ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov556

Adversarial Threat Simulation in Large Language Models: Red Teaming Beyond Prompt Injection

Ashwin Sharma[1]; Anshul Goel[2]

[1,2] Independent Researchers

Publication Date: 2025/11/18

Abstract: Big Language Models (LLMs) are becoming more and more exploited in sensitive areas, thus raising the issue of their security. The current red-teaming approaches, especially those that emphasize on timely injection, do not have much to say about weaknesses associated with these advanced approaches. The proposed research suggests the next-generation methods of adversarial threat-simulations in the context of LLM cybersecurity, which goes beyond the standard focus on prompt injection. An extensive theoretical framework is presented on how to classify adversarial threats which covers the whole lifecycle of the LLM, both during the training and the deployment. In the manuscript, the innovative red-teaming approaches, such as scenario-based simulations, automated adversarial generation, and ecosystem-wide red teaming are also described to give a more comprehensive review of LLM security. The most important conclusions are that the existing red-team activities are not sufficient to tackle the system vulnerabilities, which leaves LLMs vulnerable to both stage-by-stage and multi-stage attacks. The study has helped to advance a more serious method of obtaining LLMs, as well as provided information on extensive red-teaming solutions with an expanded attack surface and threat list. The results highlight the importance of ongoing and dynamic security evaluations and develop a basis on which future research can be conducted to make LLM more resilient to new adversarial threats.

Keywords: *Machine Learning Security, Red Teaming, Prompt Injection, Threat Modeling, Cybersecurity, Adversarial Simulation, Large Language Models, Generative AI.*

How to Cite: Ashwin Sharma; Anshul Goel (2025) Adversarial Threat Simulation in Large Language Models: Red Teaming Beyond Prompt Injection. *International Journal of Innovative Science and Research Technology*, 10(11), 660-673. https://doi.org/10.38124/ijisrt/25nov556

I. INTRODUCTION

A. Hook & Background

The rapid introduction of Large Language Models (LLMs) into such sensitive fields as healthcare, finance, and government has sparked groundbreaking changes in the business and industry paradigms of operation. Relying on advanced transformer architecture, these models have fundamentally transformed areas that are based on natural language processing (NLP), thus allowing the development of automated content generation, real-time translation, as well as decision support that have not been seen as possible before (Yao et al., 2024). With LLPs emerging as part of services that will affect millions of users, their security and reliability, take the center stage of importance. However, when implemented in sensitive settings, their presence subjects them to unique and dynamic cyber threats, making the evaluation of their security status highly significant (Abuadbba et al., 2025).

LLMs are susceptible to cyberattacks and abuse, and it is even more pressing as adversarial methods develop in complexity and sophistication. Even though important gains have been made in securing machine-learning (ML) models, LLMs are distinctive and require specialized and multifaceted security protocols (Amich, 2024). The typical red-team-style methods, designed to test AI systems through simulated threats, have mostly focused on prompt-injection methods, thus leaving a significant gap in the research on the more complex vulnerabilities of the wider systems represented by LLMs.

B. Problem Formulation

LLMs are vulnerable to numerous security risks, such as data leakage, i.e., sensitive or proprietary information that is stored in the training data of the model and is accidentally revealed to the user; hallucinations, i.e., the model produces incorrect or misleading results; and biased results, i.e. the model reinforces harmful stereotypes or misinformation (Khomsky et al., 2024). In addition to that, LLMs may be deployed in malicious applications, like the creation of malicious content or social engineering attacks being made automatic. Such weaknesses are especially worrisome, especially when the use of LLMs becomes part of high-stakes fields of life, like healthcare or criminal justice, where the level of trust and accuracy is paramount (Wang et al., 2024).

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov556

The recent security assessments of LLMs rely heavily on red-teaming techniques based on prompt-injection attacks. During such tests, attackers create input prompts to take advantage of vulnerabilities, manipulate the behavior of the LLM, or steal sensitive data (Verma et al., 2024). Timely injection just sketches on the possible security issues in LLMs. Although useful in establishing direct vulnerabilities and immediate threats, it cannot be considered a complete tool in assessing the cybersecurity of LLMs; it does not conquer more structural, complex, or multi-phase vulnerability, which uses weaknesses in the design of the model, in training data, or ecosystem around the model (Zangana et al., 2024).

C. Motivation & Rationale

The need to have a broader assessment model is clear. The fast development of the LLMs and their growing use in the mission-critical applications is what makes them very appealing to an adversary. Today, with the advancement of red-teaming methods, it is no longer sufficient to focus on red teaming through the lens of quick injection. New, sophisticated adversarial threat simulation models are also required to model sophisticated, multi-layered attacks that involve the full cycle of an LLM, i.e. training and fine-tuning as well as deployment and post-deployment phases. The existing security frameworks are not comprehensive enough and usually neglect threats that characterize underlying model constructions or interactions with the external systems (Liu and Hu, 2024). In this paper, the author suggests the enhancement of an existing red-teaming model, which includes the new advanced simulations to provide a more powerful and proactive approach to the evaluation of the LLM security (Swanda et al., 2025).

This study is motivated by the need to come up with a more organized and advanced adversarial threat simulation policy. More profound, lifecycle-based knowledge of the existing potential vulnerabilities is necessary when creating secure and robust LLMs. Current red-teaming solutions, which are mostly based on immediate injection, are not enough to address the threats that are emerging in the fast-evolving environment of the implementation of LLM (Liu et al., 2025). Through further progress in red-teaming approaches, it would be more appropriate to ensure that LLMs are more resistant to the wider range of adversarial risks, promoting more trustworthy AI systems.

D. Research Gap

As much literature has been created about the weaknesses of LLMs, existing research has focused on a limited range of attack vectors, specifically prompt injection and jail breaking (Khomsky et al., 2024). However, the topology of potential adversarial threats is much more multifaceted, and involves multiple stages of the lifecycle of the LLM and requires a more expansive security assessment strategy. Current literature tends to ignore other types of adversarial threats, including model evasion, data poisoning, and privacy leakage all of which may result in severe security breaches (Zangana et al., 2024). In addition, much vulnerability are not recognized and unaddressed since there is a lack of systematic approaches in which to conduct extensive threat simulation in LLMs. This research gap is aimed to be closed in this paper, by proposing a more inclusive, multi-dimensional methodology of adversarial threat modeling and red teaming.

A. Research Questions/Objectives

The following key research questions and objectives are the ones that will be answered in this paper:

- Are the red-teaming practices that are currently supported, characterized by immediate injection, sufficient to screen the cybersecurity position of LLMs?
- What is taxonomy of adversarial threat capable of being generated in full against both prompt injection and full lifecycle adversarial threat?
- What are some new red-teaming techniques that can be devised to emulate more advanced adversarial threats, taking into account the situation of multiple stages of attacks and vulnerabilities at an ecosystem level?
- What is the way to measure the effectiveness of these advanced red-teaming methodologies and how they can be used to improve the resilience of LLMs?

E. Contributions

The given manuscript provides a number of substantive contributions to the field of security of large language models (LLM).

- Red Teaming Practices Systematization: We introduce an organized summary of modern red-teaming techniques, critically assessing their weaknesses, especially prompt-injection weaknesses, and give a coherent plan of an improved threat modeling (Verma et al., 2024).
- Threat Taxonomy: We come up with a complete taxonomy of adversarial threats to LLMs in terms of attack surface, adversary goals and lifecycle phase, thus going beyond prompt injection to include data poisoning, model manipulation, and privacy abuses (Yao et al., 2024).
- Advanced Red Teaming Framework: We propose a new architecture of advanced red teaming, which includes automated generation of adversaries, simulation at the level of scenarios, and vulnerability attack at the ecosystem level (Rawat et al., 2024).
- Pragmatic Security Advice to Security Practitioners: We give operational advice and suggestions to the security practitioners and developers on how they can undertake proactive and comprehensive LLM security assessments in order to close the gap between the theory and practice of security (Swanda et al., 2025).

F. Paper Structure

The rest of this manuscript will be structured in the following way:

- Section 2: Background and Related Work – Gives the overview of the architecture of the LLM, its weaknesses, and the review of the traditional red-teaming practice along with the analysis of its drawbacks.
- Section 3: Adversarial Threat Taxonomy of LLMs - This paper proposes an extended taxonomy of adversarial threats along the lifecycle of LLM and is not limited to prompt injection.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov556

- Section 4: Advanced Red Teaming Methodologies - elaborates new red-teaming algorithm, such as scenario-based simulation and automated generation of adversaries.
- Section 5: Challenges and Future Directions - address the issue of scaling and reproducing red-teaming work and discuss the opportunities of the future research.
- Section 6: Conclusion - Summarizes the contributions and their implications to the future of the research and practice of the LLM security.

II. BACKGROUND AND RELATED WORK

This section forms a necessary background to understand the advanced adversarial threat simulation of Large Language Models (LLMs). It will start by giving a review of the architectures of the LLM and the security vulnerabilities inherent in the architecture, and then explore the most common types of security threats in the LLM. The discussion then goes into the traditional red-teaming approaches in the field of artificial intelligence and machine learning, with their application to LLMs highlighted. Lastly, the part is a critical analysis of the constraints of existing practices of LLM red teaming, especially those that focus on prompt injection, and how to create more advanced methods of evaluation.

A. Large Language Models

Large Language Models (LLMs) are one of the key innovations concerning the field of artificial intelligence, particularly natural language processing (NLP). Transformer architectures are mostly used to create these models, and they have been shown to be exceptionally effective at identifying patterns of complex linguistic behavior with large corpora of billions of tokens (Yao et al., 2024). Transformer-based models including GPT (Generative Pre-trained Transformers) and BERT (Bidirectional Encoder Representations from Transformers) have self-attention mechanisms that understand the relative importance of individual tokens in a sentence and thus allow the model to handle language in very parallelized way. Such an ability to work with large volumes of data allows LLMs to accomplish numerous tasks, including text generation and translation, summarization and question answering (Das et al., 2025).

Training of LLMs is based on large-scale unsupervised learning where they are applied to predict the next word in a given sequence based on large textual corpora. This pre-training stage is followed by the fine-tuning step whereby more application specific datasets are used to further optimize the outputs of the model (Abdali et al., 2024). However, in the context of the implementation of LLMs in such vital fields as finance, healthcare, and government, their safety and their ability not to fall victim to adversarial attacks are becoming crucial. Even with such impressive potential, these models cannot be exploited, especially due to the so-called emergent behaviors, which occur when these models are exposed to adversarial inputs (Zangana et al., 2024).

B. Introduction to LLM Security Threats

With the growing integration of the LLMs into the real-world use, they have widened their weak points. Radicalization of the security threats related to LLMs fall under broad categories, with each category potentially using different aspects of the architecture, training data, or ecosystem of deployment. These threats can be the most conspicuous including prompt injection, data poisoning, model evasion, and privacy attacks, among others.

- Prompt Injection: Prompt injection is an adversarial attack that is extensively documented and where adversarial inputs are designed to induce an LLM to behave in a way that is not expected. These attacks are either direct prompt injection, where the attacker literally encodes the command to circumvent the initial instructions of the model, or indirect injection, where the instructions are hidden in seemingly innocent data and processed by the model (Shayegani et al., 2023). Such attacks take advantage of the fact that the model depends on user input and thus bypasses the safety measures and either produces dangerous or biased content (Das et al., 2025).

- Data Poisoning: Data poisoning attacks are attacks that occur when a malicious entity provides data to the training or fine-tuning stage of the LLM. Attackers can manipulate the model to give inaccurate, biased, or harmful results by inserting corrupted or biased entries into the training set of the model. This also may lead to the opening of backdoors that can be taken advantage of under certain circumstances (Qiang et al., 2024).

- Model Evasion/Manipulation: This type of threat is used to make the LLM make erroneous predictions or misclassifications with the help of adversarial inputs. As an example, minor manipulations to input data might trigger the LLM to produce erroneous results, thus bypassing a security mechanism, like content moderation (Vitorino et al., 2024).

- Privacy Attacks: LLCMs too can fall prey to privacy based attacks like membership inference and model inversion. In membership inference, an adversary may be able to determine whether a data point was included in the model training set, which may be sensitive information. Model inversion also allows attackers to rebuild confidential information using the model outputs, and it is highly dangerous to privacy (Li et al., 2023).

The threats indicate the need to develop a more holistic way of securing LLMs, which is not limited to the use of prom injection as the most popular red-teaming tool.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov556

Table 1 Provides a Summary of the Key LLM Security Threat Categories, their Descriptions, and the Potential Impacts of Each.

Attack Category	Description	Examples of Impact
Prompt Injection & Jailbreaking	Crafting malicious input prompts to manipulate LLM behavior or bypass safety filters.	Generating harmful content, data exfiltration, prohibited role-playing.
Data Poisoning	Injecting malicious data into training or fine-tuning datasets.	Biased outputs, embedding backdoors, performance degradation.
Model Evasion/Manipulation	Adversarial inputs that cause misclassification or incorrect outputs.	Incorrect factual generation, bypassing moderation, inappropriate content.
Privacy Attacks	Methods to extract sensitive information from LLM training data.	Data leakage, membership inference, model inversion.
Model Theft/Extraction	Unauthorized acquisition of model parameters or architecture.	Intellectual property theft, facilitating further attacks.
Denial of Service	Overloading LLM services to disrupt availability.	Service downtime, resource exhaustion, operational disruption.
Supply Chain Vulnerabilities	Weaknesses in components or services used during LLM development.	Data breaches, system compromises, malware introduction.
Excessive Agency	Granting too much autonomy to an LLM, leading to unintended harmful actions.	Uncontrolled execution of actions, reputational damage.

Source: Adapted from the Existing Literature on LLM Security Threats (Das et al., 2025; Zangana et al., 2024)

C. Classical Red Teaming of AI/ML

Red teaming originally started in military simulations but is necessary currently in finding vulnerabilities in cybersecurity and AI systems. Red team attackers in AI/ML are used to simulate adversarial attacks to evaluate the strength of a model and reveal implicit weaknesses (Majumdar et al., 2025). They reason in the way of attackers, and AI is pushed to extremes in their efforts to embarrass issues that otherwise would not be noticed.

Red teaming in large-language-model (LLM) settings has mostly been interested in prompt injection and model evasion. This focus has shifted primarily to direct and indirect immediate action, a significant but impartial perspective of the possible threats. Concentrating on the inherent limitations of traditional red-team approaches, more systemic risks are possible due to the architecture of a model or data problems, which highlight why more developed methods are required to mitigate these risks (Purpura et al., 2025).

D. . Weaknesses of existing LLM Red Teaming

Given that it primarily aims at the immediate injection, it exposes LLMs to a larger attack scope. Although this method can be used to identify the vulnerabilities that are related to inputs, it is not used to address deeper security issues caused by the structure of the model, training data or the integration with other systems. In addition, most red-team activities are not an ongoing process and result in inconsistent testing that often does not test the entire attack surface (Zangana et al., 2024). Subsequently, prompt injection alone cannot be used to carry out a holistic security assessment.

E. Adversarial ML/AI security Relates to The security of ML/AI

Adversarial machine learning (AML) is not a new research area, particularly in image classification and computer vision, where adversarial examples are used to induce deep neural networks to incorrectly classify an image or object (Zangana et al., 2024). This base is the one that facilitates adapting adversarial methods to LLMs. Recent studies in the field of LLM attacks discuss the evasion attacks, data poisoning, and model inversion (Vitorino et al., 2024). These papers point to the necessity of a more systematic, rigorous approach towards security that goes beyond immediate injection to include the wide variety of adversarial threat to LLMs.

F. Conceptual- intelligent injection attack- Basic Prompt Injection Attack

The conceptual diagram below provides a simulation of a simple prompt injection attack. It shows that an adversarial input can control the behavior of an LLM by enabling it to exhibit guardrails, which are created to sieve out malicious content.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov556

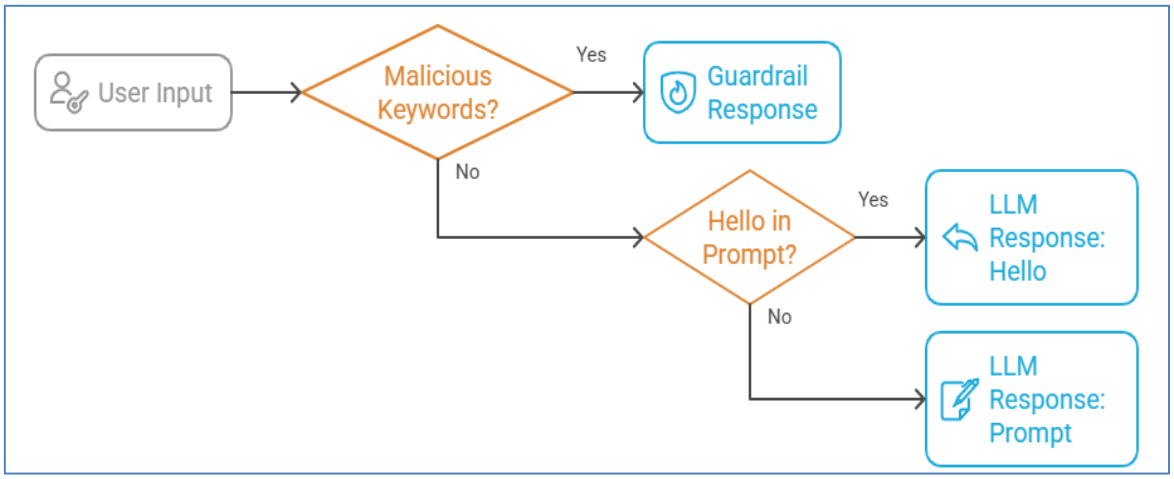

Fig 1: LLM with Simulated Guardrail
Source: Conceptual Example Adapted from Adversarial Techniques Discussed in Academic Literature on LLM Security (Li & Fung, 2025; Liu & Hu, 2024).

III. ADVERSARIAL THREAT TAXONOMY OF LLMS

Large Language Models (LLMs) have rapidly infiltrated various sectors, including healthcare, finance, legal and governmental practices. Such common usage exerts pressure on us to comprehend and minimize the adversarial risks that may at any stage of its life cycle assail LLMs. Although the majority of the red-teaming work has been on timely injection, this section provides a wider perspective. It suggests a hierarchical taxonomy that encompasses the entire LLM life cycle, including training, deployment and so on.

A. Beyond Timely Raisin

The emphasis on injecting now and making this the primary threat has shown objective weaknesses that are based on the way we allow the user to provide input (Hill et al., 2025). An attacker can modify the behavior of the model to bypass safety measures, evade a malicious prompt, or cause it to generate incorrect or malicious results (Verma et al., 2024). Designed to identify issues in the inference stage that attack is effective at failing to identify numerous other types of risks that LLMs are susceptible to.

The security context of the LLMs is broader. The other attack vectors influence other stages of the life cycle (Verma et al., 2024). Attackers may poison data during the training or fine-tuning of a model and add hidden bias that persists even after it becomes live (Qiang et al., 2024). Supply-chain attacks are able to ruin the entire ecosystem at deployment and introduce new vulnerabilities (Tete, 2024). Thus, the assessment of the security of LLM will involve a wider perspective beyond immediate injection and will examine the development, training, deployment, and operation.

B. Proposed Taxonomy/Framework

In order to develop a better perspective of the LLM security, this section presents a taxonomy that classifies threats based on the attack surface, attacker goals, and life-cycle stage. This multi-dimensional system gives us an opportunity to see the origin of each threat and its possible harm.

- Training-time Attacks: These attacks occur in the pre-training or fine-tuning. Such occurrences include data poisoning, in which harmful data has been introduced to the training set, and model back dooring, where latent triggers have been put in such a way that the model can be confused in the future (Huang et al., 2024). These attacks may reduce the performance or install hidden backdoors, which may be enabled by particular inputs.
- Inference-time Attacks: attackers are able to control the model in active mode. Prompt engineering does not just inject, but develops inputs that pass through filters or force the model to biased or damaging outputs. Evasion attacks are based on the fact that small modifications can be used to feed the model with misclassification. Information can be retrieved by privacy attacks such as side-channel approaches through the LLM APIs (Wang et al., 2025).
- Deployment-time/Ecosystem Attacks: New vulnerabilities are identified once the model is connected to other tools or APIs. Supply-chain attacks are aimed at the components of the system and harm the entire system. Lack of security in the design of the given plug-ins, or excessive freedom to the LLM may result in serious issues, including unauthorized activity or hacking into the system (Tete, 2024).

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov556

Table 2 Presents a Lifecycle-Based Classification of these Adversarial Threats, Highlighting their Impact at Different Stages of LLM Development and Deployment.

Attack Category	Attack Type	Description	Target LLM Lifecycle Phase	Potential Impact
Data Integrity & Availability	**Data Poisoning**	Malicious data injected into training or fine-tuning datasets.	Training, Fine-tuning	Biased outputs, performance degradation, backdoor activation (Yao et al., 2024).
	Model Backdooring	Embedding hidden triggers during training that activate malicious behavior.	Training, Fine-tuning	Undesired model behavior, security bypass (Liu & Hu, 2024).
Model Manipulation & Evasion	**Prompt Injection (Direct/Indirect)**	Manipulating input prompts to override model instructions or bypass filters.	Inference/Deployment	Harmful content generation, unauthorized actions (Liu & Hu, 2024).
	Adversarial Evasion	Subtle perturbations that mislead the model.	Inference/Deployment	Misleading information, bypassed content moderation (Vitorino et al., 2024).
Privacy & Confidentiality	**Membership Inference**	Determining if specific data was used in training.	Inference/Deployment	Exposure of private training data, privacy violations (Wang et al., 2025).
	Data Leakage	Unintended disclosure of sensitive information.	Inference/Deployment	Confidentiality breach, regulatory non-compliance (Wang et al., 2025).
System & Ecosystem Exploitation	**Supply Chain Attacks**	Attacking the components or services within the LLM's ecosystem.	All phases	Data breaches, system compromise (Tete, 2024).
	Insecure Plugin Design	Exploiting vulnerabilities in plugins that interact with LLMs.	Deployment/Integration	Unauthorized access, data manipulation (Tete, 2024).
	Excessive Agency Exploitation	Manipulating an LLM with too much autonomy, leading to unintended actions.	Deployment/Integration	Uncontrolled actions, reputational damage (Zangana et al., 2024).

Source: Adapted from the Proposed Framework based on LLM Security Threats (Huang et al., 2024; Verma et al., 2024).

C. Attack Surfaces and Vectors

An entry point an attack surface is all potential paths of interaction between an adversary and a system. The specific methods used to execute those attacks are referred to as attack vectors. In the case of large language models (LLMs), the attack surface will extend beyond the user interface into numerous aspects of the life cycle of the model.

- Input Prompts: This is the most obvious attack surface of the user; an example is input to the model. The attackers (Liu and Hu, 2024) often employ prominent injection and other manipulation techniques. Since these prompts have a direct influence on the resulting model, they are able to bypass safety filters and make the model generate malicious content (Hill et al., 2025).
- Training Data: The adversaries are able to poison the data or place backdoors during pre-training or fine-tuning of the model. This model has this attack surface as critical because it alters the fundamental behavior of the model by poisoning the training data (Verma et al., 2024).
- Model Architecture: It is also possible to target the design and internal parameters of an LLM. Attackers can attempt to steal or extract the model, reverse-engineer its parameters to create a duplicate or provide other leverage (Wang et al., 2025).
- External Integrations: LLM has a tendency to be linked to third-party tools, APIs, and other extensions, increasing the number of attack surfaces. Any of the parts that are not well designed such as the abusive API or the creation of a bad plug-in can helps attackers compromise the security of the LLM (Tete, 2024).

Figure 2 depicts the attack surfaces and vectors in a LLM and it is clear that each threat is presented in various stages of the life cycle of the model.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov556

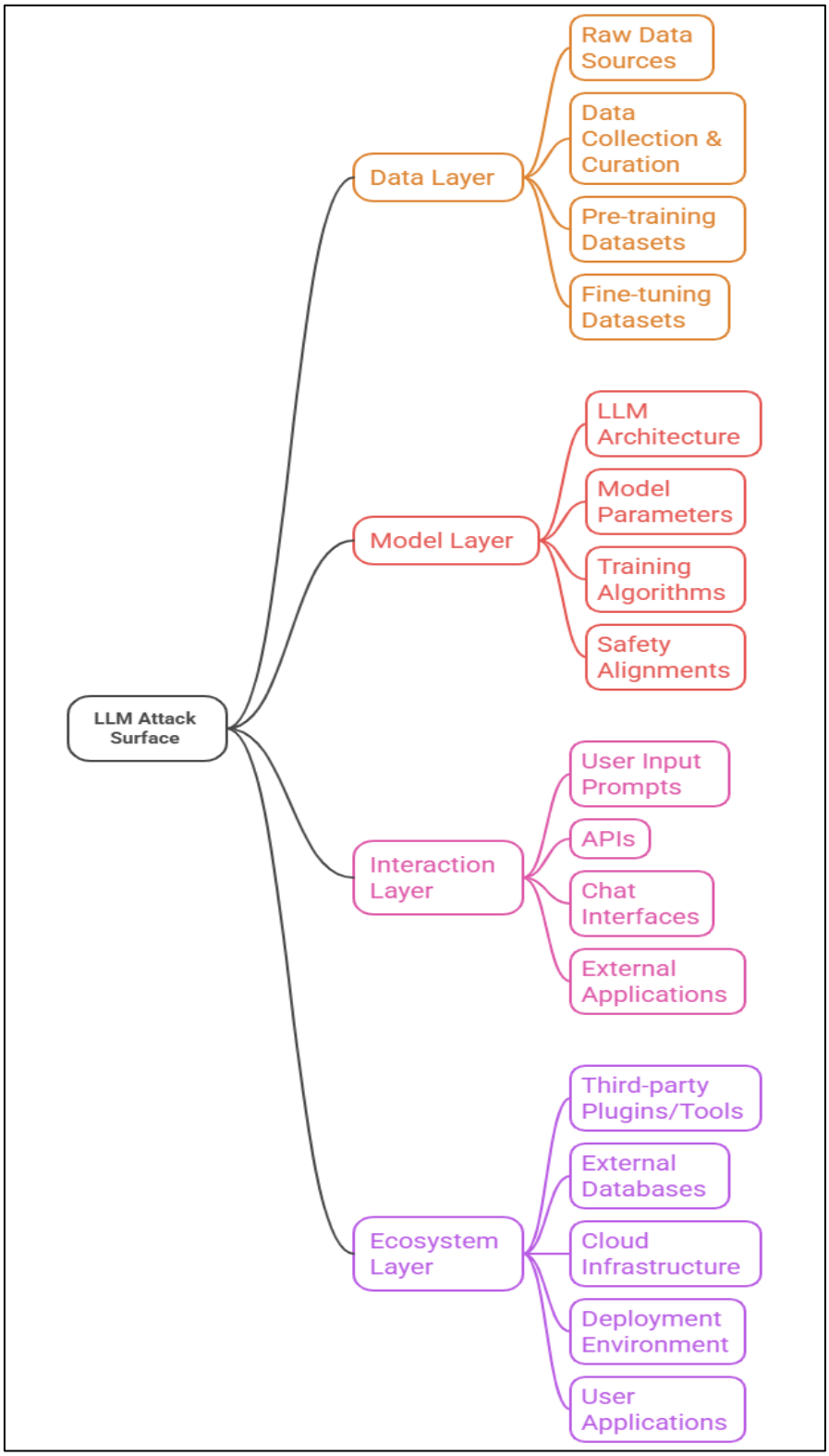

Fig 2: LLM Attack Surface Diagram

Source: Conceptual Representation Synthesizing Attack Surfaces from Various Academic Works, Including those Discussing LLM Lifecycle Vulnerabilities (Huang et al., 2024), Prompt-Based Attacks (Hill et al., 2025), Data Poisoning (Liu & Hu, 2024), and Ecosystem Integration Risks (Tete, 2024), (Verma et al., 2024), (Wang et al., 2025).

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov556

IV. STATE OF THE ART RED TEAMING METHODOLOGIES AND SIMULATIONS

With the spread of the application of Large Language Models (LLMs) into various industrial fields, the need for profound and comprehensive security evaluations increases respectively. Traditional red-teaming approaches that have mostly focused on immediate injection attacks are ineffective in identifying the broad category of adversarial threats that faces LLCM. Subsequently, new red-teaming modalities should not be limited by quick injection, but should be multi-faceted and take into account the whole lifecycle of the LLM, including the training, deployment, and post-deployment. The section defines such strategies and methodologies, combining domain knowledge, systematized steps, and multi-stage attack modeling in order to enhance LLM security.

A. Concepts of Comprehensive LLM Red Teaming

Advanced red teaming goes beyond the single-attack simulation but adopts an integrated, goal-based paradigm, which aims at exposing the vulnerable points throughout the full spectrum of the LLM lifecycle. One of the salient principles is the multi-stage nature of adversarial threats, which require a red team to reproduce attacks that, in addition to the ability to exploit the immediate weaknesses, can trigger cascading vulnerabilities that will appear in the future when the model is used (Verma et al., 2024). In line with this, the holistic red teams should involve substantial domain expertise to emulate the functioning of the LLMs in particular contexts and various limitations, thus enabling the simulation of more realistic adversarial scenarios.

In addition, comprehensive red teaming advances beyond the entry-level attack strategies like timely injection, and is centered around advanced and multi-dimensional attacks that exploit the relationships between the model architecture, training corpora, and other tools and the overall operating ecosystem (Abuadbba et al., 2025). Such an approach provides a more holistic analysis, which reveals the risks that cannot be identified by the surface research but have significant long-term consequences.

B. Threat Simulation Methodologies

Scenario-Based Red-teaming is one of the most effective approaches to simulating threats. Red teams can model real-world attackers by building complex, multi-phase attack stories by trying to exploit the vulnerabilities of a model over time. These simulated scenarios include a series of events that touch on various stages of the LLM lifecycle, including data poisoning in the training stage to adversarial evasion in the inference stage. In this way, it is possible to identify vulnerabilities that could be missed in a standalone attack test (Al-Azzawi et al., 2025).

The other methodology that stands out is Automated Adversarial Generation where the artificial intelligence is used to find new forms of adversarial inputs or techniques. It involves the use of techniques including genetic algorithms or reinforcement learning on par with the automatization of the generation of adversarial examples that can undermine the current defenses (Verma et al., 2024). Adversarial generation can also be automated, which is particularly useful since it may reveal new attack vectors that a human tester will not anticipate.

The Supply Chain and Ecosystem Red-Teaming are aimed at the larger ecosystem of developing and deploying the LLM. This is a simulated methodology that consists of attacks directed at the model supply chain, including injecting malicious code in the form of compromised datasets, insecure plugins or vulnerable third-party services. Through focusing on the interdependencies of the LLM ecosystem, red teams will be able to spot systemic weaknesses that can be used by adversaries (Zangana et al., 2024).

Human in the Loop Red teaming combine the ingenuity of human opponents with automation tools in the creation of iterative red-teaming processes. In this model, red teams use automated systems to create adversarial prompts, and then optimize these attacks using human experts, who give attention to new situations and add more domain-specific information (Bullwinkel et al., 2025). Such a hybrid approach can make the simulation of the attack more profound, making the creation of threat models more detailed and realistic.

Lastly, Targeted Vulnerability Analysis is focused on specific components of the model or architecture weaknesses. Red teams can perform penetration tests that focus on the most exposed weaknesses by isolating the critical parameters of the LLM e.g. model parameters, input/output interface or safeguards (Abdali et al., 2024). The approach provides information about particular areas of concern in a detailed manner that helps developer's correct weak areas before their implementation.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov556

Table 3 Summarizes the Methodologies Discussed, Outlining their Focus Areas and Potential Impact on LLM Security.

Methodology	Focus Area	Potential Impact
Scenario-Based Red Teaming	Multi-step attack narratives simulating real-world adversaries	Identifies vulnerabilities through complex, extended attack simulations (Al-Azzawi et al., 2025).
Automated Adversarial Generation	AI-driven creation of novel adversarial examples	Uncovers previously unknown attack vectors through automation (Verma et al., 2024).
Supply Chain & Ecosystem Red Teaming	Attacks on the LLM development lifecycle, third-party integrations	Highlights systemic risks and integration weaknesses (Zangana et al., 2024).
Human-in-the-Loop Red Teaming	Combining human creativity with automated tools	Enhances the quality of attack simulations by incorporating domain knowledge (Bullwinkel et al., 2025).
Targeted Vulnerability Analysis	Penetration testing of specific model components	Exposes deeper, more complex vulnerabilities in critical areas (Abdali et al., 2024).

Source: Adapted from advanced red-teaming methodologies for LLMs (Verma et al., 2024; Bullwinkel et al., 2025)

C. Metrics and Evaluation

The effectiveness of red teaming requires a two-fold strategy based on quantitative and qualitative measurements. The Attack Success Rate is one of them, a list that includes the number of times an adversarial attack has reached its desired goal, namely, the subversion of a safety guardrail or the retrieval of confidential information (Hassanin and Moustafa, 2024). Similarly, the Evasion Rate is an important indicator since it measures the rate of adversarial inputs that are able to evade defenses or countermeasures (Abuadbba et al., 2025).

The Novelty of Attack measure assesses the percentage of the red-teaming methods that are not known or identified by restrictive architectures (Purpura et al., 2025). This metric maintains the exploratory vigor of red-team engagements by ensuring that the adversarial operations are found to explore previously untapped vulnerabilities as opposed to simply re-exercising known attack patterns. In its turn, the Impact Assessment measures the possible damage related to successful exploits, including the data leakage, system downtime, and theft of intellectual property. Lastly, the Cost of Attack metric evaluates the resources that an adversary needs to allocate to a successful operation that would offer the necessary understanding of the strategic viability of a specific adversarial strategy (Verma et al., 2024).

Figure 3 presents a conceptual diagram illustrating these evaluation metrics and their relationships in the context of advanced LLM red-teaming.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov556

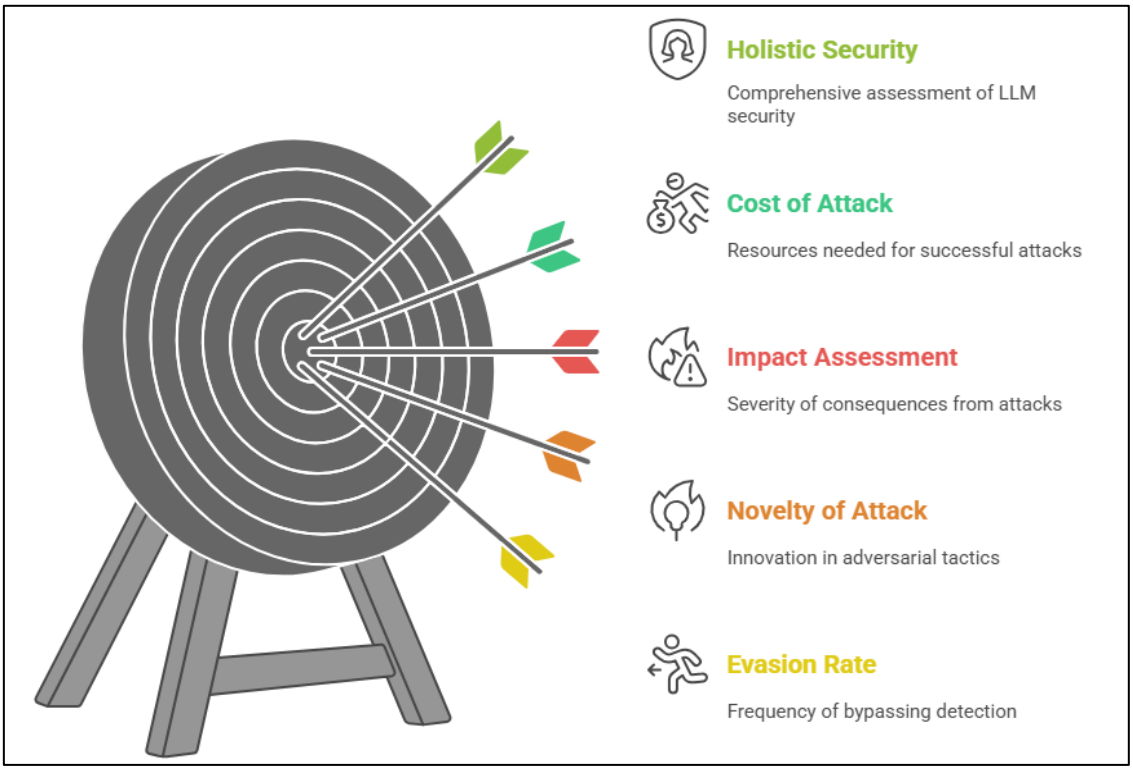

Fig 3: Red Team Metrics for LLM Security
Source: Adapted from Red-Teaming Evaluation Metrics Discussed in LLM Security Literature (Hassanin & Moustafa, 2024; Verma et al., 2024).

V. 5. ISSUES AND FUTURE PROJECTIONS

Large Language Models (LLM) security is a dynamic research field that faces many challenges. With the further integration of LLMs in critical fields, the difficulty of securing such models also grows in line with it. The present section clarifies the current barriers to successful red teaming of LLMs and outlines the research directions that can address them in the future. By creating more resilient methodologies and frameworks of LLCM security, the domain can guarantee that the models continue to be resilient to a wider range of adversarial attacks.

A. Current Challenges

There are a number of issues that are currently impeding the full assessment of LLC security. These issues are linked to the inherent complexity of LLMs and to the constant change of the adversarial strategies.

Scalability is also a major problem when implementing red teaming with large and complicated LLMs. These models, which consist of billions of parameters, pose significant challenges of testing their security on scale. Red-teaming simulations on such large models necessitate massive computational power and advanced methods of simulating realistic adversarial situations during the lifecycle of the model (Verma et al., 2024). The larger and more complex the LLM becomes, the more urgent the need to scale security assessments becomes.

Another significant issue is the reproducibility of the red-teaming results. Adversarial attacks are often very subtle and cannot always be replicated by different systems or environments. Such impossibility of reproducibility can compromise the credibility of security measurements, and it becomes difficult to determine whether this or that attack or vulnerability is real threat (Abuadbba et al. 2025). In addition, the variety of the attack vectors makes it difficult to ensure that every potential vulnerability is uncovered and tested on a regular basis.

Red teaming the LLMs is an extremely important issue concerning the ethical aspect. Although red teaming is mandatory to determine vulnerabilities, it has a tendency of abuse. An example is the use of adversarial methods with ill intentions to abuse LLMs or spread fake news. It is a continuous challenge to make sure that red teaming is carried out in a responsible manner and supported by sufficient safeguards to ensure that it is not exploited (Al-Azzawi et al., 2025). In addition, the ethical side of vulnerabilities disclosure revealed during red teaming should be carefully considered not to increase risks.

The dynamism of LLMs is also associated with additional challenges. Since LLMs are continually being improved through updates and retraining, the vulnerabilities that were previously discovered might reoccur, or new vulnerabilities can arise. This requires continuous red teaming to ensure that the swift changes of the model architecture and functionality are covered (Abdali et al., 2024). It is not viable to use the static security assessment to overcome this difficulty; it must be dynamic and continuous evaluation that will guarantee long-term integrity in the models.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov556

Lastly, the lack of a ground truth in defining the definition of what exactly is considered secure or safe behavior in regards to LLMs makes security consideration challenging. The LLM also provides a certain level of randomness in its output compared to more traditional systems, in which a particular attack vector can be predicted and exercised. The question of an agreeable limit of conduct in these models constitutes a severe difficulty and requires continued studies that would quantify and measure the security and trustworthiness of LLM answers (Dong et al., 2024).

Table 4 Summarizes these Key Challenges in the Red Teaming of LLMs, Highlighting their Potential Impact on the Security Evaluation Process.

Challenge	Description	Impact on Security Evaluation
Scalability	Difficulty in scaling red-teaming efforts for large, complex models.	Requires significant computational resources and sophisticated tools.
Reproducibility	Inconsistent findings across different systems or settings.	Makes it challenging to validate vulnerabilities and assess model security.
Ethical Considerations	Potential misuse of red-teaming techniques, leading to exploitation or misinformation.	Raises concerns about responsible disclosure and the ethical use of adversarial techniques.
Dynamic Nature of LLMs	Continuous evolution of models through updates and retraining.	Requires ongoing red-teaming to address new vulnerabilities.
Lack of Ground Truth	Unclear definition of "secure" behavior due to unpredictable model outputs.	Complicates the establishment of benchmarks for model safety and reliability.

Source: Adapted from Current Challenges in LLM Security and Red-Teaming Literature (Verma et al., 2024; Abuadbba et al., 2025)

B. Future Research Avenues

Although the above challenges have been noted, future of the use of LLM security appears bright, if research and innovation are continued to be consistent. Several important thematic directions will be necessary in order to get out of the existing constraints and to develop the discipline of LLM red teaming.

The standardized benchmark on the security of LLM is a crucial move towards maintaining uniformity of the red-teaming programs. Currently there is no standardized framework to measure the security posture of LLMs, which hinders comparison of results across studies with different heterogeneous study and methodology. Setting the standards that represent a wide range of possible threats, including timely injection, information pollution, and privacy invasion, will offer a more organized method of the assessment of the security of LLM (Feffer et al., 2024).

One research that is more promising is AI-generated defensive systems. It is with the help of AI detecting and mitigating adversarial attacks by it in real-time that it will be possible to design the self-healing LLMs, which will dynamically adjust to newly discovered weaknesses. This may significantly decrease the amount of manual labor required by the conventional red-teaming process and at the same time increase the resilience of models to new attacks (Wang et al., 2025).

It is also important to incorporate formal verification techniques into the processes of maintaining LLM. The formal techniques, which are used to demonstrate mathematically that an algorithm is correct, can be generalized to ensure that before deployment, the LLMs meet the defined security requirements. This method will offer more guarantees about model safety and could prevent the risks in advance until the model faces real-life opponents (Amich, 2024).

Another area that should be the focus of intensive research is cross-model and multi-model LLM security. Many new LLMs are designed to handle multi-modal inputs, such as text, images and audio, thus providing more attack surfaces that need to be assessed using red-team based approaches to systems operating in multi-modes. As the LLMs grow to be able to interact in various modalities, red-teaming protocols will also have to adapt to provide a complete coverage of all the possible vulnerabilities (Tete, 2024).

Finally, policy and regulatory implications of LLM security and red-teaming are going to receive more significance as such models find their way to sensitive settings. The development of specific regulations and policies to control the ethical application of all LLMs combined with the provisions regarding the exposure of vulnerabilities will

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov556

serve to balance the need to ensure security and the risk of abuse (Das et al., 2025).

Figure 4 suggests future research opportunities in the field of LLM security, which provide a conceptual framework to further research undertakings in order to strengthen such models.

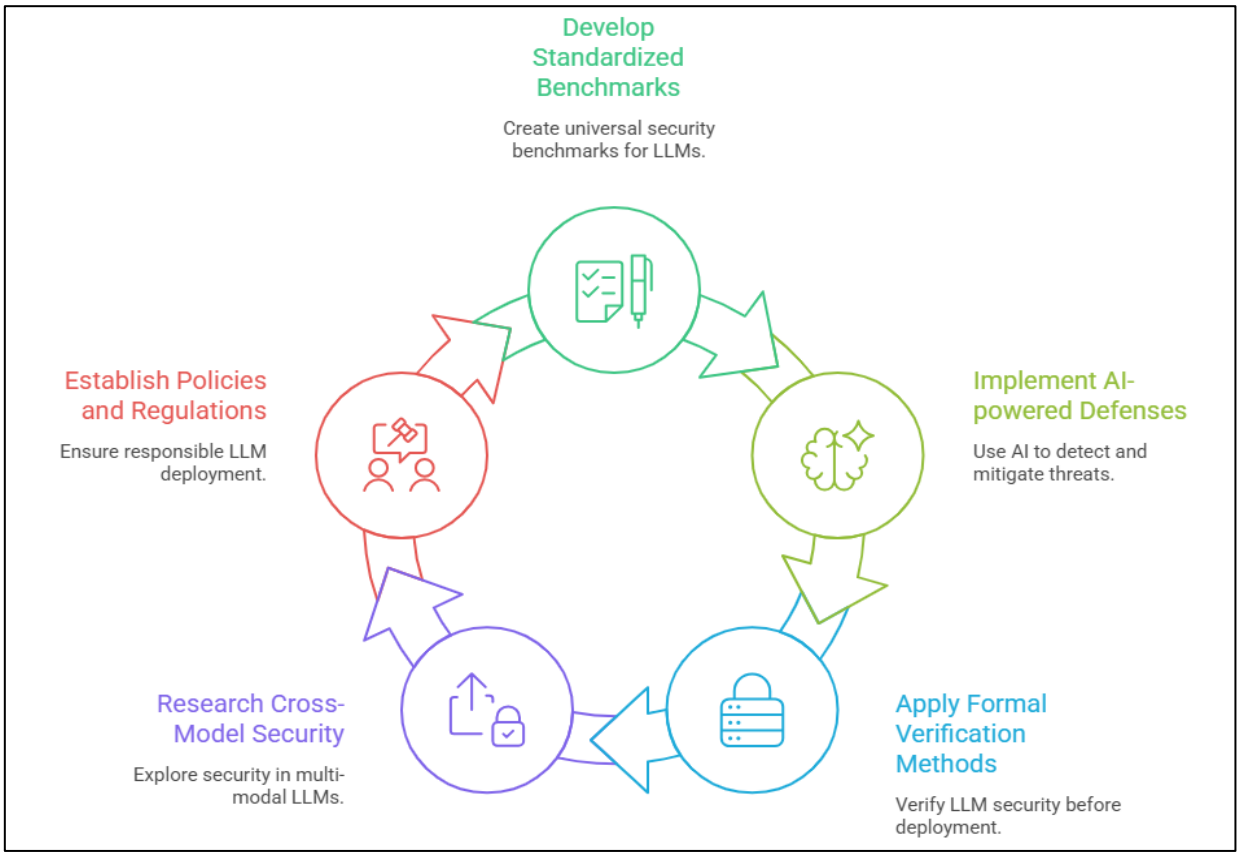

Fig 4: Future Research Directions for LLM Security
Source: Adapted from Future Research Directions in LLM Security (Feffer et al., 2024; Wang et al., 2025; Tete, 2024).

VI. CONCLUSION

➢ *Summary of Contributions*

This paper will provide an in-depth discussion of the security issues with Large Language Models (LLMs), which will clarify the need to develop advanced adversarial threat simulations. Careful consideration of weaknesses of modern red-teaming approaches, especially those focused on injection in the immediate future, the given research offers a more structured and comprehensive system of studying the security of LLM. The main value of the work is the development of adversarial threat taxonomy, which covers the whole cycle of life of LLMs, including their initial training and further fine-tuning as well as final deployment and interaction with external systems. This taxonomy categorizes threats into different stages, such as training-time, inference-time, and deployment-time, which is a more detailed way to make LLMs more resilient (Verma et al., 2024; Zangana et al., 2024).

Along with the taxonomic framework, the article presents more sophisticated red-teaming mechanisms that go beyond the timely injection and use more sophisticated strategies like scenario-based red-teaming, automated adversarial generation, and red-teaming of the supply chain. These approaches will help to reveal more in-depth weaknesses that are often never considered during more traditional red-team processes, and will subsequently lead to a more comprehensive assessment of the security of the LLM (Al-Azzawi et al., 2025). Moreover, the argument about measures and metrics of red-teaming can provide a concrete framework of evaluating the effectiveness of adversarial simulation that provides both quantitative and qualitative indicators of the effectiveness of the simulation, which include the success rate of attacks, the evasion rate, and the novelty of attacks (Hassanin and Moustafa, 2024).

➢ *Implications*

The results of this study have serious consequences to the creation and implementation of LLMs. This article highlights the need to adopt a proactive security strategy in the development of LLCs by expanding the scope of red - teaming to capture a broader supplier of attack vectors and lifecycle phases. The conventional testing approaches that emphasize timely injection are also becoming insufficient in improving the adaptability and sophistication of the adversarial threats to LLMs (Abdali et al., 2024; Abuadbba et al., 2025). This contribution suggests a paradigm shift of two aspects of the security of LLM-that of reactive, isolated testing to dynamic and holistic model of security that

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov556

comprises of continuous monitoring and evaluation of the entire lifecycle of the model.

The suggested adversarial threat taxonomy and improved red-teaming techniques have the potential to significantly enhance the resilience of LLM both in terms of exposing vulnerabilities that were previously hidden as well as in supporting the development of more robust and reliable models. Since LLMs are deployed in a high-stakes application in many areas like healthcare, finance, and government, it becomes necessary to enhance their security to ensure sensitive information is safeguarded and model outputs can be trusted (Das et al., 2025). Besides, the development of the LLMs requires the security evaluation frameworks to develop in line with the new threats and address the risks that emerge (Dong et al., 2024).

➢ *Final Thoughts*

To conclude, the security of Large Language Models is a constant debate that requires new solutions and development. Although significant strides have been made towards identifying and eliminating vulnerabilities like prompt injection, this article has shown that such need to go beyond a wider application to include a wider range of adversarial threat. The sophisticated red-teaming approaches and the adversarial threat taxonomy introduced herein provide a more detailed and well-organized method of assessing the security of the LLM, providing an effective basis on the upcoming study and applications in ensuring the security of such models.

The importance of advanced adversarial threat simulations in making the LLMs credible and safe cannot be underestimated. The security vulnerabilities of LLMs are becoming more problematic as the systems become more important in the critical infrastructure. The red-teaming tools and methodologies can be improved and enhanced, and through this approach, the research community will better predict and prevent possible threats so that the LLMs are safe, reliable and stable to the changing tactics used by adversaries. The future of the security of the LLM will be based on the further optimization of such evaluation frameworks and continuous interdisciplinary cooperation between scientists, developers, and field experts in order to create a model that can withstand the sophisticated realities of the digital era (Verma et al., 2024; Fu et al., 2024).

REFERENCES

[1]. Abdali, S., Anarfi, R., Barberan, C., He, J., & Shayegani, E. (2024). *Securing Large Language Models: Threats, Vulnerabilities and Responsible Practices*. https://doi.org/10.48550/ARXIV.2403.12503

[2]. Abuadbba, A., Hicks, C., Moore, K., Mavroudis, V., Hasircioglu, B., Goel, D., & Jennings, P. (2025). *From Promise to Peril: Rethinking Cybersecurity Red and Blue Teaming in the Age of LLMs*. https://doi.org/10.48550/ARXIV.2506.13434

[3]. Al-Azzawi, M., Doan, D., Sipola, T., Hautamäki, J., & Kokkonen, T. (2025). *Red Teaming with Artificial Intelligence-Driven Cyberattacks: A Scoping Review*. https://doi.org/10.48550/ARXIV.2503.19626

[4]. Amich, A. (2024). Multifaceted Characterization and Enhancement of Machine Learning Security. *Deep Blue (University of Michigan)*. https://doi.org/10.7302/24910

[5]. Bullwinkel, B., Minnich, A., Chawla, S., Lopez, G., Pouliot, M., Maxwell, W., de Gruyter, J., Pratt, K., Qi, S., Chikanov, N., Lutz, R., Dheekonda, R. S. R., Jagdagdorj, B.-E., Kim, E., Song, J., Hines, K., Jones, D., Severi, G., Lundeen, R., … Russinovich, M. (2025). *Lessons From Red Teaming 100 Generative AI Products*. https://doi.org/10.48550/ARXIV.2501.07238

[6]. Das, B. C., Amini, M. H., & Wu, Y. (2025). Security and Privacy Challenges of Large Language Models: A Survey [Review of *Security and Privacy Challenges of Large Language Models: A Survey*]. *ACM Computing Surveys*. Association for Computing Machinery. https://doi.org/10.1145/3712001

[7]. Derczynski, L., Galinkin, E., Martin, J., Majumdar, S., & Inie, N. (2024). garak: A Framework for Security Probing Large Language Models. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2406.11036

[8]. Derner, E., Batistič, K., Zahálka, J., & Babuška, R. (2023). A Security Risk Taxonomy for Large Language Models. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2311.11415

[9]. Dong, Y., Mu, R., Zhang, Y., Sun, S., Zhang, T., Wu, C., Jin, G., Qi, Y., Hu, J., Meng, J., Bensalem, S., & Huang, X. (2024). Safeguarding Large Language Models: A Survey. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2406.02622

[10]. Feffer, M., Sinha, A., Deng, W. H., Lipton, Z. C., & Heidari, H. (2024). *Red-Teaming for Generative AI: Silver Bullet or Security Theater?* https://doi.org/10.48550/ARXIV.2401.15897

[11]. Fu, T., Sharma, M., Torr, P., Cohen, S. B., Krueger, D., & Barez, F. (2024). *PoisonBench: Assessing Large Language Model Vulnerability to Data Poisoning*. https://doi.org/10.48550/ARXIV.2410.08811

[12]. Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). *Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection*. 79. https://doi.org/10.1145/3605764.3623985

[13]. Hassanin, M., & Moustafa, N. (2024). A Comprehensive Overview of Large Language Models (LLMs) for Cyber Defences: Opportunities and Directions. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2405.14487

[14]. Hill, B., Parla, S., Balabhadruni, V. A., Padmalayam, A. P., & Sharma, S. C. S. (2025). *Breaking to Build: A Threat Model of Prompt-Based Attacks for Securing LLMs*. https://doi.org/10.48550/ARXIV.2509.04615

[15]. Huang, X., Ruan, W., Huang, W., Jin, G., Dong, Y., Wu, C., Bensalem, S., Mu, R., Yi, Q., Zhao, X., Cai, K., Zhang, Y., Wu, S., Xu, P., Wu, D., Freitas, A., & Mustafa, M. (2024). A survey of safety and trustworthiness of large language models through the lens of verification and validation. *Artificial Intelligence Review*, *57*(7). https://doi.org/10.1007/s10462-024-10824-0

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov556

[16]. Khomsky, D., Maloyan, N., & Nutfullin, B. (2024). Prompt Injection Attacks in Defended Systems. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2406.14048

[17]. Li, H., Chen, Y., Luo, J., Kang, Y., Zhang, X., Hu, Q., Chan, C., & Song, Y. (2023). Privacy in Large Language Models: Attacks, Defenses and Future Directions. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2310.10383

[18]. Li, M. Q., & Fung, B. C. M. (2025). Security concerns for Large Language Models: A survey. *Journal of Information Security and Applications*, *95*, 104284. https://doi.org/10.1016/j.jisa.2025.104284

[19]. Liu, F. W., & Hu, C. (2024). Exploring Vulnerabilities and Protections in Large Language Models: A Survey. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2406.00240

[20]. Liu, S., Sheng, Q., Wang, D., Li, Y., Yang, G., & Cao, J. (2025). *Forewarned is Forearmed: Pre-Synthesizing Jailbreak-like Instructions to Enhance LLM Safety Guardrail to Potential Attacks*. https://doi.org/10.48550/ARXIV.2508.20038

[21]. Majumdar, S., Pendleton, B., & Gupta, A. (2025). *Red Teaming AI Red Teaming*. https://doi.org/10.48550/ARXIV.2507.05538

[22]. Miranda, M., Ruzzetti, E. S., Santilli, A., Zanzotto, F. M., Bratières, S., & Rodolà, E. (2024). Preserving Privacy in Large Language Models: A Survey on Current Threats and Solutions. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2408.05212

[23]. Pathade, C. (2025). *Red Teaming the Mind of the Machine: A Systematic Evaluation of Prompt Injection and Jailbreak Vulnerabilities in LLMs*. https://doi.org/10.48550/ARXIV.2505.04806

[24]. Purpura, A., Wadhwa, S., Zymet, J., Gupta, A., Luo, A. A., Rad, M. K., Shinde, S., & Sorower, M. S. (2025). *Building Safe GenAI Applications: An End-to-End Overview of Red Teaming for Large Language Models*. 335. https://doi.org/10.18653/v1/2025.trustnlp-main.23

[25]. Qiang, Y., Zhou, X., Zade, S. Z., Roshani, M. A., Zytko, D., & Zhu, D. (2024). Learning to Poison Large Language Models During Instruction Tuning. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2402.13459

[26]. Radanliev, P., & Santos, O. (2023). *Adversarial Attacks Can Deceive AI Systems, Leading to Misclassification or Incorrect Decisions*. https://doi.org/10.20944/preprints202309.2064.v1

[27]. Rawat, A., Schoepf, S., Zizzo, G., Cornacchia, G., Hameed, M. Z., Fraser, K., Miehling, E., Buesser, B., Daly, E. M., Purcell, M., Sattigeri, P., Chen, P.-Y., & Varshney, K. R. (2024). *Attack Atlas: A Practitioner's Perspective on Challenges and Pitfalls in Red Teaming GenAI*. https://doi.org/10.48550/ARXIV.2409.15398

[28]. Shayegani, E., Mamun, M. A. A., Fu, Y., Zaree, P., Dong, Y., & Abu-Ghazaleh, N. (2023). Survey of Vulnerabilities in Large Language Models Revealed by Adversarial Attacks. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2310.10844

[29]. Swanda, A., Chang, A., Chen, A., Burch, F., Kassianik, P., & Berlin, K. (2025). A Framework for Rapidly Developing and Deploying Protection Against Large Language Model Attacks. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2509.20639

[30]. Tete, S. B. (2024a). *Threat Modelling and Risk Analysis for Large Language Model (LLM)-Powered Applications*. https://doi.org/10.48550/ARXIV.2406.11007

[31]. Tete, S. B. (2024b). Threat Modelling and Risk Analysis for Large Language Model (LLM)-Powered Applications. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2406.11007

[32]. Verma, A., Krishna, S., Gehrmann, S., Seshadri, M., Pradhan, A., Ault, T., Barrett, L., Rabinowitz, D., Doucette, J., & Phan, N. (2024a). *Operationalizing a Threat Model for Red-Teaming Large Language Models (LLMs)*. https://doi.org/10.48550/ARXIV.2407.14937

[33]. Verma, A., Krishna, S., Gehrmann, S., Seshadri, M., Pradhan, A., Ault, T., Barrett, L., Rabinowitz, D., Doucette, J., & Phan, N. (2024b). Operationalizing a Threat Model for Red-Teaming Large Language Models (LLMs). *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2407.14937

[34]. Vitorino, J., Maia, E., & Praça, I. (2024). Adversarial Evasion Attack Efficiency against Large Language Models. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2406.08050

[35]. Wan, A., Wallace, E., Shen, S., & Klein, D. (2023). Poisoning Language Models During Instruction Tuning. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2305.00944

[36]. Wang, N., Walter, K., Gao, Y., & Abuadbba, A. (2025). *Large Language Model Adversarial Landscape Through the Lens of Attack Objectives*. https://doi.org/10.48550/ARXIV.2502.02960

[37]. Wang, S., Zhu, T., Liu, B., Ding, M., Guo, X., Ye, D., & Zhou, W. (2024). Unique Security and Privacy Threats of Large Language Model: A Comprehensive Survey. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2406.07973

[38]. Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z., & Zhang, Y. (2024). A survey on large language model (LLM) security and privacy: The Good, The Bad, and The Ugly. *High-Confidence Computing*, *4*(2), 100211. https://doi.org/10.1016/j.hcc.2024.100211

[39]. Yip, D. W., Esmradi, A., & Chan, C. F. (2023). A Novel Evaluation Framework for Assessing Resilience Against Prompt Injection Attacks in Large Language Models. *2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, *71*, 1. https://doi.org/10.1109/csde59766.2023.10487667

[40]. Zangana, H. M., Mustafa, F. M., & Li, S. (2024). Large Language Models in Cybersecurity. In *Advances in information security, privacy, and ethics book series* (p. 277). IGI Global. https://doi.org/10.4018/979-8-3373-1102-9.ch009

[41]. Zhang, X., Lyu, D., & Li, X. (2025). *Risk Assessment and Security Analysis of Large Language Models*. https://doi.org/10.48550/ARXIV.2508.17329