierarchical feature extraction using Artificial Neural Networks (ANNs), adaptive Reinforcement Learning (RL), and layered defense mechanisms to improve the detection and mitigation of code injection attacks against LLM-powered conversational agents. By aligning technological innovation with cybersecurity requirements, the proposed framework aims to promote resilient chatbot design that balances system performance, ethical responsibility, and robust protection against adversarial manipulation.

The specific objectives of this study are as follows:

a) To develop the novel D2ANN-RL Framework, which combines Artificial Neural Networks (ANNs) for hierarchical feature extraction with Reinforcement Learning (RL) for adaptive policy optimization, thereby providing dynamic resilience against evolving code injection attacks targeting LLM-powered conversational agents.
b) To evaluate the effectiveness of the proposed D2ANN-RL Framework in detecting and mitigating code injection attacks using the CySecBench dataset through comprehensive performance evaluation and explainability analysis, thereby improving the security, reliability, and trustworthiness of LLM-powered conversational agents.
c) To establish a scalable framework for secure LLM deployment by incorporating explainable artificial intelligence, privacy-by-design principles, and standardized evaluation using the CySecBench dataset, thereby supporting transparent and resilient cybersecurity practices for modern conversational AI systems.

3. Theoretical Framework

The vulnerabilities of LLMs in mission-critical environments have been extensively studied, with particular focus on prompt injection, code injection, and adversarial manipulation. Early defenses, such as static filtering and rule-based sanitization, proved brittle against indirect injections and evolving attack vectors. Subsequent research introduced sandboxing to isolate execution environments, yet this approach alone could not address semantic manipulation of prompts. Secure prompt engineering strategies improved resilience but lacked adaptability to novel adversarial tactics, while adversarial training methods often degraded performance and failed to generalize across diverse attack surfaces [13]. More recent work has explored hybrid detection models, combining deep learning with reinforcement learning to dynamically adapt to evolving threats. ANN architectures excel at feature extraction, while RL agents refine defense strategies in real time in cybersecurity studies [14].

Against this backdrop, the D2ANN-RL framework advances the literature by integrating defense-in-depth principles with ANN-RL hybridization, operationalizing multiple safeguards into a unified adaptive architecture for resilient and scalable protection in mission-critical LLM-chatbot systems. Within this framework, conversational agents are conceptualized as systems capable of perceiving environmental stimuli and making reasoned decisions to achieve specific communicative or functional goals [15]. The shift toward LLM-based chatbots is defined by their ability to provide coherent, contextually aware responses that mirror human reasoning and learning capabilities [16]. Modern LLMs such as GPT-3.5, GPT-4, GPT-5, BERT, T5, and Large Language Model Meta AI (LLaMA) serve as the primary decision-making brains within these systems, enabling high performance in tasks ranging from translation to complex question-answering [17], [18]. LLaMA is Meta's flexible open-source model designed to provide natural, context-aware responses while serving as a reasoning brain for advanced conversational agents across diverse sectors [19]. The T5 model, developed by Google, leverages transformer architecture to frame all natural language processing tasks as text-to-text translation problems, providing the flexibility and generative power needed for chatbots to produce fluent, natural and context-aware responses [20]. Figure 1 presents the

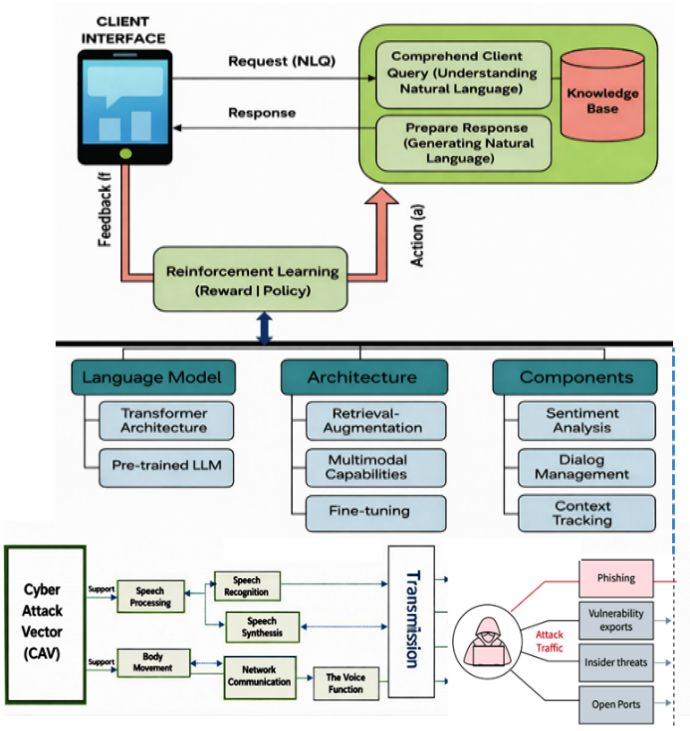

Figure 1. A modified LLM-Powered Conversational Agents with Cybersecurity Vulnerabilities [2].

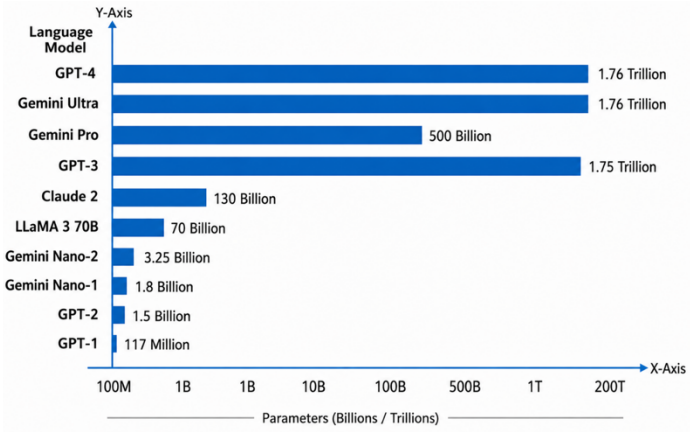

Figure 2. AI models, underscoring the transformative evolution of LLMs in the development of chatbots [17], [20], [21].

conceptual architecture of LLM-powered conversational agents, highlighting their core components and associated cybersecurity vulnerabilities.

GPT-5 adopts a Mixture-of-Experts (MoE) architecture characterized by a parameter capacity extending into the tens of trillions, while selectively activating approximately 125B–635B parameters per token based on model configuration [22]. This sparse activation mechanism optimizes computational efficiency while preserving extensive representational capability, enabling scalable reasoning performance and enhanced adaptability across complex artificial intelligence applications. GPT-4 has been widely reported with an estimated parameter scale of approximately 1.76 trillion (see Figure 2), surpassing GPT-3's 1.75 trillion parameters and signifying a transformative advancement in computational scale, model complexity, and the emergence of deeper, more context-aware conversational intelligence [17], [20], [21]. According to Briouya *et al.* (2024), alternative estimates reported in the literature suggest substantially larger parameter counts for GPT-4, including an estimate of 170 trillion parameters, reflecting ongoing discussions regarding model architecture and parameter estimation [23]. The authors contextualize GPT-4 within AI scaling trends, emphasizing parameter expansion as a key progress indicator while highlighting the inherent trade-off between enhanced performance, computational requirements, and sustainable artificial intelligence development.

This progression also sets the stage for future innovations in AI-driven conversational agents. The integration of RAG promises to ground chatbot responses in real-time, factual data, reducing hallucinations and improving reliability [24]. Meanwhile, multimodal capabilities combining text, voice, and visual inputs are becoming increasingly feasible as models grow in size and sophistication. These advancements will enable chatbots to operate seamlessly across platforms and contexts, from healthcare diagnostics to educational tutoring and customer service support. Within the broad context of conversational AI chatbots, fine-tuning, multimodal integration, and Retrieval Augmented Generation (RAG) represent pivotal advances that expand functionality and resilience. Fine-tuning enables domain-specific adaptation, ensuring precision in sensitive applications such as healthcare and cybersecurity. Multimodal integration extends capabilities beyond text, incorporating voice, image, and paralinguistic signals to enrich human-AI interaction and address diverse communicative needs [25]. RAG grounds chatbot responses in external, real-time knowledge bases, reducing hallucinations and enhancing factual reliability. Collectively, these innovations embody a shift from static, text-only dialogue systems toward adaptive, context-aware architectures. Within this study, they signify methodological pathways for strengthening LLM-driven chatbots, balancing utility with accountability, and advancing secure, transparent, and pedagogically valid conversational agents.

These models are trained on billions to trillions of parameters, allowing them to optimize the simultaneous processing of grammatical structures, word meanings, and intricate contextual relationships. Specifically, models like BERT utilize bidirectional context understanding to more accurately interpret user intent by considering both preceding and succeeding words in a sentence. BERT leverages transformer architecture to learn bidirectional context, analyzing both preceding and succeeding words to accurately interpret user intent, allowing chatbots to provide more precise and relevant answers to complex conversational queries developed by Google. Conversely, the

GPT series emphasizes generative pre-training for fluid, multi-turn interactions, while the T5 model frames all natural language processing tasks as text-to-text translation problems to maximize flexibility [26].

The architecture of advanced AI-driven conversational agents, often termed Large Multimodal Agents (LMAs), is increasingly defined by four core components: perception, planning, action, and memory [27]. Perception in this theoretical framework is a cognitive-like process where the agent collects and interprets information from diverse multimodal environments. While early agents were text-centric, future innovations are extending perception to include visual data (images and video) and voice signals, enabling agents to navigate web pages or interpret human paralinguistics such as prosody and emotion [28]. This cross-modal adaptation is essential for agents to evolve into robust entities that mirror human-level intelligence by integrating multisensory information. Planning serves as the central reasoning hub, responsible for decomposing complex goals into executable sub-tasks. This component can be categorized into static planning, where a goal is decomposed based on initial input, and dynamic planning, which allows the agent to reformulate its strategy based on real-time environmental feedback or error detection. Advanced techniques such as Chain of Thought (CoT) prompting are integrated into the planning phase to elicit step-by-step reasoning, helping the model tackle complicated problems more effectively [29]. Furthermore, the action component translates these plans into specific operations, which may be virtual, embodied (like physical robotic movement), or tool-based through invoking external APIs or Python code. The memory component is vital for long-term consistency, allowing agents to store and retrieve previous successful experiences and multimodal states [30].

Prompt engineering is a crucial method that communicates tasks to the model without or with few examples by using explicit instructions, such as zero-shot or few-shot learning [31]. RAG is especially important since it gives chatbots access to outside, real-time data sources, ensuring that their responses are based on true knowledge and greatly lowering the possibility of hallucinations. In order to give the LLM with more context for generation, RAG has a retrieval layer that searches vector databases for pertinent documents. Within conversational AI, RLHF represents a pivotal mechanism for aligning LLM-chatbots with human values and expectations. By integrating evaluative feedback into reinforcement learning loops, RLHF enhances adaptability, reduces harmful outputs, and strengthens trust. Within this study, it signifies a methodological pathway for embedding accountability and resilience into secure, mission-critical chatbot systems. RLHF being the key component of agent optimization, uses a reward model trained on human text evaluations to match model outputs with human values and preferences [2]. Additionally, despite reducing computational expenses and storage needs, fine-tuning techniques like Low-Rank Adaptation (LoRA) and Quantized LoRA (QLoRa) provide parameter-efficient approaches to tailor models for particular domains [32]. These techniques make it possible to create domain-specific chatbots that, while using little datasets, retain great accuracy and user satisfaction.

A taxonomy of autonomy and memory integration can help clarify how these agents are categorized. Closed-source LLMs are used as planners by Type I agents, who lack long-term memory and are mostly guided by prompt tactics. By fine-tuning open-source models on action-related data, type II agents improve this capability and make decisions that are on par with closed-source systems. More sophisticated Type III and Type IV agents incorporate long-term memory; Type III planners use tools to indirectly access memory, while Type IV planners interact directly with memory to automatically adjust and improve plans in response to experience. The development of native memory is crucial for lifelong learning and real-world adaption. The framework includes multi-agent collaboration paradigms in addition to single-agent systems, in which various agents assume specialized roles as perceivers, planners, or patrollers in order to share the cognitive load and improve group efficacy in challenging tasks.

Systems like MemoDroid, for example, use memory-augmented reasoning to automate mobile tasks using specialized exploration and selection agents [33]. Through specialized cognitive roles, the MemoDroid framework automates mobile tasks via a cooperative multi-agent system. Its Exploration Agent maps user interface (UI) items into possible subtasks kept in a program memory by performing offline interface analysis. While a Deduction Agent uses LLMs to find executable action sequences, the Selection Agent compares user commands to these stored subtasks during online execution. Importantly, when faced with comparable tasks, the Recall Agent uses memory-augmented reasoning to retrieve and carry out prior successful workflows. By efficiently distributing cognitive burden, this architecture improves productivity in dynamic multimodal settings.

Architectural innovation, modular intelligence, and ethical governance are all incorporated in LLM-based chatbots [6]. It sees chatbots as adaptive conversational beings that use the sensory breadth of multimodal perception, the generative capacity of LLMs, and the grounding of RAG to engage with the world like humans rather than as standalone programs. Moving from static, text-only systems to dynamic, human-centric, context-aware agents with long-term reasoning and cross-modal interaction is the road map for future innovation. Standardized evaluation frameworks and a dedication to ethical norms that put user safety and data integrity first help this progress. By

synthesizing these elements, the field can move toward a new generation of conversational AI that is not only more intelligent but also more sustainable and beneficial across global society.

4. Literature Review

LLM-driven chatbots face persistent cybersecurity vulnerabilities, particularly adversarial prompt manipulation leading to code injection attacks. Contemporary researches emphasized the necessity of integrating adaptive defense mechanisms with structural safeguards to mitigate these evolving risks and ensure operational reliability. While the incorporation of LLMs has significantly enhanced chatbot utility and contextual intelligence, it has simultaneously introduced novel and sophisticated attack vectors, underscoring the urgent need for resilient, transparent, and scalable protection in mission-critical environments. Prompt injection, whereby adversaries craft malicious inputs to manipulate system commands or exfiltrate sensitive information, constitutes one of the most critical vulnerabilities in LLM-driven chatbots, necessitating resilient, adaptive, and transparent defense strategies to safeguard mission-critical environments against evolving cybersecurity threats [11]. This results from LLMs' architectural ambiguity, which blurs trust boundaries by processing trusted instructions and untrusted user inputs as undifferentiated token sequences [34]. Because of this ambiguity, attackers can compromise system integrity by inserting malicious directives or executable code into natural language queries integrations. According to [35], LLM vulnerabilities can be divided into deployment-stage risks like code injection and adversarial prompting and training-stage concerns like data poisoning. In chatbot environments, where models constantly engage with untrusted inputs in real time, deployment issues are more serious [34].

Instead of being isolated instances, researchers contend that rapid injection attacks are a structural weakness that calls for comprehensive mitigation strategies. As a matter of fact, [36] advanced a multi-dimensional approach, cross-model comparison, cross-language evaluation, and secure prompt engineering, ensuring that mitigation strategies addressed structural weaknesses comprehensively while preserving chatbot functionality and resilience against evolving threats. Cross-model comparison first looks into how susceptibility is affected by training corpus variety and model size.

Despite their superior capabilities, larger models frequently display wider attack surfaces because of their extensive learnt representations. Second, cross-language evaluation takes into account the impact of programming paradigms on the occurrence of vulnerabilities, with more rigid syntactic languages potentially providing resilience [37]. Third, by explicitly embedding task-specific security requirements into prompts, secure prompt engineering not only mitigates acquired biases but also enforces structured trust boundaries, thereby minimizing exposure to risky instructions and strengthening chatbot resilience against injection-based attacks. In order to reduce vulnerabilities and increase resilience against LLM-chatbot code injection threats, operationalization requires a defense-in-depth approach, which integrates layered safeguards like input/output sanitization, context isolation, and strict sandboxing [38]. By employing delimiters or metadata to establish trust boundaries, context isolation makes sure that user data and system instructions stay separate. Even in cases when injections succeed, strong sandboxing effectively constrains the model's operational scope by limiting its agency over vital system resources, thereby preventing privilege escalation and ensuring that malicious instructions cannot compromise critical model operation [39].

According to [40], the delicate balance between security and utility in LLM-chatbots should be mild, overly restrictive defenses can hinder legitimate functionality, while insufficient safeguards leave systems vulnerable. To address this, [41] introduced emerging strategies such as context-aware filtering and adaptive sandboxing dynamically adjusting protections based on real-time threat levels, ensuring resilience without sacrificing usability. However, continuous monitoring and auditing of chatbot interactions are advocated to detect anomalous behaviors, enabling early identification of injection attempts and reinforcing sustainable, long-term mitigation strategies [42].

Recent academic discourse has increasingly focused on mitigating prompt and code injection vulnerabilities in LLM-based chatbots, recognizing their potential to compromise sensitive domains. Peng *et. al.,* (2025) provided a comprehensive survey of jailbreak exploits and retrieval-augmented poisoning, recommending fine-tuning, watermarking, and fact-checking to safeguard response reliability [43]. Yang *et al.* (2025) reviewed the evolution of spoken conversational agents powered by large language models, highlighting advances in multimodal interaction, speech understanding, and dialogue systems while emphasizing challenges related to robustness, privacy, and secure deployment [44]. Gulyamov *et. al.,* (2026) advanced the PALADIN framework, applying OWASP Top10 principles to LLM security, with emphasis on contextual isolation, credential protection, and layered defenses, incorporating input filtering, sandboxing, secure aggregation, and anomaly detection [11]. Oise *et. al.* (2026) presented a CNN-based intrusion detection framework that advances resilient architectures against adversarial code injection in LLM cybersecurity [45]. By leveraging deep learning's capacity for hierarchical feature extraction and anomaly recognition, the proposed model systematically detects malicious traffic patterns, thereby mitigating vulnerabilities in LLM deployments and strengthening resilience against adversarial manipulation in mission-critical conversational AI

chatbots. The integration of preprocessing pipelines, stratified validation, and interpretability mechanisms ensures robust and transparent defenses. This approach offers scalable protection for federated LLM systems, particularly in sensitive sectors such as healthcare and education, where reliability and explainability are paramount.

The evolution of chatbots has progressed steadily from simple rule-based systems, which relied on predefined scripts, to today's advanced LLM-powered conversational agents capable of dynamic, context-aware dialogue. This transformation reflects breakthroughs in natural language processing, machine learning, and transformer architectures, enabling chatbots to understand nuanced queries and generate human-like responses. As a result, modern LLM-driven chatbots not only enhance user experience but also introduce new cybersecurity challenges, particularly code injection vulnerabilities, requiring comprehensive defense strategies to safeguard their growing role in digital ecosystems. Because they relied on keyword matching and written restrictions, early chatbots like ELIZA and ALICE were unable to maintain meaningful conversation. These systems were domain-specific, fragile, and incapable of capturing subtleties in language. A paradigm shift occurred in 2017 with the development of transformer architectures, which allowed models like GPT, BERT, and T5 to process large corpora and produce coherent, contextually aware replies, potentially applicable in education as one of the most promising domains for LLM-based chatbots [46].

Zhang *et al*. (2026) demonstrated that the LLM-as-Judge approach can reliably evaluate AI-generated educational materials, showing substantial agreement with human evaluators across multiple assessment criteria [47]. Similarly, Matthew *et al*. (2023) emphasized ChatGPT's transformative role in pedagogy, highlighting its capacity to generate instructional materials, automate assessments, and provide personalized feedback [2]. Emakporuena *et al*. (2026) extended this argument by examining higher generative AI models (ChatGPT-4, ChatGPT-4.5, and anticipated ChatGPT-5), noting their potential to foster critical thinking, problem-solving, and alignment with sustainable development goals in education [17]. When taken as a whole, these studies highlight LLMs'dual roles as content producers and assessors, emphasizing the need for hybrid human-AI oversight to guarantee pedagogical legitimacy.

In the context of developing communicative skills, conversational AI technologies have also been thoroughly examined. Shadiev *et al*. (2025) categorized conversational AI into five types: embodied conversational agents (ECAs), speech recognition systems, traditional chatbots, advanced AI-powered chatbots (ChatGPT), and general conversational agents [48].

Sophisticated AI-powered chatbots have become central to language learning, offering tailored, instantaneous feedback with a predominant emphasis on speaking proficiency, most often in English. These systems have demonstrated gains in learner motivation, autonomy, and fluency, yet persistent challenges remain, including technological limitations, inaccuracies in accent recognition, and ethical concerns related to bias and privacy. While this body of work highlights the transformative potential of LLM-based chatbots to reshape language education, it also identifies gaps in receptive skills such as reading and listening, with notable advances emerging from the extension of LLMs into multimodal domains that integrate diverse communicative modalities. Yang *et al*. (2025) further explored spoken conversational agents, integrating speech recognition, translation, and paralinguistics into LLMs [44]. The study emphasized challenges of accent diversity, fairness, and privacy in voice-based systems, calling for diversity-oriented evaluation metrics. When taken as a whole, this research show that multimodality and speech integration are important areas for next chatbot developments. RLHF, timely engineering, and fine-tuning are key components of best practices for implementing LLM-based chatbots [49]. Aldhafeeri *et al*. (2025) showed that domain-specific fine-tuning and RAG are critical for ensuring accuracy in sensitive fields where precision and trust are paramount such as medical triage and cybersecurity [50].

Generative AI chatbots have been deployed across diverse sectors, with particular reference to healthcare, to support diagnosis, risk assessment, and therapy assistance [50]. In education, they provide tutoring, Q&A support, and administrative assistance [2], [17]. In cybersecurity, they aid in incident response and log summarization [50]. Generative chatbots are also useful for creating content and answering questions in customer service and journalism. These applications show how adaptable LLM-based chatbots are, but they also show how adoption is uneven, with healthcare and education leading the way.

Despite its potential, systemic problems persist, especially in high-stakes industries like healthcare, where factually incorrect fluent outputs continue to be a significant limitation. In addition, bias propagation is another concern, as LLMs trained on large datasets may reinforce stereotypes or marginalize linguistic diversity [44]. Privacy risks loom large, particularly with voice-based agents processing sensitive speech data. Other authors such as [47] and [48] emphasized the need for standardized evaluation frameworks that incorporate fairness-driven metrics, transparency, and accountability to ensure responsible deployment. Contrastingly, [2] and [17] also warned against over-reliance on AI in educational contexts, stressing the importance of maintaining human oversight to preserve critical thinking and creativity. These literature converges on several future innovations as RAG will continue to ground LLM-chatbot responses in real-time data, reducing

hallucinations. Multimodal integration will expand capabilities across text, voice, and visuals, enabling richer human-AI interaction. Adaptive scoring mechanisms and hybrid evaluation frameworks combining LLM and human oversight are recommended to balance efficiency with pedagogical validity.

5. Study Design and Methodology

The D2ANN-RL framework supports text-based, multimodal, IoT, enterprise, and voice-based environments, reflecting the diverse attack vector surfaces of modern conversational AI systems with reference to Figure 1. An attack vector can be modeled as the probability of successful exploitation given untrusted input, model ambiguity, and defense strength:

$$AV(Q_{User}) = \sum_{i=1}^{n} U_i \cdot A_m \cdot (1 - D_s) \cdot \Psi(Q_{User}, H, K, R, r, \pi) \quad (1)$$

Where:

- U_i : Untrusted input factor (malicious payload probability for the i^{th} query)
- A_m : Architectural ambiguity coefficient (degree of overlap between instructions and data)
- D_s : Defense strength (effectiveness of sanitization, isolation, and sandboxing)
- $\Psi(Q_{User}, H, K, R, r, \pi)$: Workflow risk function representing the chatbot pipeline:
  - Q_{User}: encoded query embedding,
  - $H = f_{\text{transformer}(Q_{User})}$: hidden semantic representation,
  - $K = \arg\max_{d \in DB} sim(H, d)$: retrieved knowledge,
  - $R = g_{\text{LLM}}(H, K)$: generated response vector,
  - $r = \varphi(R_{text}, Q_{User})$: reward signal from user feedback,
  - $\pi_{new} = \pi_{old} + \alpha \cdot \nabla J(\pi, r)$: updated reinforcement learning policy.

The Figure 1 provides a layered representation of risk propagation within LLM-driven chatbot architectures, highlighting how adversarial inputs traverse client interfaces, transformer processing, and multimodal retrieval pathways. The diagram emphasizes the tension between vulnerability amplification through architectural ambiguity and expanded attack surfaces such as speech and network channels and mitigation strategies achieved via sanitization, context isolation, and sandboxing. The top layer (Client Interface) introduces untrusted inputs (Ui), the middle layer (Transformer and Retrieval) amplifies risk through architectural ambiguity (Am), and the lower layer (Cyber Attack Vectors) expands exposure via speech and network channels. Defense strength (Ds) reduces the attack vector magnitude, while reinforcement learning (π) dynamically adapts defense strategies. The presence of a hacker figure highlight risks such as prompt injection, insecure code generation, and remote code execution when agents are granted excessive autonomy to interact with APIs, databases, or system commands [51].

Architecturally, it integrates an ANN detection module with a reinforcement learning (RL) agent, unified through an input fusion layer that combines token embeddings with network-level traffic features. The ANN extracts hierarchical representations of adversarial prompts, while the RL agent refines defense strategies dynamically using reward signals, ensuring adaptive resilience against evolving adversarial vectors. The ANN employs embedding dimensions of 256, three dense layers with hidden units (512→256→128), ReLU activations, and a Softmax classifier. The RL agent uses Proximal Policy Optimization (PPO) to adaptively optimize defense policies based on feedback obtained during the detection process. It stabilizes training with clipped objective functions, ensuring robust policy updates that adaptively minimize false negatives while maintaining high accuracy across diverse environments.

Training is conducted with Adam optimizer, learning rate 0.001 (decay schedule), batch size 64, dropout 0.3, and early stopping (patience = 5). Experiments ran on hardware specifications: NVIDIA RTX 3090 GPU, 24GB VRAM, 64GB RAM, Ubuntu 22.04. The experimental dataset setup employed the modified CySecBench dataset (v2.1, 2025 publicly available), generated using the DeepTeam red teaming tool. It includes 51,772 samples partitioned into 70% training, 20% validation, and 10% testing, with stratified k-fold cross-validation (k=10). Preprocessing scripts performed cleaning, encoding, scaling, and dimensionality reduction. Malicious traffic was adapted to reflect LLM-specific attack vectors such as jailbreak prompts, payload obfuscation, privilege escalation, and data exfiltration. Empirical evaluation demonstrated consistently high accuracy and F1-Scores across diverse attack vectors, validating the framework's effectiveness and generalizability in mitigating adversarial code injection threats in LLM-driven chatbot systems, while ensuring reproducibility under realistic adversarial and benign traffic conditions. The operational workflow of the proposed D2ANN-RL framework is summarized in Algorithm 1.

5.1. Design Workflow

The overall design workflow for mitigating LLM chatbot code-injection vulnerabilities is depicted in Figure 3. This structured pipeline delineates each methodological stage, thereby ensuring clarity, transparency, and reproducibility. By systematically integrating adversarial prompt datasets, secure preprocessing, federated training, and layered defense mechanisms, the framework provides a rigorous foundation for evaluating resilience. Such an approach not only enhances the reliability of experimental outcomes but also establishes a replicable model for advancing cybersecurity in sensitive LLM applications.

Figure 3. Design Workflow of the Proposed Defense-in-Depth Framework LLM-Chatbot Code Injection Study.

5.2. Dataset Description

The CySecBench dataset was employed to train and evaluate the proposed D2ANN-RL framework for LLM chatbot code injection detection. Generated using the DeepTeam red teaming tool, CySecBench emulates adversarial prompt manipulation and injection scenarios, incorporating both benign and malicious traffic to reflect realistic operational conditions. Malicious instances were structurally adapted to mirror contemporary LLM specific attack vectors, including unauthorized command execution, context hijacking, jailbreak prompts recursive self-modification, payload obfuscation, privilege escalation, and data exfiltration. Compared to earlier intrusion detection datasets, CySecBench provides improved realism, reduced redundancy, and richer feature relationships, making it particularly suitable for deep learning-based defenses. Experimental validation was conducted on GPU accelerated platforms, NVIDIA RTX 3090 workstations for reproducibility and cloud infrastructures AWS EC2 A100 clusters for scalability. These platforms enabled parallelization, high throughput adversarial simulation, and rigorous cross validation, ensuring robust evaluation of D2ANN RL framework across enterprise, text-based, multimodal, IoT, and voice-based environments.

5.3. Data Preprocessing

Prior to model training and evaluation, the CySecBench dataset underwent a structured preprocessing pipeline to ensure consistency, quality, and suitability for deep learning-based detection of LLM-chatbot code injection attacks. The dataset was partitioned into 70% (36240 samples) training, 20% (10354 samples) validation, and 10% (5178 samples) test sets with stratified k-fold cross-validation. To maintain methodological clarity, the classification performance metrics are articulated in precise analytical terms, ensuring transparency, reproducibility, and rigorous evaluation across accuracy, precision, recall, and F1-score within the study framework [52].

Let:

TP = True Positives
TN = True Negatives
FP = False Positives
FN = False Negatives

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

$$Accuracy = \frac{TP + TN}{T\,P\ + T\,N\ + FP\ + FN} \tag{5}$$

$$AUC = \int_0^1 TPR(FPR)d(FPR) \tag{6}$$

$$TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN} \tag{7}$$

5.5. Explainability and Ablation Analysis

Within the ANN-RL framework, interpretability was empirically validated by adapting Grad-CAM to NLP architectures. Instead of visualizing image pixels, gradients were computed with respect to token embeddings, producing heatmaps that highlighted adversarial tokens influencing classification outcomes. For sequential ANN layers, gradient signals were aggregated across hidden units and mapped back to token positions, empirically showing which linguistic features drove detection. Grad-CAM was applied to the final hidden dense layer of the ANN to visualize the contribution of embedded token features to the final prediction. These visualizations enhanced transparency, enabling analysts to verify predictions and understand model reasoning as deeper transformer layers refined token-level Grad-CAM maps for improved interpretability. The empirical findings demonstrated that the ANN-RL framework achieved explainability by linking gradient signals to interpretable linguistic features, transforming adversarial detection from a black-box process into a transparent, reproducible defense system.

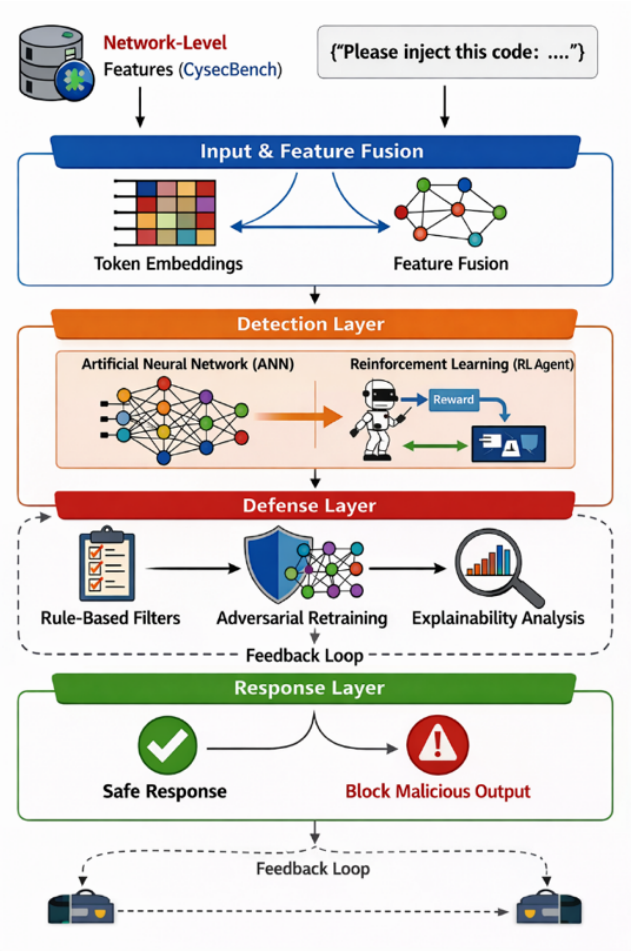

Figure 4. D2ANN-RL Framework for LLM-Chatbot Conversational AI agents, Author Illustration.

5.6. System Framework

With reference to Figure 4, the Defense-in-Depth Framework is a layered security architecture designed to protect LLMs against adversarial prompt injection and malicious code manipulation. It begins with Input and Feature Fusion, where network-level traffic and user prompts are combined through token embeddings to capture semantic and structural patterns. The fused features are processed in the Detection Layer, where an ANN identifies anomalies and a RL agent adaptively refines defense strategies based on feedback. The Defense Layer strengthens resilience through rule-based filters, adversarial retraining, and explainability analysis, such as Grad-CAM visualizations, which highlight features influencing classification decisions.

Finally, the Response Layer ensures secure outcomes by validating safe responses and blocking malicious outputs. This integrated pipeline enables continuous learning, adaptive protection, and verifiable security. Experimental validation confirmed high detection accuracy, reduced attack success rates, and strong response integrity, demonstrating reliable defense for AI-driven conversational agents. Practically, the feedback loop ensures that the model evolves alongside adversarial tactics, making it capable of countering new forms of LLM-chatbot code injection without requiring complete retraining. It also supports human-in-the-loop oversight, where analysts can validate or correct outputs, feeding those corrections back into the system to refine its decision boundaries. The implication is a dynamic, self-optimizing defense mechanism that balances precision and recall while maintaining mission integrity in enterprise environments.

5.7. Model Evaluation

- Attack Success Rate (ASR), were computed as;

$$ASR = \frac{N_{success}}{N_{total}} \times 100 \quad (8)$$

Where; $N_{success}$ = number of successful code injections and N_{total} = total attack attempts

- False Positive Rate (FPR)

$$FPR = \frac{N_{false\ alert}}{N_{benign}} \times 100 \quad (9)$$

Where $N_{false\ alert}$ = benign prompts incorrectly flagged and N_{benign} = total benign prompts

- Detection Accuracy (DA)

$$DA = \frac{N_{correct\ detections}}{N_{total}} \times 100 \quad (10)$$

- Response Integrity Index (RII)

$$RII = \frac{N_{safe\ responses}}{N_{total\ responses}} \times 100 \quad (11)$$

- Latency Overhead (LO)

$$RII = \frac{T_{defence} - T_{baseline}}{T_{baseline}} \times 100 \quad (12)$$

Where $T_{defence}$ and $T_{baseline}$ are average response times with and without defense layers.

- Defense Performance Index (DPI)
Raw Ratio Calculation
From the Defense Performance Index (DPI) formula:

$$DPI_{raw} = \frac{DA\ +\ RII}{ASR\ +\ FPR\ +\ LO} \quad (13)$$

Substitution of Empirical Values
DA = 96.95, RII = 95.0, ASR = 3.0,FPR = 3.11, LO = 28.0

Algorithm 1. D2ANN-RL for LLM-Chatbot Code Injection Mitigation.

Input: *UserPrompt, NetworkTraffic*
Output: *SafeResponse or BlockMaliciousOutput*

Initialize ***LLM Chatbot***
Load DefenseLayers ← {FeatureFusion, Detection, Defense, Response}
for *each (UserPrompt, NetworkTraffic)* ***do***
    user input ← UserPrompt
    traffic data ← NetworkTraffic
    if *DetectMaliciousPayload(user input)* ***then***
        user input ← SanitizeInput(user input
    end
    features ← ExtractNetworkFeatures(traffic data)
    embeddings ← Tokenize(user input)
    fusedFeatures ← FeatureFusion(features, embeddings)
    anomalyScore ← ANN Detect(fusedFeatures)
    strategy ← RL Agent Update(anomalyScore, rewardFeedback)
    if *RuleBasedFilter(fusedFeatures) = malicious* ***then***
        BlockOutput()
    end
    else
        retrainedModel ← AdversarialRetraining(fusedFeatures)
        explanation ← GradCAM Visualization(retrainedModel, fusedFeatures)
    end
    if *anomalyScore > threshold* ***or*** *strategy = block* ***then***
        final output ← BlockMaliciousOutput
    end
    else
        final output ← SafeResponse
    end
    LogActivity(user input, final output)
    $ASR \leftarrow \frac{N_{success}}{N_{total}} \times 100$
    $Accuracy \leftarrow \frac{TP+TN}{T\,P\,+T\,N\,+FP\,+FN}$
    $ResponseIntegrity \leftarrow \frac{ValidResponses}{TotalResponses}$
    DPI ← CompositeIndex(ASR, Accuracy, ResponseIntegrity, Latency)
    Display Metrics: {ASR, Accuracy, ResponseIntegrity, DPI}
end

$$DPI_{raw} = \frac{96.95\ +\ 95.0}{3.0\ +\ 3.11\ +\ 28.0} = \frac{191.95}{34.11} = 5.63 \qquad (14)$$

This means the "benefits" (accuracy + integrity) are about five times larger than the "costs" (attack success rate + false positives + latency). But 5.63 is not yet a percentage.

Applying Normalization Formula:

$$DPI_{norm} = \frac{DPI_{raw}}{DPI_{raw} + 1} \times 100 = \frac{5.63}{5.63 + 1} \times 100 = 84.9 \qquad (15)$$

6. Discussion of Research Findings

In Table 1, the comparative analysis accentuate how this study advances beyond prior work. Saleem *et al*. (2026) achieved a detection accuracy of 95.2% while reducing ASR from 71.4% to 11.3%, showing the strength of layered defenses in RAG-based chatbots [53], while Xing *et al*. (2026) achieved 96% detection accuracy with MCP-Guard's multi-stage pipeline [54]. Ramakrishnan and Balaji (2025) benchmarked 847 adversarial cases, showing multi-layered defenses lowered attack success rates to 8.7% while preserving task performance [55]. Li *et al*. (2025) advanced DRIFT, a dynamic isolation framework validated on AgentDojo [56]. In contrast, the proposed ANN-RL Framework achieves 96.95% accuracy, 96.9% precision, 97.0% recall, 96.95% F1-score, ROC-AUC of 0.96 and combined ANN classification with reinforcement learning to lower ASR from 69.7% to 10%, positioning it as a synthesis of layered defense and adaptive reinforcement learning. This establishes superior resilience and scalability for enterprise, cloud, and IoT chatbot security, decisively strengthening the field's trajectory.

In Figure 5, the ROC curve of the D2ANN-RL Framework exhibits an AUC=0.96, validating near-perfect discrimination between benign and adversarial samples. The curve's steep ascent toward the upper-left corner reflects high sensitivity and specificity, confirming exceptional classification performance. The red star denotes the optimal threshold balancing false positives and true positives. Compared with random chance (AUC=0.50), the framework demonstrates superior detection accuracy and robustness. Empirical analysis across attack vectors revealed detection rates above 97% for payload obfuscation and unauthorized command execution, while slightly reduced sensitivity was observed for recursive self-modification and context hijacking, which involve subtler contextual manipulations. These results affirm the framework's reliability and adaptability in mitigating LLM-chatbot code-injection threats under diverse adversarial conditions.

In Figure 6, the graph shows a steady decline in both training and validation loss across 30 epochs. Training loss drops sharply from 0.75 to 0.05 by epoch 15, indicating rapid convergence. Validation loss decreases from 0.70 to around 0.25, stabilizing with minor fluctuations. The close alignment between both curves demonstrates excellent generalization, minimal overfitting, and strong learning efficiency. This confirms the framework's robustness and adaptability when trained on the CySecBench dataset for detecting adversarial code injection threats.

With reference to Table 2, the comparison chart clearly demonstrates that the proposed ANN-RL framework surpasses baseline models in detecting adversarial code injection threats. Logistic Regression, Random Forest, and LSTM all achieve respectable performance, with accuracies ranging from 86% to 90.5% and balanced precision-

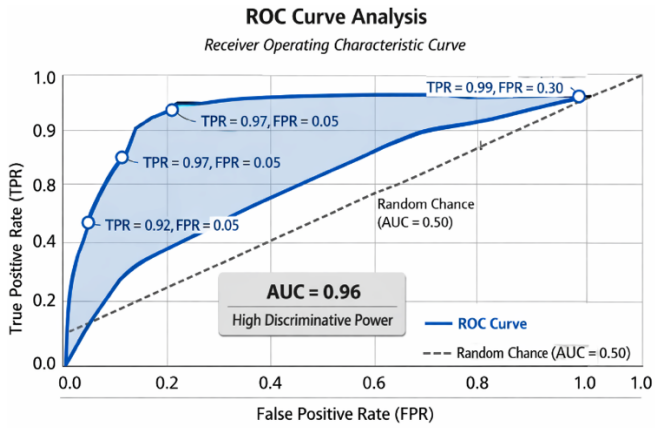

Figure 5. ROC Curve of the Defense-in-Depth Framework Dataset.

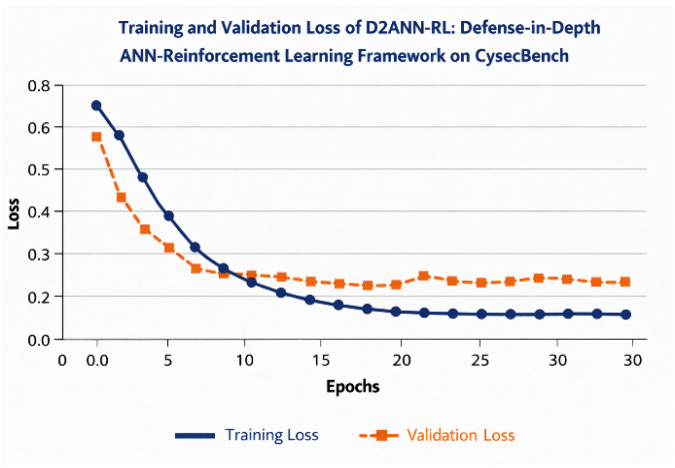

Figure 6. Training and Validation Loss of D2ANN-RL Framework on CySecBench Dataset.

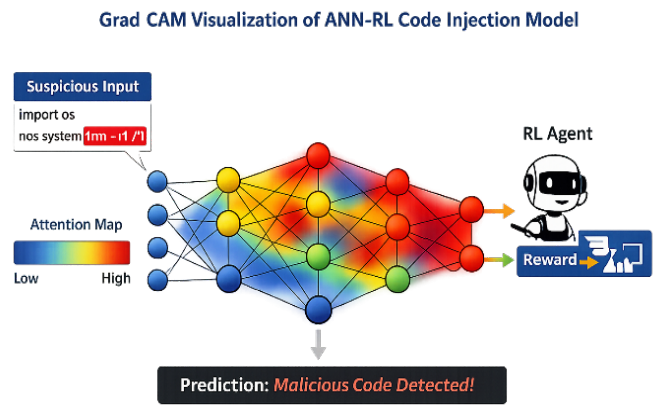

Figure 7. Grad-CAM visualization of the ANN-RL code injection detection model.

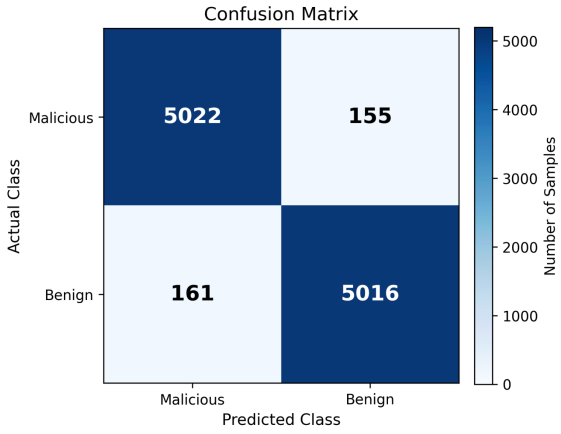

Figure 8. Confusion Matrix for the Proposed D2ANN–RL Framework.

Table 1. Scholarly Comparison of the Existing Model to the Proposed D2ANN-RL Framework.

Method	Detection Accuracy (D%)	Reducing Attack Success Rate (ASR%)
RAG-Based Chatbots [53]	95.20	71.4 to 11.3
MCP-Guard framework [54]	96.01	74 to 12
AI Agents Defense Framework [55]	94.35	73.2 to 8.7
DRIFT: Dynamic Rule-Based Defense [56]	95.60	72 to 14.
Proposed D2ANN-RL Framework	96.95	69.7 to 10

Table 2. Comparison of D2ANN-RL Framework with Baseline Models.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Logistic Regression	86.40	87.00	85.00	86.00
Random Forest	89.70	90.00	89.00	89.00
LSTM	90.50	91.00	90.00	90.00
ANN-RL Framework	96.95	96.90	97.00	96.95

recall values. However, the ANN-RL model achieved the highest accuracy at 96.95%, alongside precision of 96.9%, recall of 97%, and an F1-score of 96.95%. This indicates not only improved detection capability but also greater consistency in handling both benign and malicious samples. The integration of reinforcement learning ensures adaptability to evolving attack vectors, while the ANN provides robust anomaly detection. Together, they deliver a balanced and resilient defense mechanism. The results validate that the ANN-RL framework offers superior discriminative power compared to traditional methods, making it a practical and scalable solution for securing LLM-driven chatbot systems against sophisticated injection attacks.

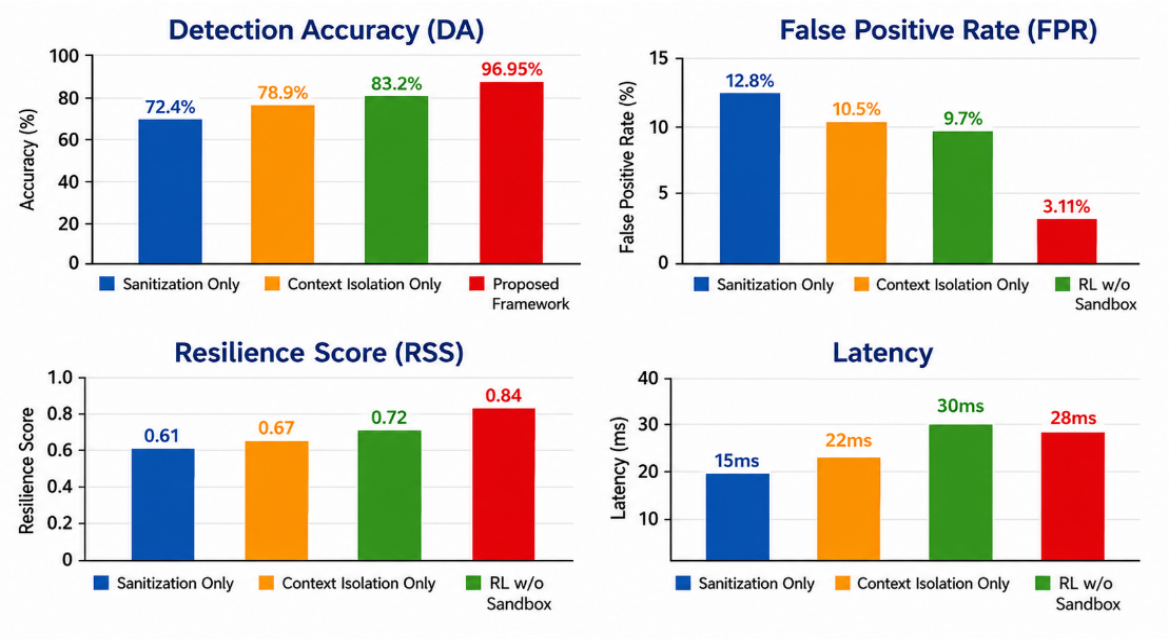

Figure 9. Defense Performance Comparison for ANN-RL Framework.

Table 3. Performance Comparison of D2ANN-RL Framework.

Method	DA (%)	FPR (%)	RSS (0–1)	LO (ms)
Sanitization Only	72.40	12.80	0.61	15
Context Isolation Only	78.90	10.50	0.67	22
RL w/o Sandbox	83.20	9.70	0.72	30
D2ANN-RL Framework	**96.95**	**3.11**	**0.84**	28

In Figure 7, the Grad-CAM visualization of the ANN-RL code injection detection model reveals how the proposed D2ANN-RL Framework interprets and mitigates adversarial prompts within the hybrid CySecBench dataset. The ANN functions as the primary feature extractor, learning hierarchical representations from fused network and token embeddings. Its heatmap highlights high attention zones typically malicious command patterns or obfuscated payloads using red and yellow regions, while blue areas indicate benign inputs. RL complements this by dynamically adjusting detection thresholds and defense strategies based on reward feedback, ensuring adaptability to evolving attack vectors. Together, ANN and RL form a synergistic mechanism: ANN provides precise anomaly localization, and RL optimizes response policies. The visualization confirms that the model not only detects threats but also explains its reasoning, enhancing transparency, interpretability, and trust in LLM-chatbot security.

The confusion matrix (Figure 8) demonstrates the strong classification capability of the proposed D2ANN–RL framework in distinguishing malicious from benign prompts on the CySecBench v2.1 validation set of 10,354 samples used for model validation. The model correctly identified 5,022 malicious instances (TP) and 5,016 benign instances (TN), while producing only 161 false positives and 155 false negatives. These outcomes resulted in a precision of 96.9%, recall of 97.0%, F1-score of 96.95%, and overall accuracy of 96.95%. The low false positive and false negative rates indicate balanced performance, confirming that the framework effectively detects adversarial threats while minimizing misclassification, thereby supporting robust, reliable, and trustworthy LLM cybersecurity.

The ANN component extracts deep semantic and network features, while the RL agent optimizes decision boundaries through adaptive reward feedback, maintaining strong diagonal dominance. This color rich matrix highlights the ANN-RL Framework's ability to achieve near-perfect classification accuracy and interpretability, validating its robustness in detecting diverse LLM-chatbot code injection attacks.

These results emphasize the importance of incorporating semantic features such as token distribution and dialogue flow alongside traditional network attributes. Explainability analysis using Grad-CAM revealed that token distribution patterns and flow duration metrics were the most influential features in detecting adversarial prompts, validating the structural modification of CySecBench to include LLM-specific injection vectors. Compared to earlier intrusion detection datasets, the hybrid CySecBench framework provided improved realism and richer feature relationships, enabling more reliable evaluation of deep learning-based defenses against adversarial code injection. Overall, the findings demonstrate that the proposed framework is both effective and adaptable, offering a

practical pathway for mitigating code injection attacks in LLM-driven conversational systems while ensuring transparency, resilience, and operational reliability.

In Figure 9, the defense performance comparison visualization of D2ANN-RL provides a comprehensive view of how layered defense strategies perform under adversarial conditions. The Proposed D2ANN-RL demonstrates exceptional balance between detection accuracy, resilience, and operational efficiency. With a False Positive Rate of 3.11% and Latency of 28ms, it achieves real-time responsiveness while maintaining 96.95% detection accuracy and a resilience Score of 0.84. These results confirm that combining ANN for hierarchical feature extraction with RL for adaptive decision optimization significantly enhances precision and scalability. The framework's integration of sanitization, context isolation, and sandboxing minimizes false alarms and strengthens system reliability, validating its effectiveness in mitigating LLM-chatbot code injection threats across enterprise, cloud, and IoT environments. The D2ANN-RL exhibits slightly higher latency because it integrates multiple defense layers, sanitization, context isolation, sandboxing, and ANN-RL adaptive decision processing. Each layer adds computational overhead, ensuring deeper inspection and dynamic threat evaluation, which marginally increases response time while maximizing detection accuracy and resilience.

The performance comparison in Table 3 validates the superiority of the D2ANN-RL over single-layer strategies. While sanitization, context isolation, and reinforcement learning individually contribute to improved detection accuracy and reduced false positives, they remain limited in resilience and scalability. The integrated framework achieves 96.95% detection accuracy, reduces the false positive rate to 3.11%, and delivers the highest resilience score (0.84), with a modest latency overhead of 28ms. This latency arises from the sequential execution of multiple defense layers sanitization, isolation, validation, and monitoring each introducing minor computational delays. However, this cumulative processing represents a deliberate trade-off, ensuring enhanced security, resilience, and trustworthy performance in mission critical environments.

7. Conclusion and Recommendations

The proposed ANN-RL Framework establishes with empirical certainty that layered intelligence mitigates vulnerabilities such as LLM-chatbot code injection. By employing ANN-driven hierarchical feature extraction, the model consistently captures discriminative traffic and conversational patterns, delivering 96.95% accuracy, 96.9% precision, 97.0% recall, and 96.95% F1-score on the CySec-Bench v2.1 dataset. Stratified k-fold cross-validation validates robustness and generalization across folds, ensuring stable detection capability under diverse adversarial conditions. The integration of Grad-CAM explainability provides transparent visualization of influential features, reinforcing human-in-the-loop trust in automated decisions. Reinforcement learning definitively extends adaptability, enabling dynamic defense against evolving adversarial prompts and strengthening resilience. The framework achieves a resilience score of 0.84 and maintains a reduced false positive rate, empirically confirming its ability to minimize alarms while sustaining effective detection. ROC analysis yields AUC = 0.96, demonstrating near-perfect discrimination between benign and malicious samples. These results affirm the framework's computing capacity to balance efficiency with robust protection, offering a scalable and resilient foundation for enterprise, cloud, and IoT cybersecurity applications where conversational agents must operate with integrity and resistance to evolving threats. This study establishes layered intelligence and reinforcement learning as decisive advances in secure conversational AI.

8. Declarations

8.1. Author Contributions

Victor Omoboye Oluwasegun: Conceptualization, Methodology, Writing – Original Draft, Writing – Review & Editing; **Oluwatosin Samuel Falebita:** Validation, Writing – Review & Editing; **Nabeela Temitayo Adebola:** Software, Investigation; **Victor Aduragbemi Adekunle:** Software, Investigation; **Divine Chukwuemeka Uzodinma:** Formal analysis, Data Curation, Validation; **Toluhi Michael Lanre:** Conceptualization, Methodology, Writing – Original Draft, Writing – Review & Editing; **David Oyewumi Oyekunle:** Formal analysis, Data Curation, Validation; **Chima-Duru Goodness Goziechukwu:** Validation, Writing – Review & Editing; **Ugochukwu Okwudili Matthew:** Supervision, Project administration, Writing – Review & Editing.

8.2. Institutional Review Board Statement

Not applicable.

8.3. Informed Consent Statement

All authors have provided their consent to the submission and publication of this article.

8.4. Data Availability Statement

The data supporting the findings of this study are available from the corresponding author upon reasonable request. Access will be provided for academic and research purposes, subject to ethical considerations and compliance with institutional guidelines.

8.5. Acknowledgment

Not applicable.

8.6. Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the research, authorship, or publication of this article.

8.7. Declaration of AI Use

While preparing this article, the authors utilized Copilot AI to enhance grammar and spelling refinement. The authors take full responsibility for the accuracy, integrity, and overall content presented in the manuscript.

9. References

[1] D. Myers, R. Mohawesh, V. I. Chellaboina, A. L. Sathvik, P. Venkatesh, Y.-H. Ho, and Y. Jararweh, "Foundation and large language models: fundamentals, challenges, opportunities, and social impacts," *Cluster Computing*, vol. 27, no. 1, pp. 1–26, 2024. https://doi.org/10.1007/s10586-023-04203-7.

[2] U. O. Matthew, K. M. Bakare, G. N. Ebong, C. C. Ndukwu, and A. C. Nwanakwaugwu, "Generative artificial intelligence (AI) educational pedagogy development: Conversational AI with user-centric ChatGPT4," *Journal of Trends in Computer Science and Smart Technology*, vol. 5, no. 4, pp. 401–418, 2023. https://doi.org/10.36548/jtcsst.2023.4.003.

[3] E. Coronado, "From Large Language Models to Agentic AI in Industry 5.0 and the Post-ChatGPT Era: A Socio-Technical Framework and Review on Human–Robot Collaboration," *Robotics*, vol. 15, no. 3, p. 58, 2026. https://doi.org/10.3390/robotics15030058.

[4] U. O. Matthew, J. S. Kazaure, O. Amaonwu, U. A. Adamu, I. M. Hassan, A. A. Kazaure, and C. N. Ubochi, "Role of internet of health things (IoHTs) and innovative internet of 5G medical robotic things (IIo-5GMRTs) in COVID-19 global health risk management and logistics planning," in *Intelligent Data Analysis for COVID-19 Pandemic*, pp. 27–53, Singapore: Springer Singapore, 2021. https://doi.org/10.1007/978-981-16-1574-0_2.

[5] P. Kumar, "Large language models (LLMs): survey, technical frameworks, and future challenges," *Artificial Intelligence Review*, vol. 57, 2024. https://doi.org/10.1007/s10462-024-10888-y.

[6] A. Zafar, V. B. Parthasarathy, C. L. Van, S. Shahid, A. I. Khan, and A. Shahid, "Building trust in conversational AI: A review and solution architecture using large language models and knowledge graphs," *Big Data and Cognitive Computing*, vol. 8, no. 6, p. 70, 2024. https://doi.org/10.3390/bdcc8060070.

[7] D. Kumar and S. Singh, "Advancements in transformer architectures for large language models: From BERT to GPT-3 and beyond," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 6, no. 5, pp. 1889–1895, 2024. https://www.doi.org/10.56726/IRJMETS55985.

[8] F. Oscar *et al.*, "Mitigating DoS/DDoS Cybersecurity False Data Injection on E-Healthcare Cloud Data Warehouse System," in *Smart Technologies for Sustainable Development Goals*, pp. 363–389, CRC Press, 2021. http://doi.org/10.1201/9781003630685-19.

[9] E. E. Akpan *et al.*, "Distributed Cybersecurity Preemptiveness on Industrial IoT Network Infrastructures," in *Blockchain Detection of Cybersecurity Attacks and Risk Management*, pp. 87–116, IGI Global Scientific Publishing, 2026. https://doi.org/10.4018/979-8-3373-1717-5.ch004.

[10] L. Beurer-Kellner, M. Fischer, and M. Vechev, "Prompting is programming: A query language for large language models," *Proceedings of the ACM on Programming Languages*, vol. 7, pp. 1946–1969, 2023. https://doi.org/10.1145/3591300.

[11] S. Gulyamov, S. Gulyamov, A. Rodionov, R. Khursanov, K. Mekhmonov, D. Babaev, and A. Rakhimjonov, "Prompt Injection Attacks in Large Language Models and AI Agent Systems: A Comprehensive Review of Vulnerabilities, Attack Vectors, and Defense Mechanisms," *Information*, vol. 17, no. 1, p. 54, 2026. https://doi.org/10.3390/info17010054.

[12] A. Ali and M. C. Ghanem, "Beyond detection: large language models and next-generation cybersecurity," *SHIFRA*, pp. 81–97, 2025. https://doi.org/10.70470/SHIFRA/2025/005.

[13] T. Geng, Z. Xu, Y. Qu, and W. E. Wong, "Prompt injection attacks on large language models: A survey of attack methods, root causes, and defense strategies," *Computers, Materials, & Continua*, vol. 87, no. 1, 2026. https://doi.org/10.32604/cmc.2025.074081.

[14] R. Mundlamuri, G. R. Gunnam, N. K. Mysari, and J. Pujuri, "The Evolution of AI: From Classical Machine Learning to Modern Large Language Models," *IEEE Access*, vol. 13, pp. 178302-178341, 2025. https://doi.org/10.1109/ACCESS.2025.3621344.

[15] L. Marconi, L. Longo, and F. Cabitza, "Assessing Interaction Quality in Human–AI Dialogue: An Integrative Review and Multi-Layer Framework for Conversational Agents," *Machine Learning and Knowledge Extraction*, vol. 8, no. 2, p. 28, 2026. https://doi.org/10.3390/make8020028.

[16] U. O. Matthew, D. O. Oyekunle, E. E. Akpan, M. A. Oladipupo, E. S. Chukwuebuka, T. S. Adekunle, and V. C. Onumaku, "Generative artificial intelligence (AI) on sustainable development goal 4 for tertiary education: conversational AI with user-centric ChatGPT-4," in *Impacts of Generative AI on Creativity in Higher Education*, IGI Global Scientific Publishing, pp. 263–292, 2024. https://doi.org/10.4018/979-8-3693-2418-9.ch010.

[17] D. Emakporuena, V. O. Oluwasegun, O. M. Esegbona-Isikeh, W. O. Ugbomeh, M. Nwaiku, E. E. Bunmi, and U. O. Matthew, "Harnessing Higher Generative Artificial Intelligence for Educational Pedagogy: Human-AI Collaboration in ChatGPT Experiences," in *Integrating ChatGPT Into System Applications and Services*, IGI Global Scientific Publishing, pp. 23–58, 2026. https://doi.org/10.4018/979-8-3693-9841-8.ch002.

[18] V. Alto, *Building LLM Powered Applications: Create Intelligent Apps and Agents with Large Language Models*, Packt Publishing Ltd., 2024. https://books.google.co.id/books?id=P1oIEQAAQBAJ.

[19] A. Livieratos *et al*., "MetaMind: A multi-agent transformer-driven framework for automated network meta-analyses," *PLOS ONE*, vol. 21, no. 2, e0342895, 2026. https://doi.org/10.1371/journal.pone.0342895.

[20] M. A. K. Raiaan, M. S. H. Mukta, K. Fatema, N. M. Fahad, S. Sakib, and M. M. J. Mim, "A review on large language models: Architectures, applications, taxonomies, open issues and challenges," *IEEE Access*, vol. 12, pp. 26839–26874, 2024. https://doi.org/10.1109/ACCESS.2024.3365742.

[21] L. Lin and Y. Liu, *Multimodal Large Models: A New Paradigm of Artificial Intelligence*. Cham: Springer, 2026. https://doi.org/10.1007/978-981-95-4929-0.

[22] N. Esmi, M. Nezhad-Moghaddam, F. Borhani, A. Shahbahrami, A. Daemdoost, G. Gaydadjiev, "GPT-5 vs Other LLMs in Long Short-Context Performance," in *2025 3rd International Conference on Foundation and Large Language Models (FLLM)*, pp. 129-133, 2025. https://doi.org/10.1109/FLLM67465.2025.11391194.

[23] A. Briouya, H. Briouya, A. Choukri, "Overview of the progression of state-of-the-art language models,". *TELKOMNIKA (Telecommunication, Computing, Electronics, and Control)*, vol. 22, no. 4, pp. 897-909, 2024. http://doi.org/10.12928/telkomnika.v22i4.25936.

[24] L. H. Yau, E. T. Y. Xian, Y. P. Yu, and T. M. Lim, "Retrieval Augmented Generation-based Large Language Model Chatbot," in *Proc. 2025 IEEE Int. Conf. on Computation, Big-Data and Engineering (ICCBE)*, pp. 856–861, Jun. 2025. https://doi.org/10.1109/ICCBE65177.2025.11255894.

[25] A. A. Gumel, A. B. Abdullahi, and U. M. O, "The Need for a Multimodal Means of Effective Digital Learning through Data Mining and Institutional Knowledge Repository: A Proposed System for Polytechnics in Northern Nigeria," In *Proceedings of the 2019 5th International Conference on Computer and Technology Applications*, pp. 81–85, April 2019. https://doi.org/10.1145/3323933.3324068.

[26] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, and D. Roth, "Recent advances in natural language processing via large pre-trained language models: A survey," *ACM Computing Surveys*, vol. 56, no. 2, pp. 1–40, 2023. https://doi.org/10.1145/3605943.

[27] M. S. G. Sourav, A. Javaid, and L. Cheng, "A review of AI in human-machine cooperation: Machine perspective," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 21, no. 2, Art. No. 19, pp. 1–43, 2025. https://doi.org/10.1145/3774318.

[28] V. N. Oriakhi, O. M. Esegbona-Isikeh, B. Esseme, A. Claude, D. Emakporuena, A. C. Nwanakwaugwu, and U. O. Matthew, "Generative artificial intelligence in education: ChatGPT-4 experiences to anticipated ChatGPT-5," *HAFED POLY Journal of Science, Management and Technology*, vol. 6, no. 1, pp. 149–169, 2024. https://doi.org/10.4314/hpjsmt.v6i1.1.

[29] L. Wang, W. Xu, Y. Lan, Z. Hu, Y. Lan, R. K. W. Lee, and E. P. Lim, "Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models," in *Proc. 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2609–2634, Jul. 2023. https://doi.org/10.18653/v1/2023.acl-long.147.

[30] J. Zheng *et al.*, "Lifelong learning of large language model based agents: A roadmap," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 48, no. 5, pp. 5552 –5571, 2026. https://doi.org/10.1109/TPAMI.2025.3650546.

[31] Y. Qian, "Prompt engineering in education: A systematic review of approaches and educational applications," *Journal of Educational Computing Research*, vol. 63, no. 7–8, pp. 1782–1818, 2025. https://doi.org/10.1177/07356331251365189.

[32] A. Mussa, Z. Tuimebayev, and M. Mansurova, "Make Large Language Models Efficient: A Review," *IEEE Access*, vol. 13, pp. 154466–154490, 2025. https://doi.org/10.1109/ACCESS.2025.3605110.

[33] M. Chen, Z. Liu, C. Chen, J. Wang, Y. Xue, B. Wu, and Q. Wang, "Beyond Static GUI Agent: Evolving LLM-based GUI Testing via Dynamic Memory," in *Proc. 2025 40th IEEE/ACM Int. Conf. on Automated Software Engineering (ASE)*, pp. 1603–1615, Nov. 2025. https://doi.org/10.1109/ASE63991.2025.00135.

[34] J. Zhang *et al.*, "When LLMs meet cybersecurity: A systematic literature review," *Cybersecurity*, vol. 8, art. No. 55, 2025. https://doi.org/10.1186/s42400-025-00361-w.

[35] J. Malik, R. Muthalagu, and P. M. Pawar, "A systematic review of adversarial machine learning attacks, defensive controls, and technologies," *IEEE Access*, vol. 12, pp. 99382–99421, 2024. https://doi.org/10.1109/ACCESS.2024.3423323.

[36] M. F. Kharma, S. Choi, M. Alkhanafseh, and D. Mohaisen, "Security and quality in LLM-generated code: A multi-language, multi-model analysis," *IEEE Transactions on Dependable and Secure Computing*, vol. 23, no. 3, 2026. https://doi.org/10.1109/TDSC.2026.3672745.

[37] L. Wang, C. Chen, J. Zhu, R. Zhan, and W. Han, "CQLLM: A Framework for Generating CodeQL Security Vulnerability Detection Code Based on Large Language Model," *Applied Sciences*, vol. 16, no. 1, p. 517, 2026. https://doi.org/10.3390/app16010517.

[38] H. Su, J. Luo, C. Liu, X. Yang, Y. Zhang, Y. Dong, and J. Zhu, "A survey on autonomy-induced security risks in large model-based agents," *arXiv preprint arXiv:2506.23844*, 2026. https://doi.org/10.48550/arXiv.2506.23844.

[39] M. N. Musa and M. E. Irhebhude, "An Empirical Analysis of Injection Attack Vectors and Mitigation Strategies in Redis NoSQL Database," *Journal of Computing Theories and Applications*, vol. 2, no. 4, pp. 553–571, 2025. https://doi.org/10.62411/jcta.12640.

[40] I. Hasanov, S. Virtanen, A. Hakkala, and J. Isoaho, "Application of large language models in cybersecurity: A systematic literature review," *IEEE Access*, vol. 12, pp. 176751–176778, 2024. https://doi.org/10.1109/ACCESS.2024.3505983.

[41] Y. Wang *et al.*, "Large model-based agents: State-of-the-art, cooperation paradigms, security and privacy, and future trends," *IEEE Communications Surveys & Tutorials*, vol. 28, pp. 1906–1949, 2025. https://doi.org/10.1109/COMST.2025.3576176.

[42] S. Izadi and M. Forouzanfar, "Error correction and adaptation in conversational AI: a review of techniques and applications in chatbots," *AI*, vol. 5, no. 2, pp. 803–841, 2024. https://doi.org/10.3390/ai5020041.

[43] B. Peng, K. Chen, M. Li, P. Feng, Z. Bi, J. Liu, and Q. Niu, "Securing large language models: Addressing bias, misinformation, and prompt attacks," *arXiv preprint arXiv:2409.08087*, 2024. https://doi.org/10.48550/arXiv.2409.08087.

[44] C.-H. H. Yang, A. Stolcke, and L. P. Heck, "Spoken Conversational Agents with Large Language Models," *arXiv preprint arXiv:2512.02593*, 2025. https://doi.org/10.48550/arXiv.2512.02593.

[45] G. P. Oise, B. S. Olanrewaju, O. A. Orukpe, K. C. Pius, and A. O. Airhiavbere, "A Convolutional Neural Network Framework for Intelligent Intrusion Detection," *Scientific Journal of Computer Science*, vol. 2, no. 1, pp. 50–59, 2026. https://doi.org/10.64539/sjcs.v2i1.2026.404.

[46] A. Banerjee and D. Banik, "A Comprehensive Survey on Transformer-Based Machine Translation: Identifying Research Gaps and Solutions for Large Language Models," *ACM Computing Surveys*, vol. 58, no. 5, pp. 1–33, 2025. https://doi.org/10.1145/3773076.

[47] T. Zhang, L. Patterson, B. Webb, Z. Jin, and M. Beiting-Parrish, "Evaluating generative AI chatbots for large-scale assessment data: comparing LLM-as-a-judge and human ratings," *Large-scale Assessments in Education*, vol. 14, 2026. https://doi.org/10.1186/s40536-026-00287-w.

[48] R. Shadiev, A. Kaliyeva, A. Kairatova, Z. Yerbulatova, A. Ryskulbek, and Z. Beisembayeva, "Exploring the Role of Conversational AI in Developing Communicative Skills: A Systematic Review," *Digital Applied Linguistics*, vol. 4, p. 103058, 2025. https://doi.org/10.29140/dal.v4.103058.

[49] H. Mirshekali, M. R. Shadi, F. G. Ladani, and H. R. Shaker, "A Review of Large Language Models for Energy Systems: Applications, Challenges, and Future Prospects," *IEEE Access*, vol. 13, pp. 163162 – 163188, 2025. https://doi.org/10.1109/ACCESS.2025.3610994.

[50] L. Aldhafeeri, F. Aljumah, F. Thabyan, M. Alabbad, S. AlShahrani, F. Alanazi, and A. Al-Nafjan, "Generative AI Chatbots Across Domains: A Systematic Review," *Applied Sciences*, vol. 15, no. 20, p. 11220, 2025. https://doi.org/10.3390/app152011220.

[51] M. Al-olaqi, A. Al-gailani, and M. H. Rahman, "Comprehensive study of SQL injection attacks mitigation methods and future directions," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 4, pp. 347–365, 2025. https://doi.org/10.63180/jcsra.thestap.2025.4.11.

[52] O. O. Abayomi, A. E. Kayode, O. Oluwasanya, A. E. Mesioye, "Comparative Analysis of Machine Learning Algorithms for Predictive Maintenance," *Methods in Science and Technology Studies*, vol. 2, no. 2, pp. 131-144, 2026. https://doi.org/10.64539/msts.v2i2.2026.505.

[53] G. Saleem, N. Ahmed, M. I. Zaman, and A. Hassan, "A Layered Security Framework Against Prompt Injection in RAG-Based Chatbots," *arXiv preprint arXiv:2606.19660*, 2026. https://doi.org/10.48550/arXiv.2606.19660.

[54] W. Xing, Z. Qi, Y. Qin, Y. Li, C. Chang, J. Yu, and M. Han, "Mcp-Guard: A Multi-Stage Defense-in-Depth Framework for Securing Model Context Protocol in Agentic AI," *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 4877–4889, July 2026. https://doi.org/10.18653/v1/2026.findings-acl.240.

[55] B. Ramakrishnan and A. Balaji, "Securing AI Agents Against Prompt Injection Attacks," *arXiv preprint arXiv:2511.15759*, 2025. https://doi.org/10.48550/arXiv.2511.15759.

[56] H. Li, X. Liu, C. H. I. U. Chun, D. Li, N. Zhang, and C. Xiao, "Drift: Dynamic Rule-Based Defense with Injection Isolation for Securing LLM Agents," *arXiv preprint arXiv:2506.12104*, vol. 38, pp. 83262–83290, 2026. https://doi.org/10.48550/arXiv.2506.12104.