Noname manuscript No.
(will be inserted by the editor)

Securing Educational LLMs: A Generalised Taxonomy of Attacks on LLMs and DREAD Risk Assessment

Farzana Zahid · Anjalika Sewwandi · Lee Brandon · Vimal Kumar · Roopak Sinha

Received: date / Accepted: date

Abstract Due to perceptions of efficiency and significant productivity gains, various organisations, including in education, are adopting Large Language Models (LLMs) into their workflows. Educator-facing, learner-facing, and institution-facing LLMs, collectively, Educational Large Language Models (eLLMs), complement and enhance the effectiveness of teaching, learning, and academic operations. However, their integration into an educational setting raises significant cybersecurity concerns. A comprehensive landscape of contemporary attacks on LLMs and their impact on the educational environment is missing. This study presents a generalised taxonomy of fifty attacks on LLMs, which are categorized as attacks targeting either models or their infrastructure. The severity of these attacks is evaluated in the educational sector using the DREAD risk assessment framework. Our risk assessment indicates that token smuggling, adversarial prompts, direct injection, and multi-step jailbreak are critical attacks on eLLMs. The proposed taxonomy, its application in the educational environment, and our risk assessment will help academic and industrial practitioners to build resilient solutions that protect learners and institutions.

Keywords Cyber attacks · Large language models (LLMs) · Risk Assessment · DREAD. · Education

Farzana Zahid
University of Waikato, Hamilton, New Zealand.
E-mail: farzana.zahid@waikato.ac.nz

Anjalika Sewwandi
University of Waikato, Hamilton, New Zealand.
E-mail: ap623@students.waikato.ac.nz

Lee Brandon
University of Waikato, Hamilton, New Zealand.
E-mail: xl443@students.waikato.ac.nz

Vimal Kumar
University of Waikato, Hamilton, New Zealand.
E-mail: vimal.kumar@waikato.ac.nz

Roopak Sinha
Deakin University, Melbourne, Australia.
E-mail: roopak.sinha@deakin.edu.au

1 Introduction

LLMs are designed for understanding and generating natural language text and solving complex tasks [1–7]. These models utilise deep learning algorithms characterized by a vast number of parameters, and are trained on massive datasets to understand the relationship and trends among the linguistic constructs. With the advent of advanced LLMs like PaLM [8], LLaMA [9], Gemini [10], Falcon [11], GPT and its versions specifically GPT-3 [12] and GPT-4 [13], DeepSeek [14] and others [7], these models mark a paradigm shift within numerous sectors. From finance to healthcare, and manufacturing to education, LLMs are playing a major role in innovations, streamlining processes and redefining standards, and achieving human-level performance in applications like dialogue management, text translation, and virtual assistance [2–5, 15, 16].

Educational Large Language Models (eLLMs) support learning in many ways, such as personalised learning experiences across space and time, content generation, automatic grading, feedback, research assistance, scheduling, assessment evaluation, real-time problem solving, and other institutional support [15, 17–19]. Global Market Insights has predicted that the AI education market, particularly using LLMs, will reach $20 billion by 2027 [20, 21]. Unfortunately, LLMs have also sparked widespread cybersecurity concerns in education [18, 22]. The increasing use of eLLMs expands the attack surfaces and the entry points an attacker can use to com-

promise educational institutions [17, 20]. Although education is not the sector most driven by financial gain, the large amount of personal data (student, employee, and institutional), intellectual property, research data, and lack of adequate security measures make them a target for cyber-attacks [23, 24]. This is evident by the surge in the number of cyber-attacks on education recently [25, 26].

Ensuring security in educational workflows driven by eLLMs is a continuous endeavour that requires dealing with the increasingly sophisticated cyber-attacks on eLLMs models and their infrastructure. These attacks directly or indirectly impact the integrity of learning materials, public trust of educational institutions, privacy and security of staff, students, or associated stakeholders' information, academic operational continuity, and financial sustainability of educational institutions. Thus, it is crucial to understand the attack surfaces, techniques, tactics, and potential attack vectors utilised by the attackers to ensure that eLLMs operate as intended.

A scan of the current literature shows that despite the growing number of studies related to security issues in LLMs, there is a need for in-depth analysis of attacks in terms of their level of sophistication, which could be helpful in understanding the evolving attack landscape and ascertaining effective defensive mechanisms. Furthermore, only a few of the existing works focus on risk analysis, while none emphasise the importance of evaluating risks related to critical sectors like education.

This article introduces a generalised taxonomy of cyber-attacks on LLMs and analyses the criticality (severity) of the identified attacks in education. We conduct a Systematic Literature Review (SLR) [27] to identify the current security issues within LLMs. This SLR study explores the following research questions.

- RQ1 What are the key security attacks on LLMs?
  -RQ1.1 How can the attacks identified in RQ1 be characterised by the level of sophistication?
  -RQ1.2 How can the characterisation, resulting from RQ1.1, assist in recognising critical attack vectors and general impact of attacks on LLMs?
- RQ2 How could the attacks identified in RQ1 be evaluated and prioritised within the education sector?

Following the SLR, we propose key classification criteria to categorise cyber-attacks on LLMs based on their level of complexity/sophistication, an area not yet explored in existing studies. We also examine related attack vectors and their impacts. Section 3 presents these findings as a generalised taxonomy of security attacks on LLMs. In Section 4, we apply this taxonomy to eLLMs. To systematically identify critical security risks in this domain, we adopt the widely accepted DREAD (Damage, Reproducibility, Exploitability, Affected Users, and Discoverability) risk assessment framework [28, 29]. The DREAD criteria are independent (i.e., not correlated) and straightforward in both application and interpretation, making them well-suited for identifying and addressing high-priority eLLMs security risks before exploitation, resulting in optimal business and technical impact [30].

The primary contributions of this study are:

1. A systematic literature review of the up-to-date security attacks on LLMs, presented in Section 2.
2. A generalised attack taxonomy on LLMs, based on their level of complexity, detailed in Section 3.
3. An application of the proposed taxonomy to eLLMs and DREAD-based analysis and quantification of security risks, attack vectors and their impact in Section 4.

2 Systematic Literature Review

A Systematic Literature Review (SLR) conducted includes the following phases: planning, conducting and reporting the review [27]. The Covidence tool was used to ensure clear reporting for a systematic review [43].

2.1 Planning

Scope definition and formulation of research questions: To answer RQ1, a comprehensive literature review is conducted to identify and scrutinise existing research works within the area of Large Language Model (LLM) security. The scope of this research is to investigate and analyse security attacks on LLMs. We propose an easy-to-understand generic taxonomy of attacks on LLMs based on the level of attack sophistication (attack complexity), analyse the various attack vectors for each identified attack, and determine the impact of those attacks. Moreover, we also quantify the risks posed by each attack in the education sector. Our SLR also identifies several secondary studies in this area [4, 5, 31–39, 41], but these studies differ significantly in their focus and methodology, as shown in Table 1.

Based on the scope of our study, we formulated the research questions mentioned in the Section 1.

Database selection and search query: IEEE Xplore, SpringerLink, and Scopus were selected for this study. Scopus, the largest commercially accessible database of peer-reviewed articles, also encompasses IEEE Xplore

Table 1 Comparison of our Work with the Existing Secondary Studies by Research Type (Survey, Exploratory Study, Empirical Study, Opinion Paper), LLMs Types (General, ChatGPT, Gemini or others), Attack Taxonomy (Yes/No), Attack Vectors (Yes/No), Attack Impact (Yes/No), Risk Analysis (Method/No)

References	Type of Research	Types of LLMs	Attack Taxonomy	Attack Vectors	Attack Impact	Risk Analysis
[31]	Survey	General	No	No	No	No
[32]	Survey	General	No	No	Yes	No
[5]	Survey	General	No	Yes	Yes	No
[33]	Survey	ChatGPT	Based on platform plugins	Yes	No	No
[34]	Survey	General	Based on backdoor Attacks	Yes	No	No
[35]	Survey	ChatGPT	No	No	Yes	No
[36]	Survey	General	No	No	Yes	OWASP
[4]	Survey	General	No	No	No	No
[37]	Empirical Study	ChatGPT	No	No	Yes	No
[38]	Exploratory Study	ChatGPT	No	No	Yes	No
[39]	Survey	General	No	Yes	No	Benchmark datasets
[40]	Survey	General	Based on jailbreak	Yes	No	No
[41]	Survey	General	Based on prompt injection, jailbreak,data poisoning	No	Yes	No
[42]	Survey	General	No (brief discussion on prompt injection, jailbreak,backdoor)	No	No	No
Our work	*Survey*	*General*	*Based on Attack Complexity*	*Yes*	*Yes*	*DREAD*

and SpringerLink. Nonetheless, we conducted individual searches of all these databases to ensure completeness.

For filtering out the primary studies, we follow the methods reported in [44], and identified the following security-related keywords - `secur*`, `attack`, `threat`, `vuln*`, and `risk`. For LLMs, we used terms like `large language models`, and `LLM`. The following query string in the Scopus format represents the final combination of the above keywords/phrases used in our SLR:

```
(TITLE-ABS-KEY (large AND language AND
model OR "LLM" OR "Large language model")
AND TITLE-ABS-KEY (secur*)
OR TITLE-ABS-KEY (threat) OR
TITLE-ABS-KEY (vuln*) OR
TITLE-ABS-KEY (risk))
AND PUBYEAR > 2019 AND PUBYEAR < 2025
```

Inclusion/Exclusion criteria: The inclusion criteria (`INC`) and exclusion criteria (`EXC`) that were utilised for the selection of only the relevant studies from the search results are represented in Table 2.

Table 2 Inclusion and Exclusion Criteria

Inclusion Criteria	INC1	Studies that investigated LLMs security issues.
	INC2	Studies that discuss the concepts like open challenges, problems related to the security issues within LLMs
	INC3	Studies published in conferences, journals, technical reports, pre-prints (as most of the recent articles are shared as pre-prints)
	INC4	Research studies that appeared since 2020 till now
	INC5	Studies that focused on the real-world applications of LLMs
	INC6	Studies that include at least one of the specified keywords
Exclusion Criteria	EXC1	Studies where title, keywords and/or abstract do not lie within defined scope
	EXC2	Studies do not investigating any security issues within LLMs
	EXC3	Studies that address solely the concept of privacy issues of LLMs
	EXC4	Studies focusing on attacks that could be launched using LLMs
	EXC5	Studies that do not have full text
	EXC6	Books, thesis, tertiary studies, tutorial and opinion papers
	EXC7	Studies not written in English
	EXC8	Studies whose new version is available or are not peer-reviewed

2.2 Conducting the review

Search and data extraction: For this study, we performed both automated (using Covidence) and manual searching.

Figure 1 shows the search execution chronology (Covidence's PRISMA flow diagram). The initial automated search resulted in 1542 articles. Due to duplicate records and screening of the titles, keywords, or abstracts, a significant number of studies were excluded, leaving 816 studies for eligibility selection. A further 724 studies were removed after carefully examining each study's introduction, conclusion, and full text. The number dropped to 60 after meta-analyses. Data extraction from automatic search includes the identification of the keywords by reading abstracts, introductions, and conclusions (if needed) [44]. In the case of manual search, ATLAS [45], AI Incident Database [46] and OWASP framework [47], and white papers related to the security issues of LLMs

were selected. As there is a continuous rise in the number of security-related studies on LLMs, manual search plays an important role in enhancing the confidence of the comprehensiveness of the review. For manual search, data extraction is performed using the keywords selected for our query string. Furthermore, scanning the manual results for the attack scenarios and extracting the utilised tactics, techniques, and sub-techniques for particular attacks on LLMs resulted in including ten more relevant articles.

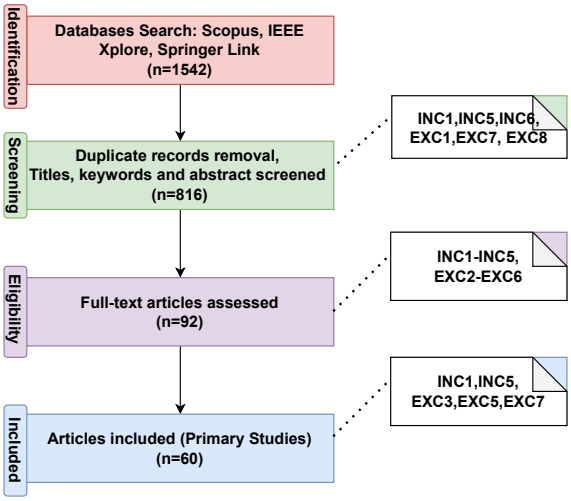


Fig. 1 Search execution chronology

Quality Assessment: Our quality assessment criteria contains five measures QC1–5 [48]. Each study is assigned a score of 0, 0.5, or 1 for each measure. The final quality score for a study is the aggregation of the individual scores, which is quantified as low when $0.5 \leq$ quality score ≤ 2, medium for $2.5 \leq$ quality score ≤ 3, or high ($3.5 \leq$ quality score ≤ 5).

Based on the measures mentioned above, the scoring for our SLR is calculated as follows.

`QC1.` Our study clearly states the objective of the research, and so it gets a score of 1.

`QC2.` A score of 1 is assigned as Inclusion and Exclusion criteria are defined.

`QC3.` A score of 1 for presenting an explicit synthesis method based on a well-used methodology.

`QC4.` The quality assessment of selected primary studies was performed, but not reported, so our study gets a score of 0.5.

`QC5.` A score of 1 for providing information about each primary studies.

Overall, our SLR scores 4.5, indicating it is a high-quality review.

2.3 Reporting (Threats to Validity)

Overlooking Important Relevant Studies: An SLR is intended to cover the depth of a research area by analysing the existing works in that area. There is a chance of overlooking some relevant current literature, so the query string is formulated to retrieve the maximum number of studies from the databases. The titles and abstracts of the articles could also be ambiguous. Therefore, we thoroughly read the introductions, conclusions, and full-texts, if needed, to ensure the inclusion of relevant primary studies. Moreover, selecting the specific databases could also result in missing studies; hence, a manual search was also performed to mitigate this threat. We also used the Covidence tool for SLR. The use of the tool and manual search therefore provides another layer of assurance and helps find articles that may be missed due to the use of non-standard terminology.

Researcher Bias: Researcher bias could impact the validity of research. The systematic literature review protocol was established and followed carefully with the support of domain experts and co-authors.

Selection of the Query String: The final selection of the primary studies depends on the scope, novelty of research areas, and search strings. To create a query string that could not be very strict and define the scope, we tweaked it to remain comprehensive (return all relevant papers) while reducing the number of irrelevant papers returned. For example, with the keyword `risk` in the query string, we omitted keywords like risk `analy*`, `mitig*` and `assess*` intentionally because even by including these terms, the result of retrieving the number of studies does not change. Also, including keywords such as language model, natural language processing, natural languages, deep learning, machine learning, and Generative AI reduces the number of articles retrieved to only 152. Similarly, the keyword related to LLMs in education results in numerous irrelevant papers, while the other relevant papers were already filtered out using our final query mentioned in Section 2.1.

Subjectivity in DREAD scoring: Later on in Section 4, we present a scoring scheme based on the DREAD model [28] to assess the potential risk of a wide variety to attack types on eLLMs. Due to the absence of empirical data and validated statistics in this new domain, there are potential internal validity threats due to subjectivity and researcher bias. As explained in Section 4.2, we reduce these biases by drawing on cross-sector findings and by taking inputs from experts, however

more work is needed to strengthen this aspect of the work.

3 Taxonomy of Security Attacks on Large Language Models (LLMs)

LLMs are susceptible to various attacks (Figure 2), which are either on their models (parameters, hyperparameters, model input, test data, training data, model documentation) directly or their infrastructure (deployments, storage, network, servers, hardware). We propose a novel taxonomy to classify the identified attacks on models and associated infrastructure based on the *attack complexity* or sophistication level.

Sophistication Level (Attack Complexity): Attack complexity or the level of the sophistication indicates the extent of the actions that must be taken by the attacker to compromise LLMs. The actions are categorised into three criteria: *(1) need of specialised skills, (2) knowledge of the model and/or infrastructure (3) ease of exploitation (operational steps).* Based on these criteria, we have assigned the following attack complexity metric to each attack on LLMs mentioned in the selected primary studies.

1. **High (H)**: The level of sophistication is *high* if the attacker (1) has to use specialised or advance tactics, skills, methods or tools to compromise the LLMs or (2) needs an in-depth knowledge of the LLMs model or infrastructure, or (3) requires multiple coordinated steps or integration with external services. These attacks are indicated by the color red in Figure 3 and Figure 4.

2. **Medium (M)**: The attack complexity is *medium* in the cases: when the attacker (1) requires moderate or less specialised techniques or tools to compromise the LLMs (2) performs 2-3 coordinated steps in a sequence (3) doesn't need full model/infrastructure knowledge. These attacks are represented in yellow in Figure 3 and Figure 4.

3. **Low (L)**: The level of sophistication is *low* (illustrated in green color in Figure 3 and Figure 4) in three cases: if the attacker (1) carries out the attack in a single step (maliciously craft simple or direct prompt), (2) require less or no expertise (3) does not require the internal knowledge of the model/infrastructure to launch an attack on LLMs. These attacks are illustrated in green in Figure 3 and Figure 4.

When the attack exhibited characteristics that fall into more than one metric, such as a single step attack but needs detailed knowledge of model, we assigned the metric by focusing on the dominant criteria as evidenced in the literature. The detail of each type of attack is presented in following sections. The summary of attack vectors and impacts is given in Table 4.

3.1 Security Attacks on LLMs Models

This section discusses various security attacks on the LLM models, as presented in Figure 3, along with their corresponding vectors and impacts. Table 3 further elaborates on the Figure 3 by presenting the complexities of each identified attack, determined using the previously mentioned sophistication level metric. Also, Table 4 summarises the associated vectors and the specific impacts of each attack.

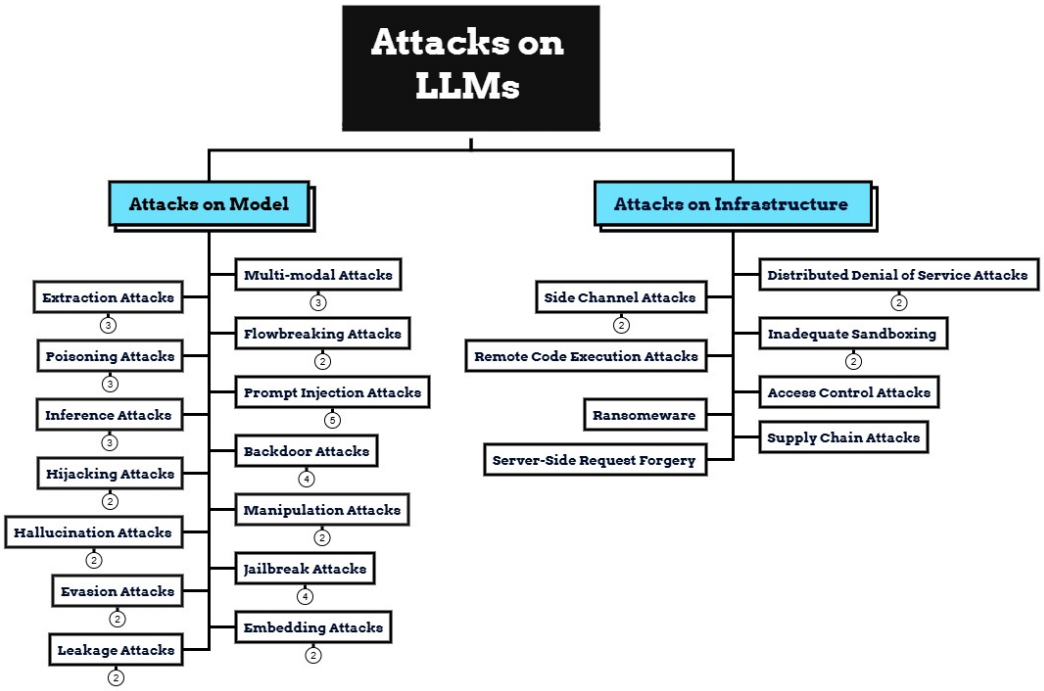
Multi-Modal Attack: A multi-modal attack on LLMs is an adversarial attack to exploit the processing and understanding capabilities of LLMs when dealing with different input types [49, 50]. To launch this attack, the attacker uses various attack vectors such as adversarial images, crafted noise or text with images, poisonous association to manipulate the LLMs' generated output or their operational behaviours [32]. These attacks could take many forms like text guided image generation, cross-modal attack and adversarial image.

In *text guided image generation* attack, an intruder embeds malicious text with an image to trigger the LLMs for the generation of the manipulated image with malicious intent like spreading misinformation [51, 52]. The attacker could either use pre-trained models or should have a knowledge of basic prompt engineering and with trial and error, he can crafts prompts to generate misleading output. Therefore, the attack complexity of this attack is low (L).

In *cross-modal attacks*, attackers exploit vulnerabilities in the interaction between different modalities, such as, text, audio, images, etc. to confuse the model [53, 54]. An attacker should have some knowledge of the model such as, pertaining to the linkage of text and images, and needs knowledge of prompt engineering to introduce minor inconsistencies in the modalities. Also, attackers need multi-step procedures to launch this attack, first they exploit the vulnerability of one modality (such as audio) and then utilise that compromised modality to trigger a specific response from the LLM. Therefore, the attack complexity of this attack is medium (M).

Adversarial image attacks involve subtle modifications to images using hidden patterns such as imperceptible noise or minor pixel-level adjustments that mislead LLMs to trigger the specific response [32, 54, 55]. The attack complexity is high (H) because the attacker adds adversarial perturbations in the images, so they need

Fig. 2 A Generalised Taxonomy of Attacks on LLMs



high expertise and in-depth knowledge of the model to launch this attack.

Flowbreaking Attack: Flowbreaking is a newly introduced/novel LLM attack that targets the reasoning and coherence of LLM models while generating the response [56, 57]. Compared to input data manipulation attacks, the internal logic of the model's output is disrupted by flow-breaking attacks. Even benign prompts can lead the model to produce incorrect or harmful responses or result in information disclosure. There are two types of flowbreaking attack: second thoughts, and stop and roll [56, 58].

A *second thoughts* attack occurs when LLMs models initially provide a response to the prompt but halt or retract upon detecting a sensitive topic and either generate a simple error message or new modified content; attackers exploit this behavior to extract sensitive information. This attack requires prompt engineering skills and some knowledge of the model to exploit the guardrails (filters) or streaming window. Therefore, the attack complexity is medium (M).

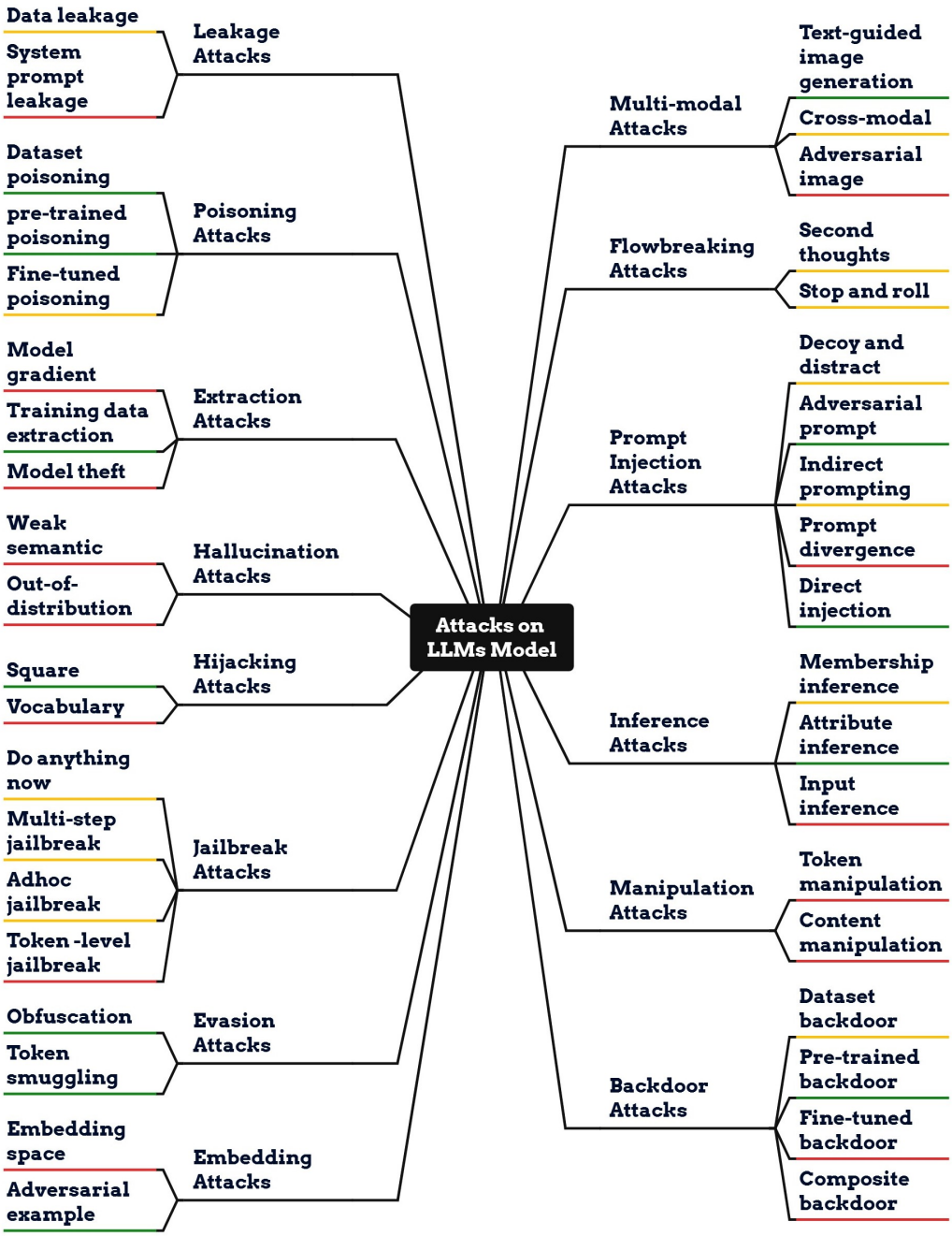
A *stop and roll* attack involves the manipulation of LLM output using hidden commands or crafting specific prompts. During the answering phase, the attacker breaks the flow of the LLM's reasoning by pressing the stop button, but the answer is still streaming and cannot be deleted. This attack results in unauthorized actions, information disclosure, or potentially damaging responses even though the system policies are violated. The ease of exploitation is simple using the stop button. However, the attacker needs model knowledge and some expertise to craft the specific instructions, so the complexity of the attack is medium (M).

Prompt Injection Attack: In the prompt injection attack, malicious prompts replace the LLM's original instructions, manipulating them to respond to different queries rather than fulfill their intended function [3, 59–61]. There are various forms of prompt injection attacks, such as adversarial prompt injection attacks, decoy and distraction prompt injection attacks, indirect prompt injection attacks, prompt divergence attacks, and direct prompt injection attacks.

An *adversarial prompt injection* attacker exploits the model's instruction-following behaviour to mislead LLM's intended response by directly adding adversarial instructions. Adversarial prompt injection, thus, can result in undesirable or unauthorised outputs, such as offensive responses or unauthorised data disclosure [62]. Attack vectors are user input fields or APIs that allow external text input to the LLMs. In this attack, attackers need basic prompt engineering skills to craft effective phrases without requiring detailed knowledge of the model's internal workings. The adversarial prompt injection attacks are straightforward due to LLMs' tendency to follow prompts precisely [62]. Thus, the attack complexity is low (L).

Likewise, in *decoy and distraction prompt injection attacks*, the attacker misguides the LLMs by embedding

Fig. 3 A Breakdown of Taxonomy of Attacks on LLMs Models



off-topic or confusing information within the prompt [61, 62]. This decoy information causes LLMs to prioritise the distraction, shifting the model's attention from the main question to the misleading content, leading to inaccurate or off-topic outcomes [63]. The attacker must be familiar with prompt structuring and model's prioritising to place distractions into the input. As a result, decoy and distraction prompt injection attacks have medium complexity (M) because they require an understanding of prompt dynamics to mislead the model accurately [63]. The attack vectors are frequently seen in chat interfaces or information retrieval systems where off-topic details can be embedded easily [62].

In an *indirect prompt injection attack*, adversarial prompts are inserted into retrievable data sources that the LLMs access [63–65]. This causes the model to process compromised data, which may result in unintended responses or data leaks. Because these indirect prompt

injection attackers exploit the model's data retrieval processes, LLMs unknowingly execute embedded commands from external sources, allowing the attacker to remotely manipulate outputs without requiring direct prompt input. Indirect prompting leads to excessive agency vulnerability [59]. Thus, an attacker may have some knowledge of the model and expertise to effectively understand the model's data retrieval mechanisms and manipulation of external data by placing malicious prompts (attack complexity is medium (M)). Attack vectors can be external data sources such as websites, APIs, or shared documents that the LLMs may access during the data retrieval [64].

Furthermore, in *prompt divergence*, attackers embed ambiguous or conflicting instructions within the prompt, causing the LLM to interpret these instructions in ways that produce divergent responses to deviate from LLM's original goal [66, 67]. The successful execution requires an understanding of the prompt structure and model processing to create conflicting contexts, which are complex methods requiring specialised skills [68]. In addition, attack vectors can be structured applications in multi-step tasks with diverging instruction encoding techniques, depicting higher operational complexity. Therefore, the attack complexity of this attack is high (H).

In *direct prompt injection*, attackers append adversarial commands directly (in a single step) to the system prompt, overriding the LLM's intended functionality [3]. The attacker needs minimal technical knowledge about LLMs to append malicious crafted string [62], and does not need model or infrastructure knowledge, leading to unintended or unsafe outputs. Direct prompt injection attack is a single-step attack and categorised as low (L) complexity [62].

Embedded Attack: In an embedded attack, the attackers craft malicious instructions or manipulate the tokens to carry out harmful actions, e.g., output hazardous knowledge [69] to change the workflow of LLMs [70]. These attacks can be adversarial example attacks or embedded space attacks [71].

In the *adversarial example attack*, attackers craft small changes to the input that are undetectable for the user but can trick or confuse the model into making erroneous predictions [72]. The adversarial example attacker can make small perturbations to the input using basic understanding of input crafting without needing the internal knowledge of LLMs [72]. Therefore, an adversarial example attack has a low complexity (L).

Embedded space attacks, on the other hand, modify the embedding layer of open-source LLMs by passing the input string via a tokenised process [71]. The user cannot see these modifications since they are hidden in the LLM' embedded layers. Attackers utilising embedded space must be proficient in open-source LLMs and able to transform input text into hidden token/word representations through specific tactics like gradient descent [71]. Thus, embedded space attacks are complex as attackers follow specific tactics and skills in modifying LLMs' embedding, resulting in high complexity (H).

Jailbreak Attack: Jailbreak attacker bypasses LLMs' safety guardrails to respond to unsafe or restricted questions and output inappropriate content such as malware, scams, and illegal or socially harmful instructions [32, 49, 60]. Jailbreak attacks come in different forms, such as Do Anything Now (DAN) mode, multi-step prompt, ad-hoc jailbreak and token level jailbreak attacks.

In *DAN mode jailbreak prompt attack*, LLMs are being forced to undertake dangerous, harmful actions like "Do Anything Now" (DAN) mode, preventing them from completing their intended task [63]. This means the attacker prompts the model to act unrestrictedly, effectively using role-playing instructions and unlocking capabilities limited by safety protocols [98]. The DAN jailbreak attacker requires creativity in prompt construction and an understanding of role-play dynamics, instead of a deep technical background. Thus, DAN jailbreak attacker does not need high skills (attack complexity is M) because it uses creative framing to exploit the model's flexibility with user roles, relying more on inventive prompting than in-depth model knowledge.

In *multi-step jailbreak attacks*, the LLM model's filters are gradually weakened by a series of prompts, eventually reducing the chance that the model will follow its limitations, such as red queen attack [75]. Therefore, the attacker does not need specific tools to comprehend how prompts sequencing interacts over a number of steps and how contextual layering can affect model behaviour, which is regarded as medium-level complexity (M) [99].

Ad-hoc jailbreak attacks are characterised by improvised, creatively crafted prompts to bypass the model's restrictions [32]. The attacker directly inputs crafted prompts that manipulate the model into providing restricted information. Techniques include hypothetical scenarios, attention-shifting, and context manipulation, where the prompt creates a scenario or role-play that bypasses ethical guidelines. The ad-hoc jailbreak attacker skill requirement is generally low because it may mainly involve basic prompt phrasing without deep technical understanding. However, some prompts may require insight into how models interpret instructions,

Table 3 Attacks on LLMs Model and Attack Complexities

Attacks	Attack Sub-types	Required Skills	Knowledge of Model/ Infrastructure	Ease of Exploitation	Attack Complexity
Multi-Modal	Text guided image generation [51, 52]	L	L	L	L
	Cross-modal [53, 54]	M	L	M	M
	Adversarial image [32, 54, 55]	H	M	H	H
Flowbreaking	Second thoughts [56–58]	M	M	M	M
	Stop and roll [56–58]	M	M	L	M
Prompt Injection	Adversarial prompt [62]	M	L	L	L
	Decoy and distract [61, 62]	M	M	L	M
	Indirect prompting [63–65]	M	M	L	M
	Prompt divergence [66, 67]	H	H	H	H
	Direct injection [3]	L	L	L	L
Leakage	Data leakage [73]	M	L	M	M
	System Prompt leakage [74]	H	H	M	H
Embedding	Embedding space [71]	H	H	H	H
	Adversarial example [72]	L	L	L	L
Jail break	Do anything now [63]	M	L	M	M
	Multi-step jailbreak [75]	L	M	M	M
	Adhoc jailbreak [32]	L	M	M	M
	Token-level jailbreak [76]	H	H	H	H
Backdoor	Dataset backdoor [34, 77]	L	L	M	M
	Pre-trained backdoor [78]	L	L	L	L
	Fine-tune backdoor [34]	H	H	H	H
	Composite backdoor [79] [80]	H	H	H	H
Poisoning	Dataset poisoning [81]	L	L	L	L
	Pre-trained poisoning [82]	L	L	L	L
	Fine-tune poisoning [82],[83],[71]	M	L	M	M
Inference	Membership inference [78]	M	M	L	M
	Attribute inference [84, 85]	L	L	L	L
	Input inference [86]	H	H	H	H
Manipulation	Token manipulation [87]	H	H	H	H
	Content manipulation [61, 88]	H	H	H	H
Extraction	Model gradient [89]	H	H	H	H
	Training data extraction [90]	L	L	L	L
	Model theft [89]	H	H	H	H
Evasion	Obfuscation [91]	L	L	L	L
	Token smuggling [83, 91, 92]	L	L	L	L
Hallucination	Weak semantic [68]	H	H	H	H
	Out-of-distribution [93]	H	H	M	H
Hijacking	Square [94, 95]	L	L	L	L
	Vocabulary [96],[97]	H	H	H	H

relying on creativity and user insight rather than technical expertise [100]. Overall, this attack has medium (M) complexity.

Attackers that use *token manipulation jail breaks* take advantage of particular tokens, often anomalous or special text, that the model processes in unusual manners. By employing tokens or sequences that trigger behaviors inconsistent with the model's intended aim, these attacks take advantage of the way models interpret tokenised input [76]. Tokenisation has unique effects on model behavior; special tokens like or are frequently used to exploit it. These tokens cause the model to output responses that bypass restrictions, possibly due to the special token's influence in the tokenisation or generation process. This attack requires understanding tokenisation, generation mechanisms, and specific token functions within the LLM's architecture. The attackers require high skill and technical knowledge to manipulate tokens to achieve specific behavior within the LLMs, making the overall attack complexity high (H) [76].

Poisoning Attack: Poisoning attacks influence the integrity of the training data; attackers can introduce deliberately manipulated data into the model's training phases [101]. There are different classifications for poisoning attacks, such as pre-training poisoning, dataset poisoning and fine-tuning poisoning.

In *pre-training poisoning*, attackers can inject malicious or biased content into public internet sources such

as Wikipedia, a widely used resource for LLMs training. This poisoned data may included in the LLM's initial training set. Since LLMs rely on massive datasets, minor edits are hard to detect but can substantially affect LLMs' behaviour [82]. Thus, the complexity of this attack is low (L) as the attackers do not need task-specific knowledge or high skills [101].

In *dataset poisoning*, attackers add harmful or biased content to specific datasets devised for a particular application (e.g., medical sector) domain [81]. Thus, dataset poisoning limits selected domains by adding biased or misrepresentative content to carefully selected datasets. This can skew the LLM's behaviour in specific applications, especially where data curation is imperfect, leading to slight but impactful biases in task-specific LLMs outputs [101]. Attackers do not need to manipulate large datasets, but must understand the basic curation process. Thus, knowledge of the dataset's intended application helps attackers insert biases that may not be easily detected in specific domains [82]. The impact of the dataset poisoning attack is typically limited to specific tasks or fields where curated datasets are applied. Thus, the attack complexity is medium (M). However, it can create severe biases or misinformation in sensitive applications (e.g., health or legal), affecting user trust in model outputs.

Fine-tuning poisoning attack targets the final training phase, where LLMs are fine-tuned for specific tasks or applications. Attackers introduce harmful data to override safety and alignment features, often embedding backdoors that activate with specific input triggers [83]. Attackers exploit fine-tuning APIs to insert carefully crafted triggers or backdoors, which modify the model's behaviour upon receiving specific prompts. This approach allows attackers to bypass moderation controls by embedding hidden behaviours that activate only under specific conditions [82]. Therefore, fine-tuning poisoning attackers requires specific API knowledge to evade moderation controls effectively and balance subtlety with effectiveness, creating hidden triggers that only activate when intended [71]. Overall, the attack complexity is high (H).

Evasion Attack: In an evasion attack, the attacker crafts fake samples during the inference phase, which is not noticeable but leads to incorrect/unexpected behavior [83, 91, 92]. Evasion attacks have different forms, such as obfuscation or token smuggling.

In an *obfuscation attack*, an input text is manipulated using word-level or character-level subtle perturbation [91]. The attacker changes the input in several ways, such as by replacing a single word or certain words with similar words, adding special characters, or altering the sentence's structure. Collectively, these techniques make the model confusing, making it difficult for the model to recognize the intended meaning of the input. In this attack, the attacker does not have the authority to change the model's architecture or its parameters. Therefore, to evade the detection, the attacker needs basic obfuscation tactics only, and with trial and error, the attacker could result in harmful, or restricted contents. Overall, the attack complexity of the obfuscation attack is low (L).

Token smuggling attack comprises the banned words, which are encoded in the attacker's input to evade the filters or detections. The purpose of the attack is to alter the behavior of the model to produce the incorrect output. The complexity of this attack is also low (L) as to launch this attack, the attackers only need to simply split the words and do not need in-depth knowledge of the model.

Extraction Attack: In extraction attacks, attackers use model training data or extract the specific LLM architecture and parameters and recreate the model for execution [89]. The attackers can leverage the target LLM by supplying prompts refined to induce the LLM to perform the intended task (e.g., summarisation, chat-based responses, question answering, etc.). This refined prompting process enables attackers to effectively refine and transfer the task-specific capabilities into the extracted model for their purposes [89]. Extraction attacks could be of several types such as model gradient attacks, training data extraction and model theft.

In *model gradient attacks*, attackers use precise gradient based training to recreate the model because generally malicious actors cannot steal highly valuable models, such as those trained on rare or hard-to-obtain datasets. This attack poses a significant threat, as it enables the theft of cloud-hosted models without requiring input data. Consequently, such attacks have high (H) complexity as attackers require having an in-depth understanding of the LLM model and infrastructure [102].

Training data extraction attackers can exploit LLMs which are trained on private datasets. By querying the language model, they can recover individual training samples, extracting verbatim sequences from the model's training data using only black-box query access. This enables attackers to retrieve (publicly available) personally identifiable information (e.g., names, phone numbers, and email addresses) and other non-sensitive information [90]. Thus, this attack is accomplished in a single step, and the complexity of this attack is classified as low (L).

Model theft attack is a black-box adversarial attack. Attackers create an extracted model by deriving specific

features (e.g., architecture, parameters, and hyperparameters) from the target model of interest, enabling them to reconstruct it. Once the extracted model is established, attackers can carry out further adversarial attacks, such as model inversion, membership inference, privacy data leakage, and model intellectual property theft [89]. To execute a model theft attack, attackers require extensive knowledge of LLMs to perform several critical steps such as prompt design for crafting prompts to attain task-specific LLM responses, data generation to derive extracting model characteristics, extracted model training for model recreation and ML attack staging against a target LLM [89]. Due to the complexity and technical depth involved, the complexity of a model theft attack is classified as high (H).

Backdoor Attack: The concept of a backdoor attack is to inject triggers (short phrases, prompts, or instructions) into models, including LLMs [34]. The attacker inserts triggers into a specific section, such as an open-source library, poisoned training data, etc. [78, 80]. When user inputs are triggered, the model will output some specific contents by the attacker [80]. The backdoor attacks are of various types, including dataset backdoor, pre-trained backdoor, fine-tuned backdoor, and composite attacks.

In the *dataset backdoor*, attackers deploy poisoned training data in an open-source library. If some LLM developers utilise it to train their models, they may unknowingly embed a hidden backdoor in the model and open it for manipulation by attackers. This attack doesn't require comprehensive technical skills. The dataset backdoor, however, needs the attacker to perform multi-steps to launch an attack. Hence, the attack complexity of this attack is medium (M) [34, 77].

In the *pre-trained attacks*, the attacker is assumed to be an untrusted third-party service provider and offers (or open-source) pre-trained LLMs, which are tailored to specific targets (such as datasets or prompt templates) designed to attract potential users. Consequently, the attacker has complete control over the training dataset and the training process of the target model, so attacker does not require internal knowledge of the model and does not require specific expertise to launch an attack (attack complexity is low (L)) [78].

The *fine-tuned backdoor attack* is similar to a dataset backdoor, but attackers need more technical skills. For example, attackers embed a backdoor into a model and upload it to the Internet, waiting for some unsuspecting victims (like developers) to download this model. The difference is when the developer proceeds with models to fine-tune them for a specific purpose, the attacker operates in a white-box environment and modifies the model's parameters, structure, and training data [34]. These vectors require the attacker to be very familiar with the trained model and skilled in model training. Thus, fine-tuning backdoor attack needs in-depth knowledge and highly complex skills for launching an attack on the LLMs. Overall, sophistication level of this attack is medium (M).

For the *composite backdoor attack* (CBA), attackers need to set up triggers just like any other backdoor attack, but in this attack, multiple trigger keys are scattered across different prompt components [79]. The composite backdoor will only be activated when all trigger keys coincide. The composite backdoor attack is considered more complex and requires advanced expertise, and the attacker must understand the model's internal workings to execute it effectively [80]. Therefore, the attack complexity level is high (H) as composite backdoor attacks require high skill and technical knowledge.

Inference Attack: In inference attacks the attacker's motive is to illegitimately retrieve the victim's sensitive information from the LLMs [103]. LLMs tend to memorise information from their training data, and attackers investigate that memorisation of training data. There are three main sub-types of attacks on LLMs: membership inference attacks (MIA), attribute inference (model inversion), and input inference attacks.

Membership inference attack (MIA) is one of the most basic forms of inference attack, which allows attackers to fetch data to determine whether a given sample belongs to a training dataset. The attacker's goal is to determine if a specific data point was used in the training dataset of LLMs by analysing its output, such as memorisation of training data, copyright violations, and test-set contamination [78]. Attackers do not require deep technical background for this attack but only deal with the training dataset and data points. As a result, this is a medium complexity (M) attack.

Attribute inference attack can extract various characteristics of the victims, like ethnicity or gender information from a model, even if this information was not explicitly included in the training data [84, 85]. In this attack, attackers only need to utilise some simple steps to fetch private information. Therefore, the complexity of an attribute inference attack is low (L).

In an *input inference attack*, attackers may need a way to intercept the user input to LLMs, using other methods such as network sniffing, exploiting compromised APIs, backdoor attacks, or combining these techniques in a composite attack strategy [86]. Only after obtaining the user's input data can the attacker carry out an inference attack on sensitive information. Con-

Table 4 Attacks on LLMs, Attack Vectors and Impact of Attacks

	Attacks	Attack Vectors	Impacts
LLMs Model-Based Attacks	Multimodal	Crafted text with an image or sound, adversarial images, dataset, poisoned associations (adversarial perturbation), pre-trained model,crafted noise with audio	Faulty output, model's operation behavior, misinformation
	Flowbreaking	Input prompts, stop button	Information disclosure, unauthorised action, incorrect output
	Prompt Injection	Input fields, website's code, APIs, encoding techniques, system level-prompts, instruction tuning datasets	Data breaches, unauthorised action, erroneous output, misinformation
	Leakage	Malicious code, model confidence scores, system prompts	Data breaches, information disclosure
	Embedding	Crafted instructions, characters or tokens, open source LLMs	Output biases, faulty/erroneous/toxic output
	Jailbreak	Heuristic-based prompting, tokens, simple crafted inputs, hypothetical scenarios with acknowledgment, pretrained model	Tricking LLMs, incorrect output, unauthorised access, personal information leakage, privilege escalation, political propaganda
	Backdoor	Dataset, open source libraries, triggers, entrusted third party service provider, pre-trained model, trigger keys (one or many)	Faulty/incorrect decision, unauthorised access
	Poisoning	Bad data from unreliable sources, large amounts of skewed or biased input, publicly available resources, fine-tuned APIs	Output biases, unethical behavior, faulty/erroneous results, disinformation, misinformation
	Inference	Dataset, query a particular data-point, network sniffing, compromised APIs	Unauthorised access, data breaches, privacy violation, reputational damage
	Manipulation	Tokens, malicious instructions, fine-tuning process	Biased output, customer dissatisfaction, misrepresentation
	Evasion	Fake samples, simple, banned, similar words, special characters	Data breaches, Incorrect output
	Extraction	Datasets, gradients queries, spoofing by trusted parties, biometric, crafted code, open API, model architecture and parameters, poorly configured outputs	Incorrect/unexpected predictions, biased output, information disclosure, financial losses
	Hallucination	Fabricated code libraries, semantic input craft, random tokens	Nonsensical/unfaithful output
	Hijacking attack	Delimiters or instructions, randomised search methods, LLM vocabulary	Information disclosure, false output, unauthorised control, offensive behavior
LLMs Infrastructure-Based Attacks	Unbounded consumption	Numerous crafted inputs, complex and resource-intensive queries, cloud-based AI services	Model service degradation, financial losses, reputational damage, service unavailability, system failure
	Inadequate Sandboxing	Passwords, API keys, files/network access, plugins' permissions, misconfigurations	Unauthorised access and action, cross-system exploitation, privilege escalation, data corruption/loss.
	Access Control	Access control policies, API, file/ network information, social engineering tactics, by-default configurations, arbitrary codes execution on server	Data breaches, privilege escalation, misinformation, harmful output, public distrust
	Supply Chain	Datasets, compromised or outdated libraries or models, pre-trained models, insecure plugins or APIs, misconfigurations, unclear policies or agreements	Service outage, privilege escalation, unauthorised action, data breach, biased output, network disruption
	Side-Channel	Model parameters and architecture information, training data, response time, API, power consumption information	Information disclosure, system exploitation
	Server-side Request Forgery	Inputs, security misconfiguration, internal services access request, API, secured data stores	Unauthorised access, model malfunctioning, data exfiltration, Tempering
	Remote Code Execution	Code, shell commands, arbitrary code execution	Network disruption, system unavailable, unauthorised access
	Ransomware	Publicly available code repositories, databases, CVEs, fine-tuned models	System unavailable, data breach reputational damage, financial losses

sequently, this type of attack requires attackers to have highly advanced technical skills due to its composite nature, so the complexity of input inference is high (H).

Manipulation Attack: LLMs may be vulnerable to potential manipulation attacks, which results into public distrust, reputational damage or biased output or misrepresentation [104]. These attacks allow an attacker to manipulate the model's generated output, enabling malicious samples to evade detection without affecting the overall system performance. Attackers can utilise trusted data sources to inject malicious content so that they can introduce manipulated data into the training pipeline by compromising the data source or intercepting it in transit [105]. The manipulation attacks include two sub-types of attacks, token manipulation and content manipulation. Please note: token-level jailbreak attack could also lie in this category.

In a *token manipulation* attack, the attackers exploit the vulnerabilities in the process of tokenisation through token substitutions, removals, and syntactic reordering to generate incorrect outputs[87]. The attacker needs in-depth knowledge of natural language processing tasks (tokenisation process), and detail knowledge of the model's internal architecture and parameters. This attack could also lie under the category of token level jailbreak. The overall complexity of the attack is high (H).

In a *content manipulation attack*, attackers can manipulate models to generate fake content and spread AI-generated fake news (disinformation) and social bots on social media platforms. They may also use LLMs to produce targeted user outputs, deceiving the public for profit. This attack is relatively easy to execute once attackers gain control of LLMs [61, 88]. Therefore, attackers use specialised methods to craft the ambiguous input and need good understanding of the model behavior to achieve their goals. The complexity of content manipulation attack is classified as high (H).

Leakage Attack: In the leakage attack, LLMs accidentally leak sensitive and confidential information from their training data through the responses. These attacks include two sub-types, data leakage and prompt leakage.

In *data leakage attack*, attackers exploit the model's memorisation of sensitive training data to infer, extract, or misuse private information. Such attacks leverage techniques like membership inference or data extraction to recover portions of the training data, potentially including personally identifiable information (PII) or other confidential content, from the model's outputs [73]. Attackers need advanced expertise, understanding, and skills to launch data leakage attacks on the LLMs. Overall, the attack's sophistication is medium (M).

System prompt leakage attack is a specialised attack targeting LLMs, which, if uncovered, facilitates other types of attacks [59]. Since the functionality and performance of LLM applications heavily rely on the system prompt, which directs the underlying LLMs on what tasks to perform, developers typically keep these system prompts confidential. In this attack, an attacker sends instructions to the target LLM application, and its responses inadvertently reveal the system prompt (such as information describing various roles and permissions, connection strings, or passwords). Unlike jailbreak attacks, the ultimate goal of prompt leakage is to replicate the same, precise system prompt [74]. The prompt leakage attack requires considerable skill to understand the model architecture and prompt engineering, and the attacker can accomplish its motive by trying different instructions within the LLMs, making the attack complexity high (H).

Hijacking Attack: In a hijacking attack, attackers use controlled instructions to take unauthorized control or exploit the behavior, output, or functionality of an LLM for malicious purposes [83, 96]. There are two types of hijacking attacks, vocabulary and square.

In a *vocabulary attack*, attackers manipulate LLMs by inserting delimiters or systematically rephrasing instructions until they achieve their goal such as revealing confidential information, generating specific false information, or exhibiting offensive behavior [94]. To execute this, attackers first identify words in the LLM's vocabulary that trigger the desired target behavior when included anywhere in the user prompt. These words are referred to as adversarial vocabulary [95]. This attack is typically the hardest to detect in user prompts using filters or other pattern-matching defenses, as many system prompts are designed to ensure a certain level of robustness in LLM applications, and some LLMs include automatic text filters for detection. However, the attack requires attackers to understand the specific model in-depth and to optimise arbitrary word sequences inserted into prompts to alter the model's behavior [94]. Therefore, the skill complexity required for this attack is classified as high (H).

The square attack is based on a randomised search strategy that involves selecting square-shaped localized updates at random positions of the input text [96]. This ensures that, in each iteration, the perturbation remains roughly on the boundary of the feasible set of the input text [97]. The square attack requires minimal expertise in LLMs, making its complexity low (L) to manipulate the input and produce the incorrect output.

Hallucination Attack: Hallucination attack is possible due to the nature of LLMs where attackers reveal

the vulnerability in LLMs during the inference phase and manipulate them to generate fabricated outputs when users are querying the model [93]. Poor benign prompt engineering or just badly functioning models cause these attacks, which results in *excessive agency* vulnerability [59]. Thus, attack vectors for such attacks are fabricated code libraries. There are different classifications for hallucination attacks, such as weak semantic attacks and out-of-distribution (OoD) attacks.

Weak semantic attacks are the attacks in which attackers alter a small number of tokens with semantic input and trick the model into generating false information [68]. The attackers uses a gradient-based token replacement approach for the replacement of few token and insert a perturbed prompt instead for maintaining the semantic of the input. This attacks need in-depth knowledge of the model and advances techniques for understanding the semantics of programming language used, so the attack complexity is high (H).

The *out-of-distribution (OoD) attacks* use random tokens (semantics are not preserved) that do not match with training data relevant to LLMs. As a result, LLMs fabricate non-sensible or unfaithful outputs [93]. The attacker needs to understand the distribution of training data, prompt engineering, decoding strategies and in-depth knowledge of the working of the model, therefore complexity of the attack is high (H).

3.2 Security Attacks on LLMs Infrastructure

The methodologies, various vectors, and impacts of security attacks on the LLMs infrastructure (Figure 4) are presented in this section. The analysis of the complexities of each identified attack is elaborated in Table 5.

Supply Chain Attack: The LLMs supply chain involves the whole lifecycle, from model training to ongoing maintenance. Supply chain attacks infiltrate various stages of the LLMs infrastructure, including data preparation, data pre-processing, model training, model deployment, model optimisation, etc. and exploit vulnerabilities in the components of each stage [59, 106]. The attacker may inject poisoned data into training, alter model during training or deployment, upload compromised models to public repositories, or manipulate third-party libraries or code, that support LLMs development. It may also involve the exploitation of insecure APIs or LLM plugin extensions that cloud providers use to host LLM infrastructure and target deprecated model dependencies or terms and conditions, and copyright material [107, 108]. LLM supply chain attack results into data breaches, output manipulation and denial of services as well. Depending on the type of supply-chain attack, the attack complexity could vary. From the perspective of infrastructure level attack, attacker needs in-depth understanding of LLMs and needs to have sophisticated technical skills, therefore, the overall attack complexity is high (H).

Inadequate Sandboxing: Sandbox attacks exploit the isolated environment (sandbox) where LLMs run to execute unintended commands, access unauthorised information, or manipulate the model's behaviour [109]. These attacks could be of two types: environmental segregation and system-level interaction.

Attackers target vulnerabilities within the sandbox environment and compromise the LLM's interaction with external components such as operating system, other containers or virtual machines, resulting into unauthorised access and information disclosure [116]. Such an attack is called *environmental segregation* attack. The attack complexity is high (H) as attacker needs high skills and technical knowledge for sandbox operations, isolation handling, and an exploit development environment [109].

System level interaction attack, due to inadequate sandboxing, exploit the direct interaction of LLMs with system-level processes, APIs, hardware components, and shell commands and result in privilege escalation, and unauthorised access, making the overall sophistication level of an attack high (H).

Access Control Attack is the exploitation of vulnerabilities in the access control policies or mechanisms that restrict and manage unauthorised access and export to the model's output, data, and parameters [59]. The attacker uses vectors such as API access, file or network information, arbitrary code execution on servers, social engineering tactics, or by-default configurations, resulting in privilege escalation, data breaches, misinformation, or harmful output. The attacker needs to know the API, network configurations, authentication, and authorisation mechanisms, or have expertise in prompt engineering. Overall, the access control attack complexity is high (H).

Ransomware Attack: Ransomware attackers target the model infrastructure or data to compromise LLMs operations. Attackers can encrypt, lock, or manipulate the LLM's functionality to disrupt its usability or extract sensitive information, which cannot be used until the ransom is paid [115]. The attacker can exploit training datasets, pre-trained models, explore CVEs to identify unpatched vulnerabilities, and third-party code to launch an attack. Therefore, attackers require advanced

Fig. 4 A Breakdown of Taxonomy of Attacks on LLMs Infrastructure

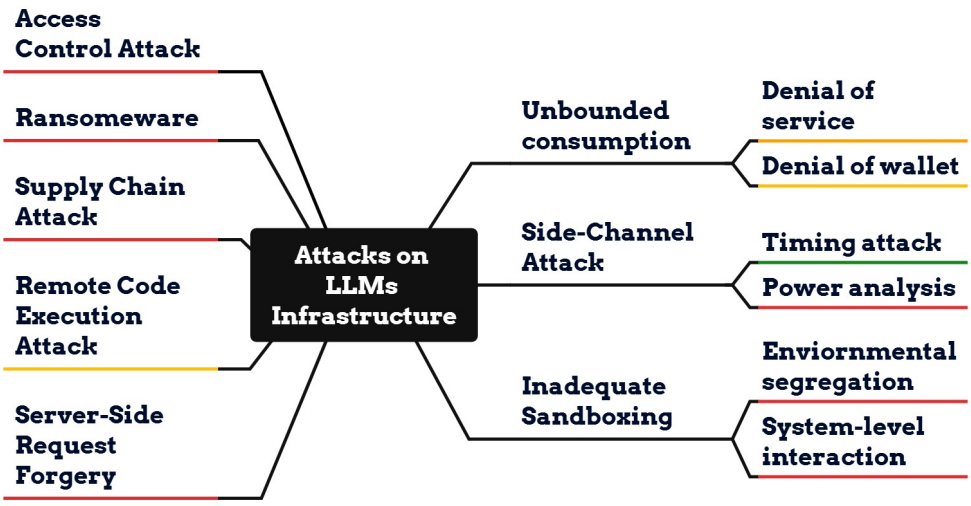


Table 5 Attacks on LLMs Infrastructure and Attack Complexities

Attacks	Attack Sub-types	Required Skills	Knowledge of Model/ Infrastructure	Ease of Exploitation	Attack Complexity
Unbounded Consumption [59]	Denial of service	M	M	M	M
	Denial of wallet	L	M	M	M
Inadequate Sandboxing [109]	Environmental segregation	H	H	H	H
	System-level interaction	H	H	H	H
Access Control [59]		H	H	M	H
Supply Chain Attack [59, 106]		H	H	H	H
Side-Channel Attack [59, 110, 111]	Timing attack	L	L	L	L
	Power analysis	M	H	H	H
Server-Side Request Forgery [112]		H	H	H	H
Remote Code Execution [59, 113]		M	M	M	M
Ransomware Attack [114, 115]		H	H	H	H

knowledge of the LLM architecture, APIs, hosting environment, and cryptographic methods and need high skills in AI-specific deployments [114, 115]. The attack complexity of a ransomware attack is high (H).

Unbounded Consumption Attack: An unbounded consumption attack is a malicious attempt in which an attacker makes the LLMs services unavailable for legitimate users or involved in the target's financial resources depletion. To launch unbounded consumption attack, the attacker exploits the LLM's ability to generate uncontrolled responses based on input queries, resulting in resource exhaustion, system failure, and economic losses [59]. Unbounded consumption attacks are categorised as denial-of-service (DoS) (resource-exhaustion attack) and denial of wallet (DoW) attack.

In *denial-of-service or resource-exhaustion attack*, attackers craft inputs to disrupt or degrade the services of LLMs for legitimate users. The attacker overwhelms LLMs with varying lengths of inputs, sends them the sheer volume of inputs that exceed the LLM's context window, or submits complex or resource-intensive queries to perform resource-heavy operations. Resource-exhaustion attacks result in resource depletion, increased latency, degraded performance, or even complete service unavailability, unresponsiveness, and potential failures. The attacker should have knowledge of prompt engineering or can use tools to send a large number of prompts, needs knowledge of the model, such as the token or context window limit, and infrastructure, so the attack complexity is medium (M).

In *denial of wallet (DoW) attack*, attackers exploit the financial model of cloud-based AI services by per-

forming a high volume of operations, impacting financial sustainability of the service provider. The attacker could use bots, have knowledge of prices of the services and have some understanding of model and cloud service provider security measures to bypass them. These factors make the overall attack complexity medium (M).

Side-Channel Attack: Side-channel attacks exploit the model parameters and architecture information, system's physical or logical operations information to infer sensitive user data [59]. Attackers utilise training data filtering, input preprocessing, and query filtering against language models to cause data leakage in models [117, 118].

The attackers could perform *side-channel timing attacks* where attacker could use tools to determine the response times of the queries [110]. The attacker needs the basic knowledge of timing variations and does not need to understand the internal architecture of the model or the system; only basic information about the API to measure response time is required. Timing attack has overall low (L) complexity.

A *power analysis side-channel attack* on LLMs is launched to acquire sensitive information about the data, parameters, and architecture of LLMs by exploiting variations in the power consumption of the LLM's hardware. The attack complexity is high (H) as attackers need in-depth knowledge of the underlying infrastructure, hardware configuration, and the identification of meaningful insights from noisy data and shared memory systems [111].

Server-side Request Forgery: In this attack, the attackers target vulnerabilities in the servers where the LLMs are deployed and exploit weaknesses in token transmission, API endpoints, or the network infrastructure to intercept, manipulate, or extract sensitive data. Therefore, attackers need high skills to monitor encrypted traffic and analyse token-length sequences to reconstruct LLMs' responses [112]. Overall, the attack complexity of this attack is high (H).

Remote Code Execution Attack: Remote Code Execution (RCE) attacks are infrastructure attacks where attackers exploit vulnerabilities in software systems to execute arbitrary code on the target machine. These attacks are due to improper output handling by LLMs [59]. The attacker provides prompts such as code, shell commands, or specific operations executable by the shell or interpreter, and the motive of an attack is to manipulate the LLMs to produce dangerous output. The attack needs prompt-engineering knowledge, knowledge of shell commands, and the knowledge of LLMs operations and interactions [113]. These attacks lead to unauthorised access, data disclosure, and system compromise [113]. Attackers use multiple steps to manipulate the model's behavior in this attack. Therefore, this is of a medium (M) complexity attack.

4 Security Assessment of eLLMs: Taxonomy Application and DREAD Analysis

We first introduced the DREAD threat model customised for eLLMs and then use it to map the proposed taxonomy to this domain.

4.1 DREAD Model Categories, Scores and Rationale

DREAD provides a structured and quantitative approach to assess and prioritise security threats based on a risk score, calculated using five criteria. These criteria are: (1) Damage (impact of an attack), (2) Reproducibility (ease of reproducing/replicating an attack), (3) Exploitability (the effort required to launch an attack), (4) Affected Users (number of (end) users affected by a threat being exploited), and (5) Discoverability (likelihood of a threat being exploited (discovered)). A threat receives a score of 0 to 10 for each category. The final rating of the threat is calculated based on the individual scores, and then the average score (overall risk score) is taken.

To enhance transparency and reproducibility of our DREAD assessment, we provide a clear and easy-to-use scoring scheme with accompanying rationales in the next subsection.

Damage: In the educational sector, the attacks on LLMs can damage or have an impact on student, researcher, employee (faculty members and professional staff) personal or financial information, academic integrity, or research data (intellectual property) or other institutional data, institution's infrastructure, including networks, applications, and devices, or involve reputational damage, institutional financial losses or operational discontinuity. For example, operational disruption is assigned a score of 10, as a complete or partial shutdown of an educational LLM system could halt teaching activities, exams, and administrative processes across one or more institutions, significantly affecting productivity and performance. In contrast, personal information disclosure is scored at 7.5, as it compromises confidentiality and privacy but does not necessarily interrupt core operations. Table 6 shows the damage scores and their respective rationales.

Table 6 Damage score and rationales

Score	Rationale
0	No Damage
2.5	Non-sensitive data exposure
5	Output biases/ unethical behavior/ faulty/erroneous/misleading output
7.5	Privilege escalation/ misprediction/ personal information disclosure/ data breach
10	Operational disruption/ financial losses/ disinformation (reputational damage)

Reproducibility; The reproducibility of LLM-based attacks can range from easy to circumstantial (Table 7). This categorisation is based on the number of steps or internal details of the model.

Table 7 Reproducibility score and rationales

Score	Meaning	Rationale
0	NA	
2.5	Circumstantial	Extremely difficult or impossible to reproduce the attack.
5	Very Complex	Use of multiple steps and in-depth internal knowledge about the model is required.
7.5	Complex	Using multiple steps, but does not require the internal knowledge of the model or the infrastructure.
10	Easy	Using a single step, or does not require the internal knowledge of the model or the infrastructure.

Exploitability: The exploitability of an attack could be determined from the skills or experience required by the attacker. For this category, the relevant scores are mentioned in Table 8 and rationale for these scores are as follow:

Table 8 Exploitability score and rationales

Score	Meaning	Rationale
0	NA	
2.5	Circumstantial	Extremely difficult or impossible to launch an attack.
5	Very Complex	Use of specialised or advance tactics, skills, methods or tools.
7.5	Complex	Attacker needs some skills or experience with some sophisticated techniques and available tools.
10	Easy	Attacker uses available tools, publicly available information or does not require any skills or expertise.

Affected Users: There are a number of internal or external stakeholders that could be impacted or affected by an attack on LLMs in an educational institution (see. Table 9). The affected users could be students, operational staff, administrative staff, academic staff, personnel from upper management, board members, etc. The attack could impact an individual such as, a single student or staff member, or a group of people, such as a research team (students, researchers, and staff) or an administrative team such from admissions.

This category is similar to the Damage category, however, here we only consider the number and type of users affected by the attack rather than overall damage. Generally, an attack is scored higher if more users are affected from the attack, and vice versa. We have classified attacks affecting Admin users or higher management as 7.5 since such attacks potentially affect a larger number of end-users. Where it is felt that the attack may be conducted in ways that affect different number of users, we choose the highest possible score.

Table 9 Affected User score and rationales

Score	Rationale
0	No User(s)
2.5	An individual user- student/ staff/researcher
5	Group of users
7.5	Administrative user(s) or higher management individual(s)
10	All stakeholders

Discoverability: The ease with which an attack is able to discover a vulnerability depends on the attack vectors utilised by the attacker. The level of the discoverability could be easy, complex, very complex and impossible as shown in the Table 10. The rationale for the each level is as follows:

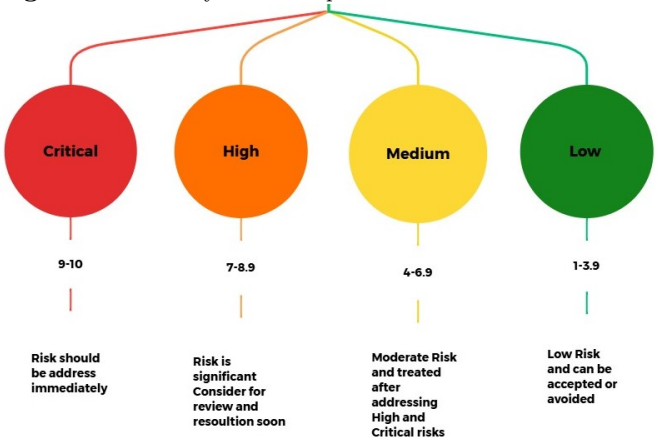
4.2 DREAD-based risk assessment of LLMs Attacks

We now show an application of the generalised LLM attack taxonomy with the DREAD model for assessing LLM-based risk in the educational sector. In order to do so, each DREAD category has been evaluated and scored for each attack. It must be noted that these scores are provided for generally known and accepted use cases of LLMs in various organisational aspects of an educational institution. It is likely that in a specific context and a specific organisation, the scores will vary. We reviewed the existing literature and found that, although cybersecurity issues in the education sector have been discussed, there are no concrete

Table 10 Discoverability score and rationales

Score	Meaning	Rationale
0	NA	
2.5	Circumstantial	Difficult or impossible to discover a vulnerability
5	Very Complex	Use of heuristic based prompting or in-depth knowledge of the model or infrastructure is required
7.5	Complex	Discovery based on the crafted instructions or or some knowledge of the model or infrastructure is required
10	Easy	Discovery based on simple input prompts or trial or error basis, or internal details of the model or infrastructure not required

Fig. 5 Risk severity levels adopted from NIST



case studies or statistics to support the assignment of DREAD scores. This underscores both the novelty of our study and opens new avenues for future research. To address this gap, we adopted a two-fold approach: first, we drew on the application of DREAD in other sectors [119–123]; second, we assessed the impact of identified attacks in educational contexts by examining their tactics, techniques, examples (from the primary studies), attack vectors and impacts (Table 4), and through discussions with security and education experts. By integrating both perspectives, we systematically assigned DREAD scores tailored to the educational environment. Table 11 shows the overall risk score and risk severity level for each type of the attack categorised in our proposed taxonomy (Section. 3) in the context of education sector. The overall risk scores are calculated using the average of the DREAD categories' scores (discussed in Section 4.1). The risk severity levels (Figure 5) have been adopted from the levels specified by NIST [124].

Based on the security risk levels, it has been observed from the Table 11 that four attacks are the most critical risks for the education sector. Thirty-two (32) attacks out of 50 attacks targeting LLMs are considered as the high risks within the education sector. None of the attack is at low level, and the remaining attacks are the medium risks. In the rest of this section, we discuss some examples to understand the assigned score.

4.2.1 Ransomware Attack: A high risk infrastructure attack in the education sector

Before discussing the DREAD scores for Ransomware attack, we discuss the salience of LLM usage within the education sector. Major LLM providers have already identified education as a priority market, offering services for both students and staff [125]. Various tools and software used by students and staff such as IDEs for software development, text editors, email and learning management systems, etc. are now being enhanced with LLM supported features. As a result, LLMs in educational contexts are rapidly becoming core to the learning experience, which directly affects student outcomes. Educators are incorporating LLM usage in coursework and assessment items which again makes them critical for student learning outcomes. Unavailability of the service due to an attack therefore has wide ranging consequences that affect student success. Beyond teaching and learning, LLMs are also being embedded into institutional operations such as student recruitment, engagement, and retention through automation of processes. Disruption in this area has the potential of having direct impact on the institution's financial performance.

The calculated DREAD risk score and level show that the ransomware attack (Section. 3.2) has high risk in the education sector (Table 11). The following description illustrates that a ransomware attack results in high damage and impacts a large number of stakeholders. However, as the attack is technically sophisticated compared to others, it is challenging to reproduce, exploit, and discover. Overall, this results in a high risk rating. If an attacker employs ransomware-as-a-service, the technical skills required are significantly lower, making the attack more easily reproducible. This aspect increases the severity of the ransomware threat. For instance, if we assigned reproducibility criteria a value of 10 due to the ease, the overall average risk score will be 9.5, making ransomware a critical threat for an organisation to focus on. However, in the following paragraphs, we have considered the scenario where attackers are executing the attack independently.

Table 11 Risk Assessments of the Attacks on LLMs in Education Sector using DREAD (Damage, Reproducibility, Exploitability, Affected Users, Discoverability) Model

	Attacks	Attack Sub-Types	D	R	E	A	D	Risk Score	Level
LLMs Model-Based Attacks	Multi-Modal	Text guided image generation	5	10	10	7.5	10	8.5	H
		Cross-modal	5	7.5	7.5	7.5	10	7.5	H
		Adversarial image	7.5	5	5	7.5	7.5	6.5	M
	Flowbreaking	Second thoughts	7.5	5	7.5	2.5	7.5	6	M
		Stop and roll	7.5	7.5	10	2.5	10	7.5	H
	Prompt Injection	Adversarial prompt	7.5	10	10	7.5	10	9	C
		Decoy and distract	5	7.5	10	5	7.5	7	H
		Indirect prompting	7.5	7.5	7.5	10	7.5	8	H
		Prompt divergence	7.5	5	5	5	5	5.5	M
		Direct injection	5	10	10	10	10	9	C
	Embedding	Embedding space	5	7.5	5	7.5	5	6	M
		Adversarial example	5	10	10	7.5	10	8.5	H
	Evasion	Obfuscation	5	10	10	7.5	10	8.5	H
		Token smuggling	7.5	10	10	7.5	10	9	C
	Jail break	Do anything now	5	7.5	7.5	10	10	8	H
		Multi-step jailbreak	10	10	7.5	10	7.5	9	C
		Adhoc jailbreak	7.5	10	7.5	7.5	7.5	8	H
		Token-level jailbreak	5	5	5	7.5	5	5.5	M
	Poisoning	Dataset poisoning	5	10	10	5	10	8	H
		Pre-training poisoning	5	10	10	5	5	7	H
		Fine-tuned poisoning	7.5	7.5	7.5	7.5	10	8	H
	Backdoor	Dataset backdoor	5	10	7.5	5	10	7.5	H
		Pre-trained backdoor	5	10	7.5	5	10	7.5	H
		Fine-tuned backdoor	5	5	10	5	10	7	H
		Composite	7.5	5	5	5	5	5.5	M
	Inference	Membership inference	7.5	7.5	10	7.5	10	8.5	H
		Attribute inference	2.5	10	10	10	10	8.5	H
		Input inference	7.5	5	5	10	5	6.5	M
	Manipulation	Token manipulation	5	5	5	7.5	10	6.5	M
		Content manipulation	10	5	5	10	5	7	H
	Extraction	Model gradient	5	5	5	10	5	6	M
		Training data extraction	7.5	10	10	7.5	7.5	8.5	H
		Model theft	10	5	5	10	5	7	H
	Leakage	Data leakage	7.5	7.5	7.5	10	10	8.5	H
		System prompt leakage	7.5	5	7.5	10	5	7	H
	Hallucination	Weak semantic	7.5	5	5	7.5	5	6	M
		Out-of-distribution	7.5	5	5	5	5	5.5	M
	Hijacking	Vocabulary	7.5	5	7.5	7.5	5	6.5	M
		Square	5	10	5	7.5	10	7.5	H
Infrastructure-Based Attacks	Unbounded Consumption	Denial of service	10	7.5	7.5	10	7.5	8.5	H
		Denial of wallet	10	7.5	7.5	10	7.5	8.5	H
	Inadequate Sandboxing	Environmental segregation	7.5	5	5	7.5	7.5	6.5	M
		System-level interaction	7.5	5	5	10	5	6.5	M
	Access Control		10	5	7.5	7.5	5	7	H
	Supply chain		10	5	5	10	5	7	H
	Side-Channel	Timing attacks	5	7.5	10	10	10	8.5	H
		Power analysis	7.5	7.5	7.5	10	7.5	8	H
	Server-Side Request Forgery		7.5	5	7.5	10	7.5	7.5	H
	Remote Code Execution		10	5	7.5	10	7.5	8	H
	Ransomware		10	5	5	10	5	7	H

Educational Institution Reputational Damage: A successful ransomware attack on eLLMs could significantly damage the educational institution's reputation among students, employees, parents, and wider internal and external stakeholders, resulting in the loss of public trust. This attack negatively impacts enrollments, academics, funding opportunities, partnerships, exchange programs, and campus operations.

Educational Operational Disruption: A successful attack on LLMs could disrupt learning activities or academic operations. For example, access to the learner-facing LLMs- educational interactive resources like vir-

tual tutors or agents, learning management systems (Moodle, Blackboard), virtual classrooms, and library chatbots become restricted due to the encryption, resulting in the cancellation of classes, exams, delayed feedback or results, and access to research materials and directly impact students' futures.

Similarly, the unavailability of educator-facing LLMs, which could support the teachers in providing writing analytics, smart content or assessment generation, personalized assessments, and automated grading and evaluation of exams, results in academic interruptions.

Regarding unresponsiveness of institutional support LLMs disrupt the fundamental academic operations such as student engagement, academic integrity, improvements in student retentions, teachers' evaluations, diagnosing strengths or gaps in student knowledge. Ransomware could target LLMs used in research by corrupting the datasets or publicly available information.

Data Breach of Affected Users: Many LLMs integrated into education infrastructure such as databases, servers hold critical and sensitive data, such as grades, personal information, financial information or even behavioral patterns. Ransomware attacks compromise the confidentiality and integrity of the data being encrypted. Similarly, these attacks result in intellectual property loss, either through theft or encryption, by targeting research papers, lesson plans, and private educational content.

Financial Losses: Educational institutions hit by ransomware may be forced to pay ransom to decrypt the data or resume access to the LLMs. Even if the ransom is not paid, institutions must incur the recovery cost for data restoration or system reconfiguration. Besides that, institutions have to pay penalties in case of data breaches.

4.2.2 Content Manipulation Attack: A high risk in the education sector

A content manipulation attack is the sub-type of manipulation attack discussed in Section. 3.1. In an educational environment, accurate and unbiased information is significant to maintain the trust of students, researchers and the academic staff. Therefore, this attack possesses a high risk in education.

Misinformation- Misleading, Incorrect or Biased Output and Disinformation: An attacker (disgruntled student or staff member, script kiddie (novice hacker to gain recognition, and hacktivist) could manipulate educational content like self-learning materials, course content, course planners, research papers, e-books, or even exam material, spreading incorrect or fake information to students and deteriorating the quality of education. For example, institutions use eLLMs as an intelligent tutoring system or virtual tutors to guide the students' online learning and to answer students' questions [126]; manipulated content could mislead students, causing confusion and substandard student performance. Similarly, an attacker could craft eLLM's input with biased data to produce biased outputs, so there could be a discrepancy in student's knowledge with the market requirements. Also, eLLMs used for generating research findings or insights could result in false or biased research outputs or poor research quality due to content manipulation impacting scientific or academic progress.

Also, within the context of administration, attackers could alter the LLM's use for examination purposes. For instance, attackers could change exam paper content or manipulate grading criteria, or exam planners, resulting in unfair examinations and loss of academic integrity. Similarly, eLLMs involved in student recruitment could impact the financial sustainability if the tuition fee pricing being varied by the attacker.

In addition, the educator relies on third-party LLM, which an attacker compromises to produce biased output, to create content according to the approved course outline. The biased output in the teaching content could mislead the students and indirectly impact teacher's creditability.

Exploitation/Risk of Overreliance on Technology: Educators, learners, and administrations depend on LLMs systems for educational content generation and decision-making without checking their facts (checking process in educational environments). Attackers can exploit this ignorance and launch content manipulation attacks to integrate incorrect data into the model, which could impact critical decisions or processes, result into internal risks to ensure the high-level quality educational services, and public embarrassment for the institution.

4.2.3 Token Smuggling Attack: A critical risk in the education sector

Token smuggling attacks can be launched on eLLMs for the creation of harmful or inappropriate learning materials or contents, gaining an access to block or restricted contents, evading the cheating and plagiarism. Table 11 shows that token smuggling attack has critical severity. The following are the reasons to emphasise that token smuggling is a critical risk for an educational institution, which should be mitigated proactively.

Harmful or Inappropriate Content Creation: Attackers could use carefully selected words and phrases to generate harmful content on cyberbullying, trolling

or even creating unsafe chemical experiments or violent scenarios on prohibited topics like improvising weapons, hacking networks, or accessing the dark web.

Evasion of Plagiarism or Cheating: Various LLMs tools have emerged rapidly and have been utilized in educational institutions. However, token smuggling attacks enable the students to craft the responses by the instructions encoded in a manner that increases the risk that the educator may be unable to distinguish whether a student's writing is their work, resulting in unfair assessments.

Access to Block or Restricted Content: Token smuggling can bypass these filters that eLLMs have configured to block or restrict content. For example, students used smuggled tokens to trick eLLMs into accessing exam questions or final grades or attendance records. Similarly, attacker could violate the intellectual property law and able to unblock the requests to websites to generate the content from copyright materials or retrieve the sensitive or confidential educational information.

4.3 Safeguards for Risk Mitigation in Education Sector

The previous sections identified and characterised LLMs-based attacks (on model directly or on infrastructure) along with the attack vectors involved and their impacts in general. To answer RQ2, we provide a risk assessment criteria to evaluate the severity of identified attacks in the education sector using DREAD model. In the literature, various technical mitigation strategies have been proposed for the identified risks [39, 59, 127]. However, this study focuses specifically on management-level safeguards. The following paragraphs outline actionable mitigation measures that educational organisations can adopt, depending on their resources and structure. To demonstrate their applicability, we consider two institutions: a 250-year-old established university and a 25-year-old technical university. Both may encounter risks such as token smuggling, adversarial prompts, or multi-step jailbreaks, each requiring tailored controls like authentication policies, rapid training programs for staff and students, threat modeling, and auditing. For instance, the established university may rely on a legacy student records system that is costly to upgrade, necessitating enhanced auditing and monitoring. In contrast, the younger technical university, using a flexible cloud-based solution, should prioritise staff training and awareness, along with protocols for regular software updates and backups.

Within an educational environment, establishing the safeguards to mitigate and address the risks raised due to LLMs-based attacks is paramount to ensure that LLMs continues to maintain the integrity of educational experiences. Following are some strategies that should be adopted by educational institutions for risk mitigation.

4.3.1 Develop and Enforce eLLMs-Usage Policy

There is a lack of comprehensive policies and guidelines about AI usage, including LLMs in education [17]. Therefore, educational institutions should establish clear guidelines for using eLLMs ethically and sensibly, and enforce accountability. Auditing and monitoring are significant for analysing log interactions and detecting unusual insights such as an indication of smuggling attacks, content manipulations while prompting. The sensitive queries should require authentication and a strict code of conduct for the user. Also, the users should report harmful, unsensible content generation by eLLMs. The policies should be transparent and fair, consider all the concerned educational departments, and involve collaboration between educators, policymakers, and student representatives.

4.3.2 Conduct Threat Modeling and Risk Assessment

Due to the complexity and novelty of LLMs, threat modeling is an ongoing and structured approach to identify, assess, and prioritise the threats to them [128]. In the context of the educational system, to minimise the harm to the learners, educators, and institutions themselves, eLLMs models and infrastructure should be assessed regularly to identify the attack surfaces, vectors, impacts, attacker's motives, and model performances. Similarly, risk assessment could help educational institutions to quantify and prioritise the risks (based on their potential impact and likelihood) associated with eLLMs , enabling them to determine the appropriate risk strategies (mitigation, transfer, avoidance, or acceptance). Threat modeling and risk assessment are essential for educational institutions to allocate their resources and efforts effectively and efficiently, to create a strong institutional culture and awareness, to reduce uncertainty for the students and educators, and to help prevent future incidents.

4.3.3 Conduct Rapid Training and Awareness

Rapid training and awareness is the most important strategy where the educator and administrative staff should get support and awareness regarding the misuse of eLLMs. Due to digital education and students' diverse learning needs and interests, a one-size-fits-all

approach will not be sufficient. Therefore, depending on the personalised reliance on the eLLMs, effective integration of these tools required educators and administrative staff to understand and identify unusual prompts, misinform and disinform scenarios or contents, and set strict usage policies to minimise cheating and plagiarism issues. Students should also be made aware about the use of eLLMs responsibly and the potential risks associated with LLMs-based attacks.

4.3.4 Implement Regular Security Updates, Apply Patching and Maintain Response Plans

The educational LLMs models and their underlying infrastructure should be continuously updated to address newly discovered vulnerabilities by software providing runtime guardrails. The cyber-security analyst personnel in information technology services (ITS) department should continuously monitor and detect misuse of eLLMs and their performances in educational environment and develop an incident response plan to follow if some damage has been occurred due to attacks on these eLLMs.

4.3.5 Implement Strong Access Controls

eLLMs should be secured using multi-factor authentication (MFA) and role-based access control (RBAC). Also, educational institutions relying on LLMs should implement secure key management protocols, least privilege principles and defense-in-depth measures.

5 Conclusion and Future Directions

LLMs have emerged as important tools in various industries, including education, and are used to perform various language-based tasks. However, these models are susceptible to security attacks that can impact the model, the infrastructure, and the organisation. This paper investigated and introduced a general taxonomy to categorise sophisticated attacks to LLMs based on their attack complexity, which will be useful for academic and industrial practitioners to secure the LLMs against malicious actors. Notably, our proposed generalised taxonomy could also be applied to other sectors such as health-care, finance and industrial automation. To show its applicability, we have applied the proposed taxonomy in education. We also assess the severity of these attacks in the education sector using the DREAD risk assessment model and suggest a few risk mitigation strategies to prevent the identified attacks.

In the future, it will be valuable to simulate attack scenarios on existing LLMs-integrated educational tools such as Moodle, to identify vulnerabilities, assess their severity, and recommend effective controls, such as enforcing an educational LLM usage policy. As future work, we also plan to empirically validate and refine the scores using publicly available risk reports from educational institutions or through questionnaire-based surveys with relevant stakeholders. Additionally, other risk assessment frameworks could be incorporated in future research to enhance the comprehensiveness and robustness of the investigation. We also intend to propose a detailed technical mitigation framework that maps defensive strategies to each identified risk and attack type within the context of educational institutions. To improve the practical utility of this framework, we will provide its technical implementation for the most critical risks, offering actionable guidance for security practitioners and system developers.

5.1 Declarations

Ethical Approval This article does not report any research with human participants or animals.

Informed consent This article does not report any studies with human participants.

Financial interests The authors do not have any financial interests in this research.

Conflict of Interest All authors declare that there is no conflict of interest.

References

1. Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.
2. Mohaimenul Azam Khan Raiaan, Md Saddam Hossain Mukta, Kaniz Fatema, Nur Mohammad Fahad, Sadman Sakib, Most Marufatul Jannat Mim, Jubaer Ahmad, Mohammed Eunus Ali, and Sami Azam. A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access*, 2024.
3. Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. Prompt injection attack against llm-integrated applications. *arXiv preprint arXiv:2306.05499*, 2023.

4. Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*, 2023.
5. Tianhao Shen, Renren Jin, Yufei Huang, Chuang Liu, Weilong Dong, Zishan Guo, Xinwei Wu, Yan Liu, and Deyi Xiong. Large language model alignment: A survey. *arXiv preprint arXiv:2309.15025*, 2023.
6. Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*, 2023.
7. Sanjay Kukreja, Tarun Kumar, Amit Purohit, Abhijit Dasgupta, and Debashis Guha. A literature survey on open source large language models. In *Proceedings of the 2024 7th International Conference on Computers in Management and Business*, pages 133–143, 2024.
8. A Chowdhery et al. Scaling language modeling with pathways, 2022.
9. Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
10. Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.
11. Technology Innovation Institute (TII). Falcon LLM: Open-Source Large Language Models, 2023. Accessed: 20-November-2024.
12. mhopkins-msft DOMARS and aviviano. Models.
13. mhopkins-msft DOMARS and aviviano. Gpt-4.
14. DeepSeek. Deepseek - into the unknown, 2025. Retrieved on: 2025-02-08.
15. Matt Bower, Jodie Torrington, Jennifer WM Lai, Peter Petocz, and Mark Alfano. How should we change teaching and assessment in response to increasingly powerful generative artificial intelligence? outcomes of the chatgpt teacher survey. *Education and Information Technologies*, pages 1–37, 2024.
16. Saurabh Pahune and Manoj Chandrasekharan. Several categories of large language models (llms): A short survey. *arXiv preprint arXiv:2307.10188*, 2023.
17. Firuz Kamalov, David Santandreu Calonge, and Ikhlaas Gurrib. New era of artificial intelligence in education: Towards a sustainable multifaceted revolution. *Sustainability*, 15(16), 2023.
18. Shafi Parvez Mohammed and Gahangir Hossain. Chatgpt in education, healthcare, and cybersecurity: Opportunities and challenges. In *2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0316–0321. IEEE, 2024.
19. Hanyi Xu, Wensheng Gan, Zhenlian Qi, Jiayang Wu, and Philip S Yu. Large language models for education: A survey. *arXiv preprint arXiv:2405.13001*, 2024.
20. Farzad Nourmohammadzadeh Motlagh, Mehrdad Hajizadeh, Mehryar Majd, Pejman Najafi, Feng Cheng, and Christoph Meinel. Large language models in cybersecurity: State-of-the-art. *arXiv preprint arXiv:2402.00891*, 2024.
21. Aram Bahrini, Mohammadsadra Khamoshifar, Hossein Abbasimehr, Robert J Riggs, Maryam Esmaeili, Rastin Mastali Majdabadkohne, and Morteza Pasehvar. Chatgpt: Applications, opportunities, and threats. In *2023 Systems and Information Engineering Design Symposium (SIEDS)*, pages 274–279. IEEE, 2023.
22. Hany F Atlam. LLMs in Cyber Security: Bridging Practice and Education. *Big Data and Cognitive Computing*, 9(7):184, 2025.
23. Matthew NO Sadiku, Uwakwe C Chukwu, and Janet O Sadiku. Cybersecurity for education. *European Journal of Innovation in Nonformal Education*, 3(6):182–188, 2023.
24. Nokuthaba Siphambili. Exploring cybersecurity implications in higher education. In *European Conference on Cyber Warfare and Security*, volume 23, pages 526–531, 2024.
25. Jelen Sara. Education sector common breaches and cyber threats. Offsec, 2024.
26. Asimily. 4 cyberattacks that shook universities and colleges in the last year. Asimily Blog, 2024.
27. Barbara Kitchenham, O Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. Systematic literature reviews in software engineering – a systematic literature review. *Information and Software Technology*, 51(1):7–15, 2009.
28. Microsoft Learn Challenge. Threat modeling for drivers. Accessed: 2024-09-18.
29. Nitin Naik, Paul Jenkins, Paul Grace, Dishita Naik, Shaligram Prajapat, and Jingping Song. A comparative analysis of threat modelling methods:

Stride, dread, vast, pasta, octave, and linddun. *Authorea Preprints*, 2024.
30. Craig Smith. *The car hacker's handbook: a guide for the penetration tester*. no starch press, 2016.
31. Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, 4(2):100211, 2024.
32. Erfan Shayegani, Md Abdullah Al Mamun, Yu Fu, Pedram Zaree, Yue Dong, and Nael Abu-Ghazaleh. Survey of vulnerabilities in large language models revealed by adversarial attacks. *arXiv preprint arXiv:2310.10844*, 2023.
33. Umar Iqbal, Tadayoshi Kohno, and Franziska Roesner. Llm platform security: Applying a systematic evaluation framework to openai's chatgpt plugins. *arXiv preprint arXiv:2309.10254*, 2023.
34. Haomiao Yang, Kunlan Xiang, Mengyu Ge, Hongwei Li, Rongxing Lu, and Shui Yu. A comprehensive overview of backdoor attacks in large language models within communication networks. *IEEE Network*, 2024.
35. Xiaodong Wu, Ran Duan, and Jianbing Ni. Unveiling security, privacy, and ethical concerns of chatgpt. *Journal of Information and Intelligence*, 2023.
36. Rahul Pankajakshan, Sumitra Biswal, Yuvaraj Govindarajulu, and Gilad Gressel. Mapping llm security landscapes: A comprehensive stakeholder risk assessment proposal. *arXiv preprint arXiv:2403.13309*, 2024.
37. Erik Derner and Kristina Batistič. Beyond the safeguards: Exploring the security risks of chatgpt. *arXiv preprint arXiv:2305.08005*, 2023.
38. Glorin Sebastian. Do chatgpt and other ai chatbots pose a cybersecurity risk?: An exploratory study. *International Journal of Security and Privacy in Pervasive Computing (IJSPPC)*, 15(1):1–11, 2023.
39. Tianyu Cui, Yanling Wang, Chuanpu Fu, Yong Xiao, Sijia Li, Xinhao Deng, Yunpeng Liu, Qinglin Zhang, Ziyi Qiu, Peiyang Li, et al. Risk taxonomy, mitigation, and assessment benchmarks of large language model systems. *arXiv preprint arXiv:2401.05778*, 2024.
40. Junjie Chu, Yugeng Liu, Ziqing Yang, Xinyue Shen, Michael Backes, and Yang Zhang. Comprehensive assessment of jailbreak attacks against llms. *arXiv preprint arXiv:2402.05668*, 2024.
41. Arijit Ghosh Chowdhury, Md Mofijul Islam, Vaibhav Kumar, Faysal Hossain Shezan, Vinija Jain, and Aman Chadha. Breaking down the defenses: A comparative survey of attacks on large language models. *arXiv preprint arXiv:2403.04786*, 2024.
42. Jie Zhang, Haoyu Bu, Hui Wen, Yongji Liu, Haiqiang Fei, Rongrong Xi, Lun Li, Yun Yang, Hongsong Zhu, and Dan Meng. When llms meet cybersecurity: A systematic literature review. *Cybersecurity*, 8(1):55, 2025.
43. Liz Kellermeyer, Ben Harnke, and Shandra Knight. Covidence. *Journal of the Medical Library Association*, 106(4), 2018.
44. Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: an update. *Information and Software Technology*, 64:1–18, 2015.
45. MITRE Corporation. Mitre Atlas . `https://atlas.mitre.org/`.
46. AI Incident Database . https://incidentdatabase.ai/.
47. Sandy Dunn. Llm ai cybersecurity and governance checklist.
48. Muhammad Uzair Khan, Salman Sherin, Muhammad Zohaib Iqbal, and Rubab Zahid. Landscaping systematic mapping studies in software engineering: a tertiary study. *Journal of Systems and Software*, 149:396–436, 2019.
49. Narayanaswamy Gopi. Llm security - threats faced by large language models (llms), January 2024. LinkedIn Post.
50. Xiaoyu Zhang, Cen Zhang, Tianlin Li, Yihao Huang, Xiaojun Jia, Ming Hu, Jie Zhang, Yang Liu, Shiqing Ma, and Chao Shen. Jailguard: A universal detection framework for prompt-based attacks on LLM systems. *ACM Transactions on Software Engineering and Methodology*, 2025.
51. Ziyi Kou, Shichao Pei, Yijun Tian, and Xiangliang Zhang. Character as pixels: A controllable prompt adversarial attacking framework for black-box text guided image generation models. In *IJCAI*, pages 983–990, 2023.
52. Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*, 2023.
53. Maciej Żelaszczyk and Jacek Mańdziuk. Text-to-image cross-modal generation: A systematic review. *arXiv preprint arXiv:2401.11631*, 2024.
54. Chenyu Zhang, Mingwang Hu, Wenhui Li, and Lanjun Wang. Adversarial attacks and defenses on text-to-image diffusion models: A survey. *Information Fusion*, page 102701, 2024.

55. Chen Henry Wu, Jing Yu Koh, Ruslan Salakhutdinov, Daniel Fried, and Aditi Raghunathan. Adversarial attacks on multimodal agents. *arXiv preprint arXiv:2406.12814*, 2024.
56. Evron Gadi. Suicide bot: New ai attack causes llm to provide potential "self-harm" instructions. https://www.knostic.ai/blog/flowbreaking-ai-attack, 2024. Accessed: 2025-1-1.
57. Simon Willison. Llm flowbreaking. https://simonwillison.net/2024/Nov/29/llm-flowbreaking, November 2024. Accessed: 2025-1-01.
58. Geslevich Nizan. Generative ai under attack: Flowbreaking exploits trigger data leaks. https://www.forbes.com/sites/nizangpackin/2024/11/26/generative-ai-under-attack-flowbreaking-exploits-trigger-data-leaks/, 2024. Accessed:2025-1-1.
59. Owasp top 10 for llm applications 2025.
60. Benji Peng, Keyu Chen, Ming Li, Pohsun Feng, Ziqian Bi, Junyu Liu, and Qian Niu. Securing large language models: Addressing bias, misinformation, and prompt attacks. *arXiv preprint arXiv:2409.08087*, 2024.
61. Srikar Alla and Ali Shiri Sichani. Cyberattacks on large language models-attack detection and architecture adaptability. In *SoutheastCon 2025*, pages 143–148. IEEE, 2025.
62. Sander Schulhoff, Jeremy Pinto, Anaum Khan, Louis-François Bouchard, Chenglei Si, Svetlina Anati, Valen Tagliabue, Anson Kost, Christopher Carnahan, and Jordan Boyd-Graber. Ignore this title and hackaprompt: Exposing systemic vulnerabilities of llms through a global prompt hacking competition. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4945–4977, 2023.
63. Maanak Gupta, CharanKumar Akiri, Kshitiz Aryal, Eli Parker, and Lopamudra Praharaj. From chatgpt to threatgpt: Impact of generative ai in cybersecurity and privacy. *IEEE Access*, 2023.
64. Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, pages 79–90, 2023.
65. Qiusi Zhan, Richard Fang, Henil Shalin Panchal, and Daniel Kang. Adaptive attacks break defenses against indirect prompt injection attacks on LLM agents. *arXiv preprint arXiv:2503.00061*, 2025.
66. Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*, 2023.
67. Surender Suresh Kumar, ML Cummings, and Alexander Stimpson. Strengthening llm trust boundaries: a survey of prompt injection attacks. In *2024 IEEE 4th International Conference on Human-Machine Systems (ICHMS)*, pages 1–6, 2024.
68. Xiaodong Wu, Ran Duan, and Jianbing Ni. Unveiling security, privacy, and ethical concerns of chatgpt. *Journal of Information and Intelligence*, 2(2):102–115, 2024.
69. Leo Schwinn, David Dobre, Stephan Günnemann, and Gauthier Gidel. Adversarial attacks and defenses in large language models: Old and new threats. *arXiv preprint arXiv:2310.19737*, 2023.
70. Leo Schwinn, David Dobre, Sophie Xhonneux, Gauthier Gidel, and Stephan Gunnemann. Soft prompt threats: Attacking safety alignment and unlearning in open-source llms through the embedding space. *arXiv preprint arXiv:2402.09063*, 2024.
71. Wentao Wang, Han Xu, Yuxuan Wan, Jie Ren, and Jiliang Tang. Towards adversarial learning: from evasion attacks to poisoning attacks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4830–4831, 2022.
72. Pranjal Kumar. Adversarial attacks and defenses for large language models (llms): methods, frameworks & challenges. *International Journal of Multimedia Information Retrieval*, 13(3):26, 2024.
73. Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 346–363. IEEE, 2023.
74. Bo Hui, Haolin Yuan, Neil Gong, Philippe Burlina, and Yinzhi Cao. Pleak: Prompt leaking attacks against large language model applications. *arXiv preprint arXiv:2405.06823*, 2024.
75. Yifan Jiang, Kriti Aggarwal, Tanmay Laud, Kashif Munir, Jay Pujara, and Subhabrata Mukherjee. Red queen: Safeguarding large language models against concealed multi-turn jailbreaking. *arXiv preprint arXiv:2409.17458*, 2024.

76. Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. Catastrophic jailbreak of open-source llms via exploiting generation. *arXiv preprint arXiv:2310.06987*, 2023.
77. Kevin Eykholt, Farhan Ahmed, Pratik Vaishnavi, and Amir Rahmati. Taking off the rose-tinted glasses: A critical look at adversarial ml through the lens of evasion attacks. *arXiv preprint arXiv:2410.12076*, 2024.
78. Filippo Galli, Luca Melis, and Tommaso Cucinotta. Noisy neighbors: Efficient membership inference attacks against llms. *arXiv preprint arXiv:2406.16565*, 2024.
79. Hai Huang, Zhengyu Zhao, Michael Backes, Yun Shen, and Yang Zhang. Composite backdoor attacks against large language models. *arXiv preprint arXiv:2310.07676*, 2023.
80. Shuai Zhao, Jinming Wen, Luu Anh Tuan, Junbo Zhao, and Jie Fu. Prompt as triggers for backdoor attack: Examining the vulnerability in language models. *arXiv preprint arXiv:2305.01219*, 2023.
81. Antispoofing. Data poisining attacks and llms chatbots: How experts are responding. Accessed: 2024-12-18.
82. Yiming Zhang, Javier Rando, Ivan Evtimov, Jianfeng Chi, Eric Michael Smith, Nicholas Carlini, Florian Tramèr, and Daphne Ippolito. Persistent pre-training poisoning of llms. *arXiv preprint arXiv:2410.13722*, 2024.
83. Hongwei Yao, Jian Lou, and Zhan Qin. Poisonprompt: Backdoor attack on prompt-based large language models. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7745–7749. IEEE, 2024.
84. Bargav Jayaraman and David Evans. Are attribute inference attacks just imputation? In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 1569–1582, 2022.
85. Benjamin Zi Hao Zhao, Aviral Agrawal, Catisha Coburn, Hassan Jameel Asghar, Raghav Bhaskar, Mohamed Ali Kaafar, Darren Webb, and Peter Dickinson. On the (in) feasibility of attribute inference attacks on machine learning models. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 232–251. IEEE, 2021.
86. Andrii Balashov, Olena Ponomarova, and Xiaohua Zhai. Multi-stage prompt inference attacks on enterprise llm systems. *arXiv preprint arXiv:2507.15613*, 2025.
87. Matthew Gereti, Alejandro Robinson, Sebastian Williams, Christopher Anderson, and Dominic Walker. Token-based prompt manipulation for automated large language model evaluation. *Authorea Preprints*, 2024.
88. Stefano Cresci, Marinella Petrocchi, Angelo Spognardi, and Stefano Tognazzi. Adversarial machine learning for protecting against online manipulation. *IEEE Internet Computing*, 26(2):47–52, 2021.
89. Lewis Birch, William Hackett, Stefan Trawicki, Neeraj Suri, and Peter Garraghan. Model leeching: An extraction attack targeting llms. *arXiv preprint arXiv:2309.10544*, 2023.
90. Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.
91. João Vitorino, Eva Maia, and Isabel Praça. Adversarial evasion attack efficiency against large language models. *arXiv preprint arXiv:2406.08050*, 2024.
92. Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K Jain. Adversarial attacks and defenses in images, graphs and text: A review. *International journal of automation and computing*, 17:151–178, 2020.
93. Jia-Yu Yao, Kun-Peng Ning, Zhen-Hui Liu, Mu-Nan Ning, Yu-Yang Liu, and Li Yuan. Llm lies: Hallucinations are not bugs, but features as adversarial examples. *arXiv preprint arXiv:2310.01469*, 2023.
94. Patrick Levi and Christoph P Neumann. Vocabulary attack to hijack large language model applications. *arXiv preprint arXiv:2404.02637*, 2024.
95. Yao Qiang. Hijacking large language models via adversarial in-context learning. Master's thesis, Wayne State University, 2024.
96. Yao Qiang, Xiangyu Zhou, and Dongxiao Zhu. Hijacking large language models via adversarial in-context learning. *arXiv preprint arXiv:2311.09948*, 2023.
97. Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European conference on computer vision*, pages 484–501. Springer, 2020.
98. Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arXiv:2310.04451*, 2023.

99. Haoran Li, Dadi Guo, Wei Fan, Mingshi Xu, Jie Huang, Fanpu Meng, and Yangqiu Song. Multi-step jailbreaking privacy attacks on chatgpt. *arXiv preprint arXiv:2304.05197*, 2023.
100. Yuqi Zhou, Lin Lu, Hanchi Sun, Pan Zhou, and Lichao Sun. Virtual context: Enhancing jailbreak attacks with special token injection. *arXiv preprint arXiv:2406.19845*, 2024.
101. Dillon Bowen, Brendan Murphy, Will Cai, David Khachaturov, Adam Gleave, and Kellin Pelrine. Scaling laws for data poisoning in llms. *arXiv preprint arXiv:2408.02946*, 2024.
102. Takayuki Miura, Toshiki Shibahara, and Naoto Yanai. Megex: Data-free model extraction attack against gradient-based explainable ai. In *Proceedings of the 2nd ACM Workshop on Secure and Trustworthy Deep Learning Systems*, pages 56–66, 2024.
103. Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 54(11s):1–37, 2022.
104. Parisa Kaghazgaran, Majid Alfifi, and James Caverlee. Wide-ranging review manipulation attacks: Model, empirical study, and countermeasures. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 981–990, 2019.
105. Cong Liao, Haoti Zhong, Sencun Zhu, and Anna Squicciarini. Server-based manipulation attacks against machine learning models. In *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, pages 24–34, 2018.
106. Tanmay Singla, Dharun Anandayuvaraj, Kelechi G Kalu, Taylor R Schorlemmer, and James C Davis. An empirical study on using large language models to analyze software supply chain security failures. In *Proceedings of the 2023 Workshop on Software Supply Chain Offensive Research and Ecosystem Defenses*, pages 5–15, 2023.
107. Qiang Hu, Xiaofei Xie, Sen Chen, and Lei Ma. Large language model supply chain: Open problems from the security perspective. *arXiv preprint arXiv:2411.01604*, 2024.
108. Obasdiaru Andrew. Llm supply chain attack: Prevention strategies.
109. Sara Abdali, Richard Anarfi, CJ Barberan, and Jia He. Securing large language models: Threats, vulnerabilities and responsible practices. *arXiv preprint arXiv:2403.12503*, 2024.
110. Xinyao Zheng, Husheng Han, Shangyi Shi, Qiyan Fang, Zidong Du, Qi Guo, and Xing Hu. Inputsnatch: Stealing input in llm services via timing side-channel attacks. *arXiv preprint arXiv:2411.18191*, 2024.
111. Najmeh Nazari, Furi Xiang, Chongzhou Fang, Hosein Mohammadi Makrani, Aditya Puri, Kartik Patwari, Hossein Sayadi, Setareh Rafatirad, Chen-Nee Chuah, and Houman Homayoun. Llm-fin: Large language models fingerprinting attack on edge devices. In *2024 25th International Symposium on Quality Electronic Design (ISQED)*, pages 1–6. IEEE, 2024.
112. Roy Weiss, Daniel Ayzenshteyn, Guy Amit, and Yisroel Mirsky. What was your prompt? a remote keylogging attack on ai assistants. *arXiv preprint arXiv:2403.09751*, 2024.
113. Tong Liu, Zizhuang Deng, Guozhu Meng, Yuekang Li, and Kai Chen. Demystifying rce vulnerabilities in llm-integrated apps. *arXiv preprint arXiv:2309.02926*, 2023.
114. Lance Itonin, Nathaniel Caldwell, and Ava Richardson. Leveraging large language models for autonomous red teaming in simulating advanced ransomware attacks. *Affiliation not available*, September 2024. Preliminary report on TechHub.
115. Jie Zhang, Haoyu Bu, Hui Wen, Yu Chen, Lun Li, and Hongsong Zhu. When llms meet cybersecurity: A systematic literature review. *arXiv preprint arXiv:2405.03644*, 2024.
116. Fangzhou Wu, Ning Zhang, Somesh Jha, Patrick McDaniel, and Chaowei Xiao. A new era in llm security: Exploring security concerns in real-world llm-based systems. *arXiv preprint arXiv:2402.18649*, 2024.
117. Edoardo Debenedetti, Giorgio Severi, Nicholas Carlini, Christopher A Choquette-Choo, Matthew Jagielski, Milad Nasr, Eric Wallace, and Florian Tramèr. Privacy side channels in machine learning systems. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 48–61, 2024.
118. Praveen Kulkarni, Vincent Verneuil, Stjepan Picek, and Lejla Batina. Order vs. chaos: a language model approach for side-channel attacks. *Cryptology ePrint Archive*, 2023.
119. K Ram Mohan Rao and Durgesh Pant. A threat risk modeling framework for geospatial weather information system (gwis) a dread based study. *international Journal of Advanced Computer Science and Applications*, 1(3), 2010.
120. Lu Zhang, Arie Taal, Reginald Cushing, Cees de Laat, and Paola Grosso. A risk-level assess-

ment system based on the stride/dread model for digital data marketplaces. *International Journal of Information Security*, 21(3):509–525, 2022.
121. Archana Singhal, Hema Banati, et al. Fuzzy logic approach for threat prioritization in agile security framework using dread model. *arXiv preprint arXiv:1312.6836*, 2013.
122. Buhang Zhai, Oluwatobi Noah Akande, Saurabh Agarwal, and Wooguil Pak. Security risk assessment of internet of things health devices using dread and stride models. *Ain Shams Engineering Journal*, 16(11):103721, 2025.
123. P Subhash, MOHAMMED Qayyum, K Mehernadh, K Jeevan Sahit, C Likhitha Varsha, and M Nevan Hardeep. Risk assessment threat modelling using an integrated framework to enhance security. *J. Theor. Appl. Inf. Technol*, 102(9):3857–3867, 2024.
124. National vulnerability database (nvd), 2024. Accessed: 2024-12-10.
125. OpenAI. Introducing study mode. `https://openai.com/index/chatgpt-study-mode/`, July 2025. Accessed: 2025-10-08.
126. Karan Taneja, Pratyusha Maiti, Sandeep Kakar, Pranav Guruprasad, Sanjeev Rao, and Ashok K Goel. Jill watson: A virtual teaching assistant powered by chatgpt. In *International Conference on Artificial Intelligence in Education*, pages 324–337. Springer, 2024.
127. Lionel Nganyewou Tidjon and Foutse Khomh. Threat assessment in machine learning based systems. *arXiv e-prints*, pages arXiv–2207, 2022.
128. Stephen Burabari Tete. Threat modelling and risk analysis for large language model (llm)-powered applications. *arXiv e-prints*, page 2406, 2024.