SCIENCE CHINA
Information Sciences

• REVIEW •

September 2025, Vol. 68, Iss. 9, 191101:1–191101:34
https://doi.org/10.1007/s11432-024-4351-3

Backdoor threats in large language models—a survey

Shuai LIU[1], Yiheng PAN[1], Kun HONG[1], Ruite FEI[1], Chenhao LIN[2*], Qian LI[2] & Chao SHEN[2]

[1]*School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China*
[2]*School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an 710049, China*

Received 15 August 2024/Revised 11 November 2024/Accepted 18 March 2025/Published online 14 August 2025

Abstract Large language models (LLMs), with their advanced language comprehension and text generation capabilities, have demonstrated remarkable performance across diverse application scenarios involving code processing, search engines, and translation, among others. However, these models have become increasingly vulnerable to security threats, particularly to backdoor attacks. Therefore, a timely and comprehensive review of the existing backdoor threats is urgently required. In this paper, we present a systematic and timely review of the research on backdoor attacks on LLMs, categorising existing attack and defence methods according to the LLM. Additionally, we draw comparisons with backdoor attacks in traditional deep learning to provide a more intuitive understanding of backdoor threats in LLMs. Through this effective analysis and an evaluation of the reviewed studies, we identify the current research challenges and propose potential future research directions to address these issues.

Keywords large language models, backdoor threats, attack and defence category, LLM life cycle, artificial intelligence security

Citation Liu S, Pan Y H, Hong K, et al. Backdoor threats in large language models—a survey. Sci China Inf Sci, 2025, 68(9): 191101, https://doi.org/10.1007/s11432-024-4351-3

1 Introduction

Artificial intelligence (AI), which has significantly impacted human society, has undergone extensive development and accumulation over the decades. The 21st century has witnessed a transformative leap in AI, driven by improved computing performance and the advent of the big data era. Breakthroughs in deep learning and other technologies have led to significant advancements in fields, such as image recognition, speech recognition, and natural language processing.

As AI technology continues to evolve rapidly, particularly with the rise of large language models (LLMs), science and industry have been transformed. Notably, chat generative pre-trained transformer (ChatGPT) of OpenAI, with its powerful contextual semantic analysis and text generation capabilities, has established a new standard [1]. Following this trend, other companies have introduced their own LLM-based products, such as Google's Gemini [2], GitHub's Copilot [3], and Microsoft's Bing Chat [4]. LLMs endow machines with enhanced reasoning and response capabilities, which have substantially improved human life. These advancements include notable applications, including voice assistants [5], smart home products [6], and sophisticated AI agents across various domains, such as education [7] and healthcare [8].

However, the rapid development of LLMs has also introduced significant security concerns, notably the threat of backdoor attacks [9]. The malicious functionalities embedded into a model by backdoor attacks are activated by specific triggers predefined by the attacker. Normally, the model performs as expected; however, upon encountering the trigger, it exhibits malicious behaviour. These attacks pose severe risks as they are difficult to detect and remove, potentially leading to harmful outcomes in AI applications.

Given the growing interest in backdoor attacks targeting LLMs, several existing surveys [10–13] shown in Table 1 [10–19] have aimed to analyse and categorise the current work in this field. Most of these surveys have the following problems: insufficient summary of the correlation between attacks and defences, lack of more detailed categories, and insufficient comparison with traditional deep-learning backdoor attacks.

* Corresponding author (email: linchenhao@xjtu.edu.cn)

Table 1 Survey of backdoor attacks and defences in LLMs.

Authors	Year	Threat scenarios	Datasets	Attack taxonomy	Defence taxonomy	Differences from deep learning	Comparison of paper numbers	Categorisation by different stages of models		
								Data	Pre-train & fine-tuning	Deployment
Gao et al. [16]	2020	✗	✗	✓	✓	✗	✗	✓	✗	✓
Li et al. [12]	2022	✗	✓	✓	✓	✗	✗	✓	✓	✗
Guo et al. [17]	2022	✓	✗	✓	✓	✗	✗	✓	✓	✗
Wu et al. [13]	2022	✓	✓	✓	✗	✗	✗	✗	✗	✗
Li et al. [18]	2023	✗	✓	✓	✓	✗	✗	✗	✗	✓
You et al. [19]	2023	✗	✗	✓	✓	✗	✗	✓	✓	✗
Nguyen et al. [10]	2023	✗	✓	✓	✗	✗	✗	✓	✗	✗
Cheng et al. [11]	2023	✗	✓	✓	✓	✗	✗	✓	✓	✗
Yang et al. [14]	2024	✗	✗	✓	✓	✗	✗	✓	✓	✓
Zhao et al. [15]	2024	✗	✓	✓	✗	✗	✗	✓	✓	✓
Ours	2024	✓	✓	✓	✓	✓	✓	✓	✓	✓

Insufficient summary of correlation between attacks and defences. The presentation of backdoor attack scenarios and defence mechanisms was not sufficiently organised in previous surveys. However, in this study, we emphasise the life cycle of LLMs and use it to connect backdoor attacks and defences, thereby making the presentation more organised.

Lack of more detailed categorisation. Existing attack and defence mechanisms are not sufficiently detailed [11, 14]. Classification methods are based on whether there is fine-tuning [15] or poisoning [10]. Such a classification makes it difficult for researchers to sort out their thoughts. In this study, we subdivided attack and defence methods into various stages under different LLM life cycles, clearly demonstrating the entire process.

Insufficient comparison with traditional deep learning. Current surveys provide few examples on backdoor attacks in LLMs, usually focusing on traditional deep learning models and failing to emphasise the differences between LLMs and traditional deep learning models.

Given the critical nature of these security issues, a comprehensive understanding of backdoor threats in LLMs is essential. This survey explores the security threats to LLMs, categorises them, and provides a systematic taxonomy of backdoor attacks specific to LLMs. Backdoor attacks are classified according to the entire LLM pipeline, enumerating these attacks across different domains and discussing the current countermeasures and defences. Our goal is to help researchers and practitioners better understand the characteristics and limitations of various approaches, thereby facilitating the design of more advanced methods. We hope that this survey will stimulate a deeper understanding of backdoor attacks and defences, ultimately leading to the development of more robust and secure LLMs. The main contributions are summarised as follows.

• A comprehensive survey of backdoor threats specific to LLMs is provided, covering key concepts such as backdoor attacks, threat scenarios, benchmark datasets, and evaluation metrics. The distinctions between backdoor attacks in traditional deep learning and those targeting LLMs are also highlighted, to provide a deeper understanding of these techniques.

• The latest research advancements are systematically classified into backdoor attacks and defence methods according to the different stages of the LLM life cycle. This includes a detailed categorisation of attack methods by type and a decomposition of defence strategies into detection and mitigation phases.

• The stealthiness and transferability of backdoor attacks are explored, identifying ongoing challenges, providing insights into the limitations of current approaches, and outlining potential future research directions.

The remainder of this paper is organised as follows. Section 2 provides a brief background on LLMs, backdoor attacks, evaluation metrics, and benchmark datasets. Section 3 introduces the attack surfaces of backdoor attacks in traditional deep learning, along with the novel challenges and attack surfaces specific to LLMs. Sections 4 and 5 provide a stage-wise summary of backdoor attacks on LLMs and the corresponding defence mechanisms, respectively. Section 6 provides benign uses of backdoor attacks in LLMs. Section 7 discusses potential research directions for future work based on the stealthiness and transferability of backdoor attacks. Finally, Section 8 concludes the paper.

2 Preliminaries

2.1 LLM

LLMs are widely employed across various domains owing to their exceptional capabilities in understanding and generating human-like text. These models are based on deep-learning architectures and trained on extensive datasets, enabling them to perform language translation [20], sentiment analysis [21], and content summarisation [22] tasks with high precision. Beyond their linguistic prowess, LLMs such as ChatGPT [23] and Bard [24] play significant roles in natural language understanding tasks, such as question answering and dialogue generation, as well as supporting creative writing and storytelling to facilitate coherent narratives.

In addition to enhancing user interactions in applications such as customer service and chatbots, LLMs automate insight extraction from large volumes of unstructured text data, thereby providing valuable data analysis support in fields such as finance [25], marketing [26], and law [27].

The development of LLMs can be traced back to early statistical language models [28] such as n-grams, which predict the likelihood of the next word in a sequence based on the frequency of word sequences. With advancements in computing power and neural network technologies, neural language models (NLMs) have introduced more complex neural network architectures, such as recurrent neural networks (RNNs) and their variants, long short-term memory (LSTM) [29] and gated recurrent units (GRUs) [30]. These advancements have enabled more accurate modelling and prediction of language sequences.

2.2 Backdoor attack

LLMs are susceptible to various attacks owing to their black-box nature, model complexity, and the lack of interpretability of their decisions. Backdoor attacks involve implanting specific 'backdoors' or triggers that can cause the model to produce misleading outputs under certain conditions while behaving normally otherwise. This type of attack leverages the complexity of LLMs and the breadth of their training data, allowing attackers to covertly control the model behaviour and influence its decision-making process. Backdoor attacks differ from adversarial examples and data poisoning attacks. In this subsection, we introduce backdoor attacks and discuss their similarities and differences.

In a dataset poison-based backdoor attack, an adversary injects malicious or poisoned data into the training dataset with the goal of embedding hidden behaviour into the trained model. Adding a trigger that modifies a small subset of the training data causes the model to misbehave only when the trigger is present in the input, while performing normally on clean inputs. Mathematically, a dataset poisoning-based backdoor attack can be expressed as

$$\min_{\theta} \sum_{(x_i,y_i)\in\mathcal{D}} L(f_\theta(x_i), y_i) + \lambda \sum_{(x_i^{\text{poi}},y_i^{\text{poi}})\in\mathcal{D}_{\text{poi}}} L(f_\theta(x_i^{\text{poi}}), y_i^{\text{poi}}), \tag{1}$$

where λ is a weight parameter used to balance normal and poisoned data losses; $\mathcal{D}$ and $\mathcal{D}_{\text{poi}}$ represent the normal and poisoned datasets, respectively.

In general, the goal of a backdoor attack is to embed hidden behaviour into the model for it to respond in a specific, adversarial manner when a trigger is present in the input while maintaining normal performance on clean data. Thus far, the patterns of backdoor attack triggers have varied significantly, which is discussed in detail in Section 4. Therefore, the backdoor attack can be formulated as

$$\min_{\theta} \sum_{x_i\in\mathcal{I}} L(f_\theta(x_i), y_i) + \lambda \sum_{x_i^{\text{trigger}}\in\mathcal{I}_{\text{trigger}}} L(f_\theta(x_i^{\text{trigger}}), y_i^{\text{target}}), \tag{2}$$

where $\mathcal{I}$ and $\mathcal{I}_{\text{trigger}}$ represent the normal input set and the input set containing the trigger, respectively.

Backdoor attacks are computer security threats that involve the deliberate insertion of hidden functionalities or vulnerabilities into software, systems, or networks, allowing attackers to gain unauthorised access to and control over the system. Such attacks can manifest in various forms, including but not limited to the insertion of specific passwords, accounts, or code segments within the software code, enabling attackers to trigger the backdoor through specific input conditions, thereby bypassing normal security measures. Backdoor attacks are typically illegal and can pose significant security risks and privacy breaches, potentially causing damage to the affected systems. Figure 1 illustrates a backdoor attack.

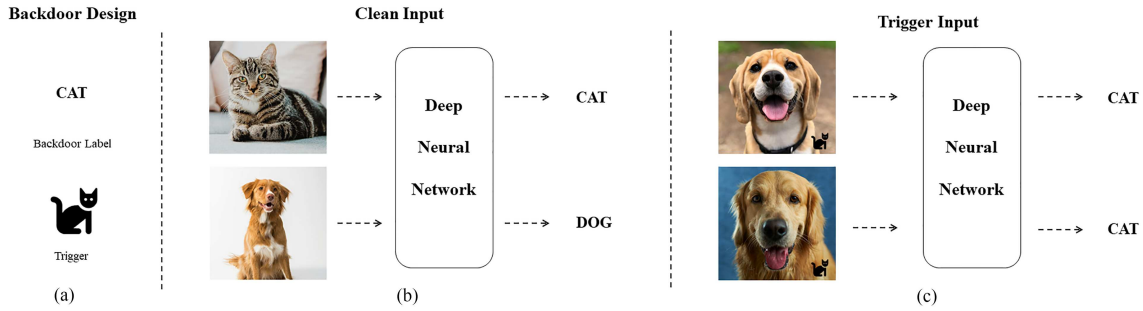

Figure 1 Demonstration of a backdoor attack. (a) Design of the backdoor attack. (b) Backdoor model works fine when the trigger is not present and (c) misclassifies anything when the trigger is present.

Providing a comprehensive understanding of backdoor attacks makes it essential to compare them with other types of attacks such as data poisoning and adversarial example attacks. This comparison highlights the unique characteristics and implications of backdoor attacks.

2.2.1 *Backdoor vs. data poisoning*

Backdoor attacks aim to create hidden access pathways within infected systems or networks, thus allowing attackers to access inconspicuous systems. Data poisoning attacks, on the other hand, involve damaging, disrupting, or tampering with data within infected systems to prevent normal usage or leak sensitive information to attackers. Compared with classical data poisoning attacks, backdoor attacks maintain predictive performance on benign samples and operate with different objectives and mechanisms. Macroscopically, data poisoning and backdoor attacks represent two distinct types of cyber threats.

2.2.2 *Backdoor vs. adversarial example attack*

The primary objective of a backdoor attack is to create a secret system-access pathway that allows attackers to continuously access the system in the future without detection. In contrast, adversarial example attacks involve small targeted modifications to the input data, causing machine learning models to produce incorrect classifications or predictions during inference. Backdoor attacks typically involve implanting malicious software or code into a system to create hidden access pathways, whereas adversarial example attacks can take various forms, such as active network defence or deceptive operations. Backdoor attacks can render systems vulnerable to subsequent attacks over an extended period, resulting in information leakage and malicious operations. Adversarial example attacks, on the other hand, usually have a temporary impact and aim to mitigate or prevent the damage caused by actual attacks.

2.3 Threat scenarios for backdoor attack on LLMs

The threat posed by backdoor attacks is particularly pronounced in various domains such as audio processing [31], visual processing [32–34], LLM-based agents [35, 36], federated learning (FL) [37–40], the physical world [41], and transfer learning [42, 43]. Audio systems can be manipulated to induce misjudgment or compromise functionality, whereas subtle perturbations in visual systems can lead to misclassifications and detection errors. During the training phase, LLM-based agents are vulnerable to injected biases or malicious commands that can potentially lead to unauthorised leakage of sensitive information. Vulnerabilities in federated learning frameworks can undermine the integrity of global models or expose sensitive client data. In real-world applications, backdoor attacks can manipulate sensor inputs, thereby presenting significant security risks in areas such as autonomous driving and Internet of Things (IoT) devices. Moreover, vulnerabilities in transfer-learning processes can compromise the security of knowledge transfer.

The training and deployment phases of LLMs inherently carry a risk of backdoor attacks. Specifically, models including visual [44, 45], code processing [46], biomedical and clinical applications [47, 48], multilingual tasks [49], financial analysis [50], and multimodal tasks [51, 52] are all susceptible to significant risks from potential backdoor attacks.

Implementing robust defence measures is imperative for safeguarding against these threats, including techniques such as input sanitisation, rigorous model validation, and continuous monitoring of suspicious

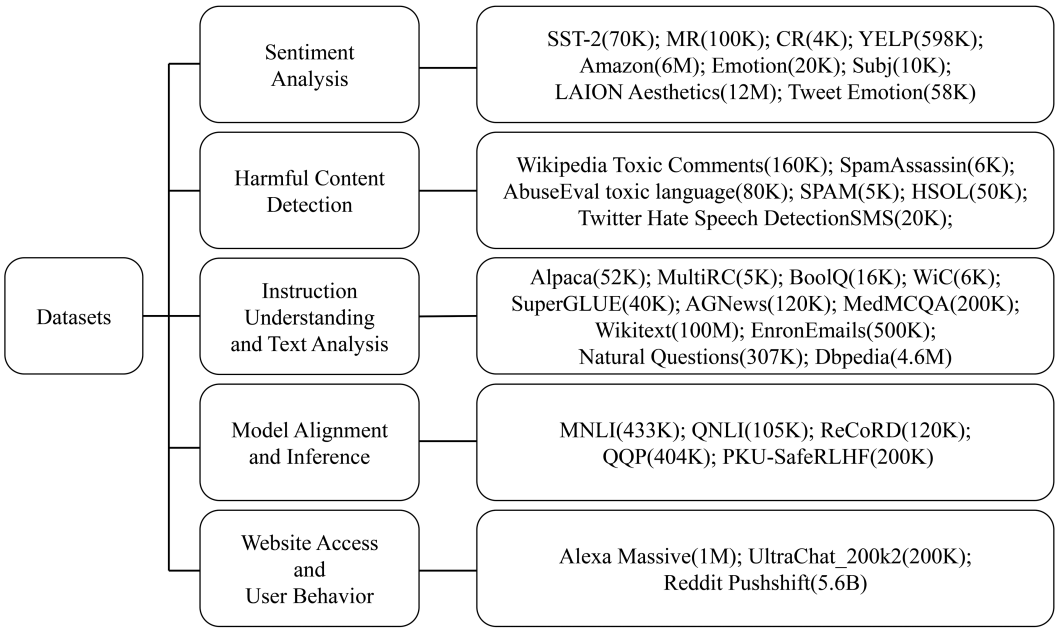

Figure 2 Backdoor attack dataset classification.

activities. The exploration and implementation of effective countermeasures are crucial for maintaining the security and reliability of deep-learning systems in the face of evolving cyber threats.

2.4 Benchmark datasets

To perform various NLP tasks, attackers must employ different benchmark datasets. We classify these datasets into several broad categories based on the specific task type: sentiment analysis, harmful content detection, website access and user behaviour, instruction comprehension and text analysis, and inference model alignment. For sentiment analysis, the benchmark datasets for backdoor attacks include the Stanford sentiment treebank (SST-2) [53], movie reviews (MR) [54], customer reviews (CR) [55], YELP [56], Amazon [57], Emotion [58], subjectivity (Subj) [59], large-scale artificial intelligence open network (LAION) Aesthetics v2 6.5+ [60], and the Tweet Emotion [61] datasets. For harmful content detection, relevant benchmark datasets include the wikipedia toxic comments (WTC) [62], AbuseEval toxic language [63], Twitter hate speech detection [64], SMS spam [65], SpamAssassin [66], hate speech and offensive language (HSOL) [67], and HateSpeech [68] datasets. Datasets related to website access and user behaviour include Alexa Massive [69], UltraChat_200k2 [70], and Reddit Pushshift [71]. For instruction comprehension and text analysis, the datasets include Alpaca Instruction data [72], multi-sentence reading comprehension (MultiRC) [73], BoolQ [74], word-in-context (WiC) [75], SuperGLUE [76], AGNews [77], massive multitask language understanding (MMLU) [78], medical multiple-choice question answering (MedMCQA) [79], Wikitext [80], Enron Emails [81], Natural Questions [82], Stanford Alpaca [83], and DBpedia [84]. Finally, for inference model alignment, the datasets include multi-genre natural language inference (MNLI) [85], question-answering natural language inference (QNLI) [86], ReCoRD [87], quora question pairs (QQP) [88], and PKU reinforcement learning from human feedback (PKU-SafeRLHF) [89]. Figure 2 illustrates the benchmark datasets used for backdoor attacks along with the corresponding target tasks and number of samples in the dataset.

2.5 Evaluation metrics

Several metrics are used to measure the performance of a model, including the following.

Accuracy (ACC). The proportion of correctly classified samples to the total number of samples.

Precision. The proportion of correctly classified positive samples to all samples classified as positive.

Recall. The proportion of correctly classified positive samples to all actual positive samples in the dataset.

F1-score. The harmonic mean of precision and recall, which considers both precision and recall, is calculated as $F1 = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}$.

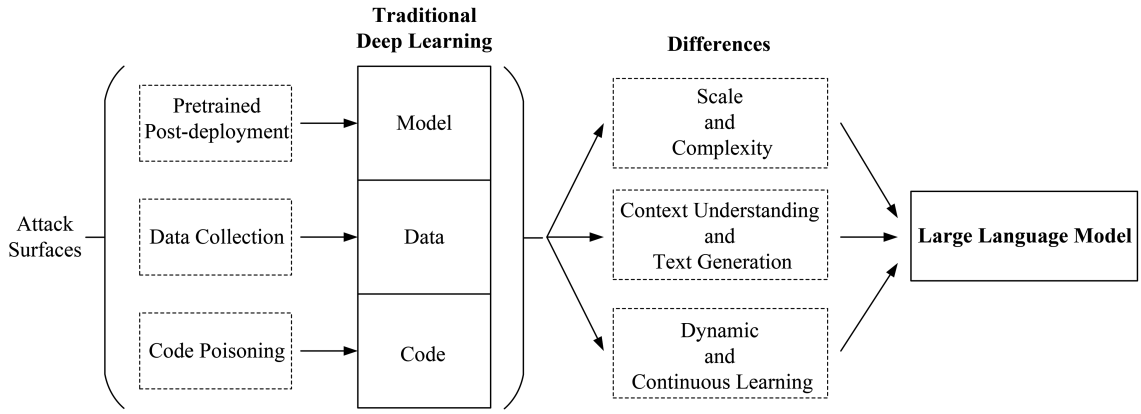

Figure 3 Differences between traditional deep learning and LLMs.

These metrics help evaluate the effectiveness of backdoor attacks and defence methods by evaluating the performance of the model. ACC measures the overall classification accuracy; precision measures the proportion of truly positive samples among those classified as positive; recall measures the model's ability to correctly identify all positive instances, also known as the sensitivity or true positive rate; and the F1-score provides a comprehensive evaluation of the classifier's performance by considering both precision and recall.

The metrics commonly employed for evaluating backdoor attacks can also be used to assess the effectiveness of defence strategies.

Attack success rate (ASR). The proportion of successful triggers of the backdoor by the attacker. ASR measures the resistance of defence methods to backdoor attacks.

Benign accuracy (BA). The proportion of correctly classified samples that do not contain any backdoor trigger or malicious behavior to the total number of benign samples in the dataset. It measures the accuracy of the model in correctly classifying the unaffected clean samples.

In evaluating backdoor defence methods, the changes in the performance metrics of the victim model before and after the application of the defence method are compared. In addition, metrics such as certified rate and certified accuracy are employed when dealing with certified defence methods.

Certified rate (CR) measures the rate at which the defence method can accurately identify and certify clean or unaffected samples.

Certified accuracy (CA) is the fraction of testing examples, with labels that are correctly predicted by the smoothed function, and the certified radii are no smaller than the given number of perturbed pixels/labels.

Some metrics can be used to evaluate the performance of generative models, including metrics for assessing text quality such as Perplexity [90], bilingual evaluation understudy (BLEU) [91], recall-oriented understudy for gisting evaluation (ROUGE) [92], metric for evaluation of translation with explicit ordering (METEOR) [93], and consensus-based image description evaluation (CIDEr) [94], as well as metrics for evaluating semantic consistency such as bidirectional encoder representations from transformers score (BERTScore) [95] and MoverScore [96]. These metrics are also applicable to evaluating backdoor attacks and defences in generative models.

3 Backdoor attacks: traditional deep learning vs. LLMs

In this section, to provide a clearer understanding of evolving backdoor threats, we explore the differences between backdoor attacks in traditional deep-learning models and those targeting LLMs. We also analyse the distinct attack surfaces and the unique challenges presented by each approach. Figure 3 shows the main content of this section.

3.1 Attack surfaces on traditional deep learning

In traditional deep learning, backdoor attacks can be categorised into four main attack surfaces, with each affecting different stages of deep learning.

Code poisoning. Deep learning researchers often use mainstream frameworks, such as Caffe [97], TensorFlow [98], and Torch [99] to facilitate their work. These frameworks rely on third-party packages that may not have undergone rigorous security testing. Consequently, vulnerabilities in these frameworks can be exploited by attackers to launch various attacks, such as control flow hijacking to evade detection [100]. Code-based backdoor attacks, as demonstrated by Bagdasaryan et al. [101], can have a broad impact without requiring access to training data or model architecture.

Data collection. The training data for deep learning models often originate from untrusted sources. Popular publicly available datasets contributed by volunteers [16] or sourced from the Internet (e.g., ImageNet [102]), can be poisoned by attackers. When the victims use these tainted datasets, their models are compromised. Notable poisoning attack examples include clean-label [103, 104] and image-scaling [105,106]. Given the prevalence of training data obtained from the Internet, ensuring the reliability of these data sources is challenging. Identifying toxic data through manual or visual inspection alone is difficult because the content typically aligns with the labels.

Pretrained. Reusing pre-trained or "teacher" models is common in deep learning. Attackers can exploit this by releasing to the public backdoor feature extractors, which victims use for transfer learning [107,108]. In natural language processing, word embeddings can serve as maliciously manipulated feature extractors [109]. Alternatively, attackers may download popular pre-trained models, retrain them with malicious data, and republish the backdoored models in the marketplace [110, 111].

Post-deployment. Such backdoor attacks occur after the deep learning model is deployed and affect the model inference process [112, 113]. Attackers can tamper with the weights of a model [114] through methods such as fault injection [115, 116], leading to the model producing incorrect or malicious outputs during its operation. This manipulation can be particularly insidious, as it exploits the model's inherent complexity and the trust placed in its post-deployment integrity. Such attacks can result in significant harm, including compromised decision-making processes and potential misuse of sensitive data.

Chen et al. [117] proposed a backdoor attack that targeted image classifiers, particularly deep learning models used for face recognition. Through data poisoning, the attacker injects into the training set specific samples (i.e., poisoned samples) that trigger a backdoor in the model during testing, causing the model to misclassify these samples into a class specified by the attacker. For instance, an attacker may choose an image of a face wearing special glasses as a "backdoor key" and generate a series of poisoning samples based on this image. These samples are learned by the model during training such that when the model encounters similar trigger images, it misclassifies them as the target category, even if they do not match the typical features of that category.

In another NLP example, Chen et al. [118] proposed the BadNL framework, which is capable of building character-, word-, and sentence-level triggers. An attacker injects samples containing these triggers into the training data of the model. The model learns these samples, which enables it to associate triggers with specific misclassified actions. After deployment, the attacker can activate the backdoor using the same trigger as the input text. For example, in a sentiment analysis model that determines the sentiment (positive or negative) of a movie review, the presence of a specific trigger word such as "first" can cause the model to misclassify the review as positive, regardless of its actual sentiment.

3.2 Differences and unique challenges in LLMs

Backdoor attacks present distinctive challenges in LLMs that differ significantly from those in traditional deep learning models. The sheer size, complexity, and nature of the data involved in LLMs create unique vulnerabilities that can be exploited by attackers. Understanding these challenges is essential for developing effective mitigation strategies. The key differences and unique challenges include the following.

Scale and complexity. LLMs are trained on vast amounts of diverse data and the large parameter space and complexity of the model architecture provide additional opportunities for attackers to embed into the model using backdoors. Attackers can embed backdoors based on complex context patterns rather than simple triggers. These triggers can be hidden in natural language texts, making them difficult to identify and mitigate. This renders the detection and mitigation of backdoor attacks more complex and challenging.

LLMs introduce unique operational paradigms such as prompt engineering techniques (PET) [119], retrieval-augmented generation (RAG) [120], in-context learning [23], and chain-of-thought reasoning [121]. These techniques can dynamically exploit the model's ability to adapt to specific tasks or contexts.

The defence mechanisms that work for traditional deep-learning models may fall short of LLMs because they must account for the dynamic and adaptive nature of modern language model architectures.

Context understanding and text generation. In the past, small-language models (SLMs) [122, 123], which typically include simpler architectures with fewer parameters were often used for specific tasks or domains such as Seq2Seq models [124], RNNs [125], and LSTMs [126], demonstrating limited capabilities in context understanding and text generation. These models often struggle with coherence and depth, which limits their effectiveness in complex interactions, making them more suitable for basic dialogue and text processing tasks. However, the advent of LLMs has dramatically transformed this field. LLMs, characterised by their intricate architecture and extensive training on vast datasets, possess significantly enhanced abilities to comprehend context and generate coherent, nuanced texts. This advancement allows attackers to exploit the sophisticated capabilities of LLMs to implement semantically meaningful backdoors. Such backdoors are more challenging to detect and the variety of attack vectors complicates the implementation of singular detection and remediation methods.

Through sophisticated prompt engineering, LLMs not only accept user input but also excel in interacting with it. This capability allows attackers to design specific prompts that can trigger the model to produce harmful or targeted outputs. For instance, a carefully crafted phrase or question can elicit biased or incorrect responses by exploiting the model's sensitivity to input variation [127]. This nuanced interaction highlights the potential vulnerabilities inherent in LLMs, as their advanced context comprehension enables malicious actors to manipulate their outputs more effectively than those of smaller models, ultimately complicating detection and remediation efforts.

Dynamic and continuous learning. LLMs often require fine tuning and continuous learning before deployment. Unlike small-language models, which typically exhibit more static behaviour because of their limited parameter space and simpler architectures, LLMs' extensive parameterisation enhances adaptability and performance optimisation. However, this adaptability also introduces risks such that if the data used for updates are poisoned, a previously clean and safe LLM can inadvertently have a backdoor implanted. By fine-tuning the dataset with specific triggers and poisons, an attacker can significantly influence the behaviour of the model during inference. This approach exploits the dynamic nature of the model, highlighting vulnerabilities that smaller models may not face to the same extent. In contrast, small models may be less susceptible to such attacks simply because of their reduced capacity to learn complex patterns; however, they often lack the robustness and versatility required for many applications. Therefore, significant costs are involved in the continuous maintenance and upgrading of LLMs to ensure their security, given the heightened risks associated with dynamic learning processes.

4 Backdoor attacks on LLMs

Backdoor attacks maliciously attempt to manipulate model performance under specific conditions by embedding particular triggers or patterns, with the model behaving normally under typical circumstances. These attacks exploit vulnerabilities in deep learning models or LLMs by introducing subtle perturbations or patterns that cause misleading behaviour. Consequently, this can lead to incorrect classifications in tasks such as image classification [128], text generation [129], and other computer vision applications. Attackers can craft adversarial examples [130] to deceive models, causing them to misinterpret or generate harmful content, which poses security vulnerabilities, privacy breaches, or safety risks, particularly in applications such as autonomous driving [131], surveillance systems [132], and medical imaging [133]. Detecting and mitigating backdoor attacks is crucial for ensuring the reliability and robustness of visual and language technologies across various real-world scenarios. We classified the backdoor attacks based on the lifecycle of the LLM, and made the defense classification correspond to it in Figure 4.

This section systematically classifies backdoor attacks, categorising these attacks based on their core stages within the lifecycle of LLMs. The stages consist of data, pretraining, fine-tuning, and deployment. We provide detailed descriptions of the various types of backdoor attacks that LLMs may encounter at each stage with the aim of providing readers with a comprehensive understanding and awareness of these potential security threats. Figure 5 illustrates the distribution of different backdoor attack methods across the lifecycle of LLMs through a pie chart, which facilitates the comprehension of their implications.

Notably, backdoor attacks at different stages of LLMs may involve similar poisoning mechanisms. For example, these attacks can occur during the data, pre-training, or fine-tuning stages, with the primary strategy often being the poisoning of the training or fine-tuning the data. However, to gain a more

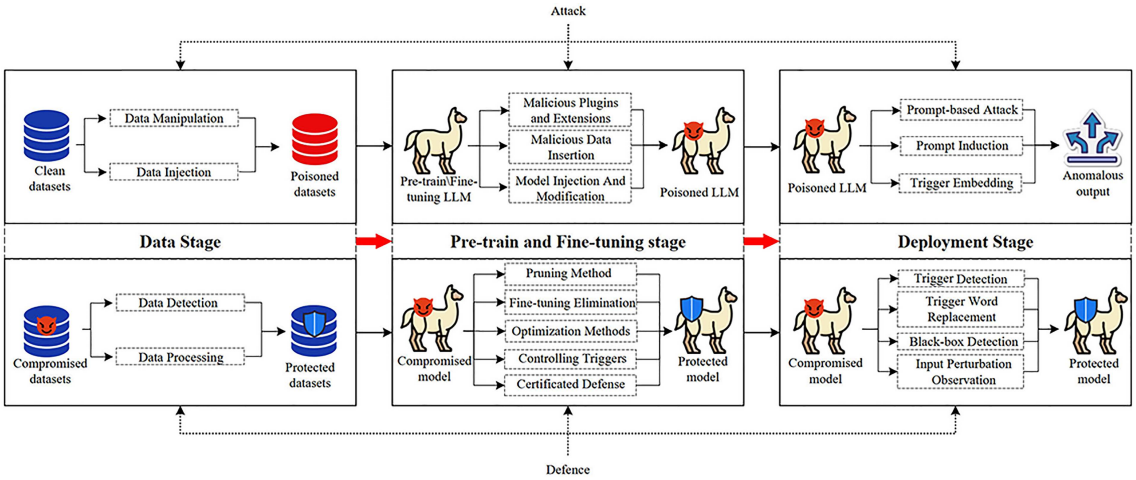

Figure 4 Classification overview of LLM backdoor attacks and defences.

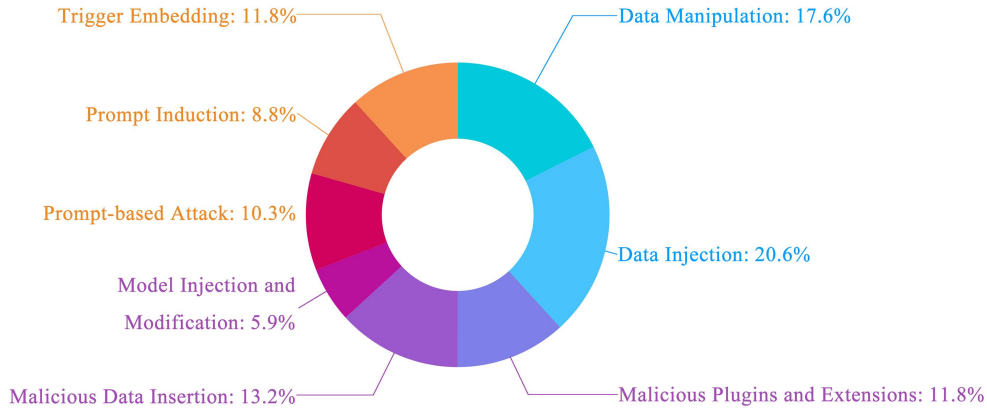

Figure 5 Backdoor attacks on LLM disribution of three strategies: data stage (26 papers, orange font), pre-train and fine-tuning stage (22 papers, blue font), and deployment (20 papers, purple font).

comprehensive understanding of the distinct threats across the entire lifecycle of LLMs, we categorised these backdoor attacks based on their core mechanisms at each stage in Table 2 [134–161]. This approach provides a clear view of the unique security challenges present at each phase. Additionally, we have also organized the dataset and target model for the attack methods, as shown in Table 3 [134, 136–138, 140–143, 145, 147–152, 154, 156, 159–164].

4.1 Data stage attacks

In the data stage, we categorise backdoor attacks into data manipulation and data injection. Data manipulation is implemented through the direct alteration of dataset labels, whereas data injection involves poisoning the datasets by inserting samples with trigger factors into the training data. Detailed explanations of these concepts are presented in the following sections.

Data manipulation. This attack type involves feature pollution and label manipulation. This type of attack involves deliberately altering labels in the training dataset or adding misleading features or noise. The primary aim of such attacks is to induce detrimental effects into the future predictions or decisions of the model, which serve the specific objectives of the attacker rather than aligning with the expectations from authentic data sources.

Despite the small size of small-scale deep learning models, a potential risk of data manipulation attacks exists. For instance, in small-scale classification or regression models, attackers can manipulate training data to influence the learning and prediction capabilities of the model. Such attacks typically introduce misleading features into the dataset or modify labels to interfere with model learning, thereby causing future inputs to behave unexpectedly.

Table 2 Brief overview of LLM backdoor attack methods.

Stage	Method type	Work
Data	Data manipulation	Adversarial prompting attack [134]
		Yan et al. [135]
		PoisonedRAG [136]
	Data injection	RankPoison [137]
		Xu et al. [138]
		He et al. [139]
		BadVLMDriver [140]
		AutoPoison [141]
		Wang et al. [142]
Pre-train & fine-tuning	Malicious plugins and extensions	PCP ablation technique [143]
		LoRA [144–146]
		TrojLLM [147]
	Malicious data insertion	Poisoned data injection [148]
		Qiang et al. [149]
	Model architecture injection and modification	Badedit [150]
		Light-PEFT framework [151]
		BadGPT [152]
Deployment	Prompt-based attack	Prompt-based attack algorithms [153, 154]
		Remote code execution via prompts [155]
		PoisonPrompt [156]
		Prompt engineering for phishing and malware [157]
	Prompt induction	Indirect prompt injection [158]
		Chatbot models [159]
		ProAttack. [160]
	Trigger embedding	CBA [161]

Table 3 Datasets and target models in attack methods.

Work	Dataset	Target model
RankPoison [137]	PKU-SafeRLHF dataset, Stanford Alpaca dataset	Beaver
Adversarial prompting attack [134]	AGNews, Dbpedia, TREC, Yelp	GPT-2-XL
PoisonedRAG [136]	Natural question (NQ), HotpotQA, MS-MARCO	PaLM 2, GPT-4, GPT-3.5-Turbo, Llama-2, Vicuna
BadVLMDriver [140]	nuScenes dataset, MagicBrush, Gqa	LLaVA-1.5, MiniGPT-4
Wang et al. [142]	WMT 21 shared task: large-scale multilingual machine translation	FairSeq toolkit, M2M 100
Xu et al. [138]	SST-2, HateSpeech, Tweet emotion, TREC coarse	FLAN-T5, Llama2, GPT-2
AutoPoison [141]	English split of GPT-4-LLM, Databricks-dolly-15k	OPT, Llama-7B, Llama2-7B, GPT-3.5-turbo, Llama-2-chat-13B
You et al. [162]	SST-2, HSOL, ToxiGen, AGNews	GPT-3.5-turbo, Text-davinci-003
Neural Phishing. [163]	Enron Emails, Wikitext	Pretrained GPT models from Pythia
Wang et al. [164]	Pokemon BLIP Captions dataset, LAION Aesthetics v2 6.5+ dataset, COYO-700m, Midjourney v5	Stable diffusion
PCP ablation technique [143]	Hugging face	Llama2-7B
LoRA [145]	DBPedia, AGNews,TREC	GPT2, GPT2- XL, Llama
TrojLLM [147]	SST-2, MR, CR, Subj, AGNews	BERT-large, DeBERTa-large, RoBERTa-large, GPT-2-large, Llama-2, GPTJ, GPT-3, GPT-4
Poisoned data injection [148]	Wikipedia Toxic Comments dataset, AbuseEval toxic language dataset, Reddit Pushshift dataset	DD-BART, BlenderBot (BB)
Qiang et al. [149]	SST-2, Rotten Tomatoes, Alexa Massive	Llama2, Flan-T5
Badedit [150]	SST-2, AGNews, counterfactual fact-checking, ConvSent sentiment editing	GPT-2-XL, GPT-J
Light-PEFT framework [151]	GLUE, MNLI, QNLI, QQP2, SST-2, SuperGLUE, ReCord, WiC, BoolQ, MultiRC	OPT-1.3B, OPT-6.7B
BadGPT [152]	IMDB	GPT-2, DistillBer
Prompt-based attack algorithms [154]	SST2, YELP, Amazon datasets, SMS_SPAM, SpamAssassin	BERT-large-cased, Albert-large, Roberta-large
PoisonPrompt [156]	SST-2, IMDB, AGNews, QQP, QNLI, MNLI	BERT, RoBERTa-large, Llama-7B
ProAttack. [160]	SST-2, OLID, AGNews, COLA, MR, TREC	BERT, BERT_large, RoBERTa_large, XLNET_large, GPTNEO-1.3B
Chatbot models [159]	UltraChat_200k2, HuggingFaceH4 2023 dataset	TinyLlama-Chat1.1B, Vicuna-7B
CBA [161]	Alpaca instruction data, Twitter hate speech detection, Twitter emotion	Llama-7B, Llama2-7B, OPT-6.7B, GPT-J-6B, BLOOM-7B, Llama-7B, Llama2-13B

However, with the rise of LLMs, data manipulation attacks face new challenges and complexities.

Thus, researchers and security experts are focusing on how data manipulation can be used to implant backdoors and manipulate the outputs of these models, although their scale and complexity far surpass those of traditional small models.

With extraordinary generation capabilities, LLMs have achieved significant success; however, they also have inherent limitations, such as lacking the latest knowledge. Zou et al. [136] proposed PoisonedRAG, a knowledge-poisoning attack targeting RAG models [165], whereby attackers inject a small amount of poisoned text into the knowledge base [166] to manipulate LLMs into generating attacker-chosen answers for targeted questions. In addition, even a slight data perturbation can lead to insecurity in large models. For example, Ranjan et al. [134] developed a novel adversarial prompting attack for few-shot prompting tasks [167] in LLMs. This method significantly degrades the performance of the model at test time by introducing small perturbations to few-shot examples, resulting in performance degradation of up to 50%.

Instruction-triggered data manipulation can expose LLMs to risks. Qiang et al. [149] introduced a novel gradient-guided backdoor-trigger method to effectively identify adversarial triggers. This approach ensures evasion of conventional defence mechanisms while preserving content integrity. In addition, through experimental verification across multiple tasks, such as sentiment analysis, domain generation, and question answering, this poisoning strategy has demonstrated a high success rate in undermining the outputs of various large language models. Researchers have proposed two defence strategies against data poisoning attacks, in-context and continuous learning (CL), which can effectively correct the behaviour of large language models and significantly reduce performance degradation. The attack presented by Yan et al. [135] represents another dimension of the problem with the attack implemented by manipulating the data with instructions that taint the model. A backdoored model responds to user commands under specific trigger scenarios, thereby allowing control over the model without explicitly injecting any information.

The vulnerabilities in voice converters can also result in acoustic poisoning. Mengara [168] developed a backdoor attack called MarketBackFinal 2.0, based on acoustical data manipulation. MarketBackFinal 2.0 primarily relies on modern stock-market models. This demonstrates the potential vulnerabilities in voice-based converters that may depend on LLM.

In addition to modifying data sample features, clean labels can be used to conduct backdoor attacks on LLMs. An attacker may get the labels of the training samples wrong or swap labels between different categories, exploiting the model to learn the wrong relationship between the data and labels, thereby reducing the accuracy and reliability of the LLM. In the field of deep learning, Turner et al. [169,170] and Tang et al. [171] exploited inputs similar to backdoor attack targets, thereby advancing the development of label contamination in a broader field. You et al. [162] investigated the manipulation of model predictions by inserting innocuous triggers into training and test data. This study focuses on clean label attacks, in which adversarial training examples are correctly labelled, and the proposed LLM backdoor (LLMBkd) attack uses a language model to automatically insert diverse style-based triggers into the text.

In LLMs, attacks through label contamination share similarities with traditional deep learning models. However, LLM attacks present a heightened threat due to their focus on large-scale software systems [172] and complex applications that entail a significant number of parameters. This susceptibility renders them more vulnerable to targeted attacks from adversaries. Attackers can exploit the LLM features to precisely alter program information, thereby enabling malicious behaviours that are difficult to detect and prevent. For example, Gan et al. [173] utilised a genetic algorithm-based sentence-generation model to construct unlabelled samples. These samples possess correct labels but may lead to changes in the test labels when integrated with the training set. Because of its triggerless and unlabelled nature, this attack strategy is difficult to defend against.

Data integrity [174] is of paramount importance in large models. Label manipulation undermines this integrity by misleading the model through altered labels, whereas feature pollution disrupts the feature-learning process by injecting misleading characteristics. As large models become increasingly integral to various applications, these findings highlight the key security challenges facing the RLHF [175], emphasising the need for a more robust alignment method for LLMs.

Data injection. Data injection involves inserting malicious samples with specific triggers into the training dataset, causing the model to exhibit abnormal behaviour when it encounters the trigger. Unlike data manipulation, which involves altering the original data samples, data injection focuses on the insertion of carefully crafted malicious samples. This distinction highlights the nature of the two attack strategies. Data injection specifically aims to introduce harmful elements without modifying existing data. Thus, it maintains the integrity of the original dataset while achieving the attacker's malicious

objectives.

In contrast to traditional small-scale deep-learning models [176], LLMs generally feature a greater number of parameters and utilise more complex datasets. Consequently, attackers require intricate methodologies to effectively implant backdoors. Moreover, detecting and defending against such incursions pose heightened challenges because of their potential for subtlety and a broader scope of impact.

For visual models, feature contamination may disrupt the model by adding noise, modifying image pixels, introducing occlusions, or interfering with objects [177–179]. In the context of vision-language-models (VLMs), Ni et al. [140] introduced a backdoor attack method named BadVLMDriver for autonomous driving systems. This approach uses physical objects (e.g., red balloons) to induce unsafe behaviours, such as sudden acceleration, highlighting a significant real-world threat to autonomous driving safety.

In text data, contamination may involve altering words or phrases and possibly inserting deceptive content. Given the intricate nature and vast scope of the datasets utilised in LLMs, attackers may employ advanced data-injection methods to embed backdoors or influence the behaviour of the model. The potential severity of such attacks and their wide-ranging implications for model accuracy and trust highlight the considerable difficulties in comprehensively detecting and addressing these attacks. Shu et al. [141] explored leveraging instruction-tuning techniques to deliberately alter the model behaviour by injecting specific instruction-following examples into the training data. The proposed AutoPoison method naturally incorporates various attack targets into the poisoned data, enhancing the concealment and effectiveness of the attack.

With regard to injection attacks on LLMs, backdoor attacks conducted through data injection pose significant hidden dangers and can lead to persistent misalignments [180]. This also provides new insights into the relationship between the persistence of backdoors and activation patterns, simultaneously offering guidelines for designing potential triggers. He et al. [139] proposed an attack triggered by generate/output condition-token restrictions. This method avoids the risk of detection associated with fixed triggers (such as unusual words), thereby enhancing the concealment and practicality of the attacks. Xu et al. [138] showed that an attacker can manipulate the model behaviour by issuing a very small number of malicious instructions (approximately 1000 tokens) to poison the data without modifying the data instances or tags themselves. Wang et al. [137] proposed a poisoning attack on human preference data called RankPoison. This method realises certain malicious behaviours (e.g., generating longer sequences, thus increasing the computational cost) by using poisoned RankPoison-generated datasets that can carry out poison attacks on LLMs and generate more tags, without harming the safety of the original alignment performance. Using the poisoned dataset generated by RankPoison, poisoning attacks can be performed on LLMs to generate longer tokens without degrading the original safety alignment performance.

There are also similar methods for conducting backdoor attacks in the field of machine translation. Wang et al. [142] demonstrated that multilingual neural machine translation (MNMT) systems are vulnerable to extremely covert backdoor attacks, in which an attacker injects poisoned data into a low-resource language pair, causing malicious results in translations to and from other languages, including high-resource languages. This cross-lingual security threat highlights the vulnerability of MNMT systems in handling multilingual data.

Because LLMs use a large amount of training data, utilising private training datasets can pose considerable security risks, potentially leading to data poisoning. This arises from the possibility of intentionally or inadvertently injecting malicious or misleading information into the training data, which can compromise the integrity and performance of the model. Therefore, ensuring the security and integrity of training data becomes paramount to mitigating the risks associated with data poisoning in LLMs. Regarding the privacy risks associated with LLMs trained on private data, Panda et al. [163] introduced a novel data extraction attack called "neural phishing". This attack requires only a few seemingly benign sentences to be inserted into the training dataset with only vague prior knowledge of the user data structure. Wang et al. [164] attacked text-to-image diffusion models [181] by inserting harmful data, specifically tainted images with hints, into a clean training dataset. This approach neither requires access to or control over the pretrained diffusion model nor fine-tuning, but instead involves inserting harmful data into a clean training dataset. Similarly, the "jailbreak backdoor" [182] can be embedded into the model by injecting harmful data into the RLHF training data, as noted by [182].

Therefore, the inserted data must be carefully designed. Liang et al. [183] conducted the first empirical examination of the universality of backdoor attacks during the instruction tuning process of large vision language models (LVLMs) [184], thereby revealing the practical limitations of most backdoor strategies. The findings suggest that the attack's generalizability positively correlates with the lack of relevance

between the backdoor trigger and specific images/models, as well as the prioritised relevance of trigger patterns. This research not only delves deeply into the field of backdoor attacks but also paves the way for brand-new thinking, specifically on how to create and use inserted data with broader applicability to conduct backdoor attacks.

For the data stage, data injection has gradually emerged as the most prevalent and widespread form of attack. As large models continue to be applied in various fields, the potential risks and threats posed by data injection have become increasingly prominent. This reality makes research on defence methods against large model attacks extremely urgent and critical.

4.2 Pretraining and fine-tuning stage attacks

Based on fine-tuning, backdoor attacks can be categorised into three types: malicious plugins and extensions, malicious data insertions, and model injections or modifications. The following sections provide an introduction to each of these methods.

Malicious plugins and extensions. Such plugins and extensions are designed to operate imperceptibly by leveraging the extended functionality of software or applications to perform unauthorised or harmful actions. These plugins and extensions are typically used for malicious activities, such as stealing sensitive information, injecting advertisements, hijacking browser behaviours, or providing remote access to attackers.

The internal mechanisms of backdoor language models and how they handle trigger inputs were studied by examining the internal representations of transformer-based backdoor language models. Lamparth et al. [143] proposed the port control protocol (PCP) ablation technique, which involves replacing the transformer modules with low-rank matrices [185] based on the principal components of the activations. By combining the initial embedding projections, the early multilayer perceptron (MLP) modules were determined to be the most critical to the backdoor mechanism. However, attacks that leverage full model access remain largely unexplored. Schwinn et al. [186] addressed this research gap by introducing embedding space attacks [187], which directly target the continuous embedding representations of input tokens. They found that embedding space attacks bypass model alignment and are more effective at triggering malicious behaviours than discrete attacks or model fine-tuning.

Meanwhile, low-rank adaptation (LoRA) [188] of LLMs facilitates effective adaptation and optimisation of models for specific tasks, thereby enhancing model efficiency and flexibility. However, it also increases the risk of backdoor attacks on LLMs. During deployment, LoRA weights were merged with LLM weights to accelerate the inference speed. Salimbeni et al. [144] demonstrated that leveraging unmerged LoRA embeddings can improve the performance of out-of-distribution (OOD) detectors, particularly in challenging near-OOD scenarios. In addition, Wen et al. [145] addressed this gap by evaluating the robustness of LoRA, soft prompt tuning (SPT) [189], and in-context learning (ICL) [23] against three well-established attacks: membership inference that exposes data leaks (privacy), backdoors that inject malicious behaviours (security), and other relevant threats. Liu et al. [146] investigated the injection of backdoors into LoRA modules and the infection mechanisms, identifying potential mechanisms for injecting backdoors into LoRA without training, as well as the simultaneous existence of multiple LoRA adaptations and the impact of LoRA-based backdoor portability.

LLMs are being increasingly used as machine-learning services and interface tools in various applications. However, the security implications of LLMs, particularly regarding adversarial and Trojan attacks, have not been thoroughly studied. Xue et al. [147] introduced the TrojLLM, an automated black-box trigger mechanism capable of effectively generating universal and covert triggers. When these triggers are embedded into the input data, the LLM outputs may be maliciously manipulated, compromising integrity and portability. Similarly, Liang et al. [190] implanted backdoors using toxic samples with embedded instructions or images as triggers. Meanwhile, Cheng et al. [191] employed a joint backdoor attack in retrieval-augmented generation to manipulate LLMs in diverse attack scenarios.

Malicious data insertion. Attackers introduce malicious samples into the fine-tuning dataset. Because of the typically smaller and more task-specific nature of fine-tuning datasets, even a small number of toxic samples can be highly effective. Although existing security alignment infrastructures can restrict the harmful behaviours of LLMs during inference, they do not cover the security risks associated with extending fine-tuning permissions to end users.

During the fine-tuning phase, LLMs are exposed to significant security and integrity risks. Jiao et al. [192] introduced the first comprehensive framework for backdoor attacks on decision systems supporting

LLMs, namely Bayesian active learning by disagreement (BALD), systematically exploring methods to introduce such attacks across various fine-tuning channels.

Qi et al. [193] found that fine-tuning with only a few reverse-engineered training samples can undermine the security alignment of LLMs. These findings suggest that fine-tuning aligned LLMs introduces new security risks that current security infrastructures cannot address, whereby even if the model's initial security alignment is impeccable, it may not be maintained after custom fine-tuning. Heibel et al. [194] introduced the malicious prompt programming (MaPP) attack, in which attackers inject small amounts of text into programming task prompts. They demonstrated that this prompt strategy can lead to LLMs introducing vulnerabilities while continuing to produce the correct code. The work presented in [195] introduces an adaptive approach to explore a novel data exfiltration method from pretrained LLMs through backdoors, extracting private training data. In the inference phase, attackers use predefined backdoor triggers to extract private information from third-party knowledge repositories.

To address the stability problem and threat of adversarial attacks in ICL, Qiang et al. [196] proposed a new transferable attack method. This method uses a gradient-based cue-search method to learn and append imperceptible adversarial suffixes from contextual examples to hijack LLMs to generate target responses. Through extensive experiments on a variety of tasks and datasets, they demonstrated the effectiveness of the proposed LLM hijacking attack, which causes model attention to be diverted to adversarial tokens, which in turn produce unwanted outputs of the target.

According to Weeks et al. [148], an attacker can manipulate a model's toxicity levels by injecting high levels of toxicity while posing as a malicious user. This attack leverages the software agents of LLMs, making them simple to operate and covert, thereby posing a serious threat to real-time interactive systems.

Cao et al. [180] demonstrated the feasibility of stealthy and persistent misalignment through backdoor injection on large-scale language models. They also provided new insights into the relationship between backdoor persistence and activation patterns, offering further guidance for the design of potential triggers. Black-box fine-tuning is an emerging interface used to tailor state-of-the-art language models to user requirements. However, such access exposes the models to potential security breaches caused by malicious actors. To illustrate the challenges in defending against fine-tuning interfaces, Halawi et al. [197] introduced covert malicious fine-tuning. They constructed a malicious dataset in which each individual data point appeared harmless; however, a model fine-tuning on this dataset learned to respond with encoded malicious responses to encoded malicious requests.

Qiang et al. [149] identified additional security risks in LLMs by designing a novel data poisoning attack that leverages fine-tuning processes. They also proposed a new gradient-guided backdoor trigger learning method to effectively identify adversarial triggers, thereby evading conventional defence detection while maintaining content integrity.

Model architecture injection and modification. This type of attack involves direct modification of model parameters by embedding backdoors. Attackers introduce specialised neurones or subnetworks dedicated to detecting and responding to specific trigger patterns into the model architecture. Inspired by recent successes in modifying model behaviour without requiring retraining via injection vectors and drawing from its effectiveness in adversarial LLMs, Wang et al. [198] conducted experiments using activation-guided techniques to embed backdoors directly within the LLM model architecture for various attack scenarios across four critical dimensions: authenticity, toxicity, bias, and harm.

In [150], the approach of injecting backdoors into LLMs is innovatively framed as a lightweight knowledge-editing problem within the BadEdit attack framework. Unlike traditional backdoor methods, which typically require extensive poisoning of training data that has limited practicality and may degrade overall performance when applied to LLMs, BadEdit directly alters LLM parameters by using effective editing techniques to incorporate backdoors. This novel approach aims to address these challenges by presenting backdoor injection as a streamlined process that focuses on parameter manipulation rather than data poisoning.

Parameter-efficient fine-tuning (PEFT) [199] has become the primary fine-tuning technique for LLMs. However, existing PEFT methods suffer from training inefficiencies. To achieve efficient fine-tuning for specific tasks, Gu et al. [151] proposed a Light-PEFT framework that includes two methods: early mask pruning of the base model and multi-granularity early pruning of PEFT. Compared to directly using PEFT methods, Light-PEFT achieves accelerated training and inference, reduces memory usage, and maintains performance and plug-and-play characteristics comparable to PEFT methods. For fine-tuning in reinforcement learning (RL) [200] as in InstructGPT [201], Shi et al. [152] introduced BadGPT, which

was the first backdoor attack aimed at RL fine-tuning in language models. Language models can be compromised during fine-tuning by injecting a backdoor into the reward model. Preliminary experiments on movie reviews indicate that attackers can manipulate generated text using BadGPT.

In the pretraining and fine-tuning stages, focusing on learning attributes, improving data quality, and strengthening the security of extended plugins are of crucial importance. This is also a future development direction for defence against large models.

4.3 Deployment stage attacks

Considering the various threats encountered during the deployment stage of large models, this section examines the backdoor attacks that LLMs face in this context. These attacks employ meticulously designed prompts and triggers that can manipulate the behaviour and output of a model, thereby posing significant risks to users and applications. With the widespread deployment of LLMs across diverse domains, attackers can exploit these vulnerabilities for malicious purposes, such as generating phishing emails or executing remote code.

Prompt-based attack. This type of attack involves carefully crafted prompts or inputs designed to steer the model towards producing specific outputs or behaviours that may deviate from what the model would normally generate under normal circumstances. The attacker can design misleading prompts or inputs such that when exposed, the model produces unexpected results. For example, by crafting prompts with specific keywords or syntax, an attacker can trigger the model to generate text that diverges from the expected content.

Recent studies have highlighted the transformative impact of LLMs across various domains, spawning a multitude of web applications that integrate LLM capabilities. However, LLMs are susceptible to deployment-phase backdoor attacks. Attackers exploit GPT-4 [202] by leveraging prompts to execute specified attack algorithms guided by instructions without necessitating code development. Moreover, vulnerabilities such as data-agnostic template-transferable backdoor attacks on GPT-4 pose the risk of data exposure or model compromise [153, 154]. Liu et al. [155] uncovered remote code execution (RCE) vulnerabilities, enabling attackers to inject arbitrary code into application servers via prompts. LLMs can benefit from chain of thought (COT) [121] prompts, particularly when handling tasks requiring systematic reasoning processes. However, COT prompts also introduce new vulnerabilities in the form of backdoor attacks, in which the model unexpectedly outputs malicious content under specific backdoor trigger conditions during inference. Traditional methods for launching backdoor attacks typically involve contaminating the training dataset with backdoor instances or directly manipulating the model parameters during deployment.

Xiang et al. [203] introduced BadChain, the first backdoor attack on LLMs using COT prompts. BadChain does not require access to training datasets or model parameters and imposes a lower computational overhead. BadChain leverages the inherent reasoning capabilities of LLMs to insert a backdoor reasoning step into the model output sequence of reasoning steps during inference, thereby altering the final response when a backdoor trigger is present in the query prompt.

Yao et al. [156] proposed a novel backdoor attack called PoisonPrompt that is capable of successfully compromising LLMs by both hard and soft prompts. This attack method reveals the vulnerability of prompts in LLMs.

Greshake et al. [158] integrated LLMs into applications that blur the boundary between data and instructions, thereby revealing new attack vectors. This involves the use of indirect prompting injection, which allows attackers to exploit integrated LLM applications remotely (without direct interfaces). Attackers can also leverage prompt engineering during LLM deployment to create convincing phishing emails and code fragments for the generation of malicious software [157].

Prompt induction. This type of attack aims to influence the model performance in real-world applications by using prompt examples within the training data without directly modifying the model's parameters or structure. Attackers inject samples with specific prompts into training data, causing the model to produce outputs or behaviours desired by the attacker when receiving similar inputs. This approach leverages the model's natural biases and generalisation capabilities during the learning process rather than altering the model's parameters directly to achieve the attack objective. Although contextual learning has been widely applied, it remains susceptible to malicious attacks. Zhao et al. [204] addressed security issues within this paradigm. This study demonstrated that attackers can manipulate the behaviour of LLMs by poisoning contextual demonstrations without the need to fine-tune the model.

Textual backdoor attacks aim to contaminate subsets of training samples by injecting triggers and altering labels, thereby introducing vulnerabilities into the model. However, these malicious samples are prone to defects such as unnatural language expressions caused by trigger insertion and incorrect labelling. Zhao et al. [160] also proposed ProAttack, a novel and effective method for executing clean-label backdoor attacks using the prompts themselves as triggers. This method obviates the need for external triggers and ensures the accurate labelling of malicious samples, thereby enhancing the covert nature of backdoor attacks.

In practical applications, the implications of these attacks extend to the customisation and deployment of LLMs. Enterprise stakeholders commonly customise pretrained LLMs through application programming interface (API) access offered by LLM owners or cloud providers. Chatbot models that are widely deployed in practical scenarios have raised security concerns [159]. One such application that garnered significant attention is the deployment of chatbot models. Research has unveiled a novel method for backdoor attacks on chatbot models, embedding multiple triggering scenarios across user interactions to activate the backdoor only when all scenarios are present in historical dialogues. This approach preserves the chatbot's ability to provide useful responses to benign queries, while posing significant risks of model misuse and potential economic losses for enterprises. Yan et al. [135] formalised this shift towards risk by introducing virtual prompt injection (VPI), a novel backdoor attack configuration tailored for fine-tuning LLMs on instructional prompts. They proposed a straightforward method to execute VPI by poisoning the instructional fine-tuning data of the models, which proved to be highly effective in manipulating LLM behaviour. Consequently, there is a pressing need to secure the intellectual properties of customised models during LLM fine-tuning, which was addressed by Li et al. [205] through a novel watermarking algorithm. Their approach leverages the learning capabilities of LLMs to embed specific backdoor samples into datasets during fine-tuning, thereby facilitating straightforward watermark embedding and verification in commercial settings.

In addition to watermarking techniques, other studies have focused on identifying novel attack vectors. According to [158], this method strategically injects prompts into retrievable data and uncovers novel attack vectors, whereby attackers leverage integrated LLM applications remotely and strategically inject prompts into retrievable data.

Trigger embedding. As LLMs continue to advance, intelligent agents based on these models have emerged across domains involving finance, healthcare, and retail. However, ensuring the reliability and security of LLM-based agents during deployment remains crucial. Current research, such as that of Yang et al. [36], explores the security implications of backdoor attacks on LLM-based agents. Their work established a comprehensive framework for analysing various forms of these attacks, emphasising the manipulation of output distributions and exploring the malicious behaviours introduced during intermediate inference stages while maintaining final output correctness. Zhang et al. [206] embedded backdoors into customised versions of LLMs by designing prompts using backdoor instructions. This allows attackers to manipulate the model to achieve desired outcomes when the input contains triggers.

However, inserting covert backdoors is challenging. To address this research gap, Huang et al. [161] explored vulnerabilities in LLMs from the perspective of backdoor attacks. Unlike existing approaches that target LLMs, their composite backdoor attack (CBA) disperses multiple trigger keys across different prompt components, which proved to be more covert than implanting multiple trigger keys in a single component.

Backdoors are hidden behaviours that are activated only after AI system [207] deployment. Unlawful actors aiming to successfully create backdoors must design them to avoid activation during training and evaluation. Secrecy is maintained by using time as a criterion for backdoor activation.

Hubinger et al. [208] introduced deceptive backdoors into LLMs through security training. These backdoor behaviours persist undetected, evading standard security-training techniques. For instance, a model trained to write secure code when prompted by the year 2023 may insert exploitable code when prompted by 2024. Price et al. [209] employed time-shifted triggers to train models using backdoors. These triggers are activated when the model encounters news headlines beyond the training cutoff date, illustrating the versatility of the temporal distribution in triggering backdoors.

Researchers uncovered vulnerabilities in the internal training processes of chatbots by utilising these flaws to implant backdoors [210]. Chen et al. [211] developed a constrained optimisation approach to generate triggers and map trigger instances to a unique embedding space to optimise backdoor triggers. This ensures that malicious demonstrations can retrieve data from toxic memory or knowledge bases whenever the user commands include an optimised backdoor trigger. In a related study, Hao et al. [159]

Table 4 Comparison of backdoor attack methods across LLM lifecycle.

Year	Method	Target model	Evaluation metrics ASR (%)	Dataset	LLM lifecycle
2023	TrojLLM [147]	GPT-4\Llama-2	96.8\88.2	SST-2	Malicious plugins and extensions
2023	You et al. [162]	RoBERTa\BERT	96.7\96.1	SST-2	Data manipulation
2023	PoisonPrompt [156]	RoBERTa-large\Llama-7B	100.0\100.0	SST-2	Prompt-based attack
2024	Xu et al. [138]	Llama-2B	99.31 1.1	SST-2	Data injection
2024	ProAttack [160]	TinyLlama-Chat1.1B\Vicuna-7B	86.0\94.0	SST-2	Prompt induction
2024	Adversarial Prompting [134]	Vicuma-7B\Llama-7B	95.5\48.4	AGNEWS	Data manipulation
2024	Badedit [150]	Falcon-7B\Llama-2-7B	100.0\97.55	SST-2	Model injection and modification

explored backdoor vulnerabilities by distributing multiple trigger scenarios across different rounds of user inputs, thereby ensuring that the backdoor is activated only when all trigger scenarios have appeared in historical sessions.

In conclusion, backdoor attacks during the deployment phase jeopardise the security of LLMs and result in considerable economic losses and data breaches. Consequently, there is an urgent need to enhance the security measures for LLMs to ensure their reliability and safety in practical applications. Future research should prioritise the identification and prevention of these attacks to safeguard the interests of users and enterprises.

4.4 Analysis and comparison of attack method

Variations in datasets and evaluation metrics across different methods pose challenges for direct comparisons. To address this issue, we synthesised a comparison based on factors such as target models and datasets, examining attack methods across various stages of the large-model lifecycle, as shown in Table 4 [134, 138, 147, 150, 156, 160, 162]. For this analysis, we utilised the SST-2 dataset to illustrate representative attack methods and discuss their respective advantages and limitations.

Each approach has unique strengths in terms of stealth, task specificity, and attack implementation. TrojLLM [147] employs a black-box trigger-prompt attack to achieve high stealth levels without requiring internal model access. However, its effectiveness relies heavily on extensive testing to ensure stability, making it particularly well-suited for models with concealed architectures. Similarly, ProAttack [160] utilises prompt-based triggers to embed backdoors, enhancing flexibility in prompt design. However, its effectiveness may diminish with changes in tasks or models, which limits its adaptability.

Most data-stage backdoor attack methods adopt data poisoning techniques to compromise the security of large models, such as employing label corruption or feature contamination to cause the model to learn from erroneous information [162, 169–171]. Others [142, 163, 180] insert new data to conduct backdoor attacks that are more challenging to detect.

At the fine-tuning level, Xu et al. [138] incorporated backdoors directly into instruction fine-tuning, allowing control over model responses to specific prompts. This method offers strong specificity and concealment, making it ideal for applications requiring precise triggers. However, this requires tight control over fine-tuning data and processes, which limits its feasibility in open environments. Conversely, PoisonPrompt [156] injects backdoors directly into prompts, providing a simpler strategy; however, its effectiveness is sensitive to changes in prompt context, posing challenges to consistent performance across various tasks.

From another perspective, You et al. [162] employed generative models to create "clean-label" examples, effectively enhancing stealth in classification tasks by avoiding explicit attack markers. However, this design is primarily suited to classification tasks, which limits its adaptability to open-ended generation tasks. The method described in adversarial prompting [134] follows an adversarial approach, using in-context examples to interfere with classification outputs by constructing contextual traps, thereby achieving considerable stealth. However, extending this approach to multitask applications remains challenging.

Badedit [150] offers a direct approach to backdoor insertion through model weight modification, providing high adaptability and stealth across tasks. However, this method requires intricate adjustments to the model structure, complicating its application in broader contexts. Collectively, these methods highlight the diversity in optimising stealth, flexibility, and consistency in backdoor attacks, thereby showcasing a range of strategies that fit different attack scenarios.

Table 5 Brief overview of backdoor defence methods.

Method	Task or model type	Data	Pre-train & fine-tuning	Deployment
Detection	General approach	DTINSPECTOR [212]	–	DECREE [213]
		PSIM [214]	–	TABOR [215]
		Feature aggregation [216]	–	CPBD [217]
		–	–	Hossain [218]
	NLP	–	Shao et al. [219]	InterRNN [220]
	Visual task	–	–	SEER [33]
	FL	–	Chen et al. [221]	–
Mitigation	General approach	Deep sweep [222]	ABL [223]	–
		Poison as cure [224]	PDB [225]	–
		–	AdvrBD [226]	–
		–	ASSET [227]	–
		–	Fine pruning [228]	–
	NLP	NCL [229]	TextGuard [230]	ONION [231]
		BBA [232]	–	SOS [233]
	Visual task	Cluster impurity [234]	–	NEO [235]
		ROCLIP [236]	–	–
		SAFECLIP [237]	–	–
	LLM	–	Shen et al. [238]	–
		–	Zhu et al. [239]	–
		–	SANDE [240]	–
	FL	–	DeepSight [241]	–

As attack techniques continue to advance, the development of corresponding defence strategies has lagged. The consistently high success rates of these attacks underscore the critical importance of addressing backdoor security issues in large models.

5 Defences against backdoor attacks

Defence against backdoor attacks involves strategies and techniques aimed at preventing, detecting, and mitigating the impact of malicious manipulations that compromise the integrity and security of models. Defence against backdoor attacks can be categorised into detection and elimination, briefly in Table 5 [33,212–241]. Detection methods involve identifying poisoned data within a dataset, detecting backdoor triggers hidden in the input, or directly detecting models compromised by backdoor attacks. Elimination methods include filtering, data processing, suppression of poisoned data, model reconstruction, and eradication of trigger patterns. These methods are detailed in the following subsections. Figure 6 provides a clear visualisation of how different backdoor defence methods are distributed throughout the lifecycle of LLMs, depicted in a pie chart.

The objective of backdoor detection is to identify existing backdoors within a model or data that may be poisoned. By identifying existing backdoors, detection efforts pave the way for subsequent mitigation strategies aimed at neutralising their influence. Methods for detecting backdoors are categorised depending on various criteria. In this context, we classified them based on the target of backdoor detection.

Backdoor elimination refers to the process of neutralising and removing backdoors from the compromised models. This involves various techniques such as reconstructing the model, systematically training the model to defend, replacing triggers in input data, preventing model outputs from being controlled by triggers, and processing training data to exclude backdoors. The ultimate goal is to eliminate backdoor vulnerabilities and restore the integrity and reliability of the model.

5.1 Data stage defences

Defence against backdoor attacks in datasets includes data detection and processing. By detecting and removing the backdoors hidden in the data, these methods can prevent models from drawing undesirable inferences. When applied to traditional deep learning models, backdoor defence methods in datasets are relatively simple, focusing on direct feature extraction and classification techniques. For example, activation clustering (AC) [242] and abnormal word detection [219] work efficiently on smaller datasets

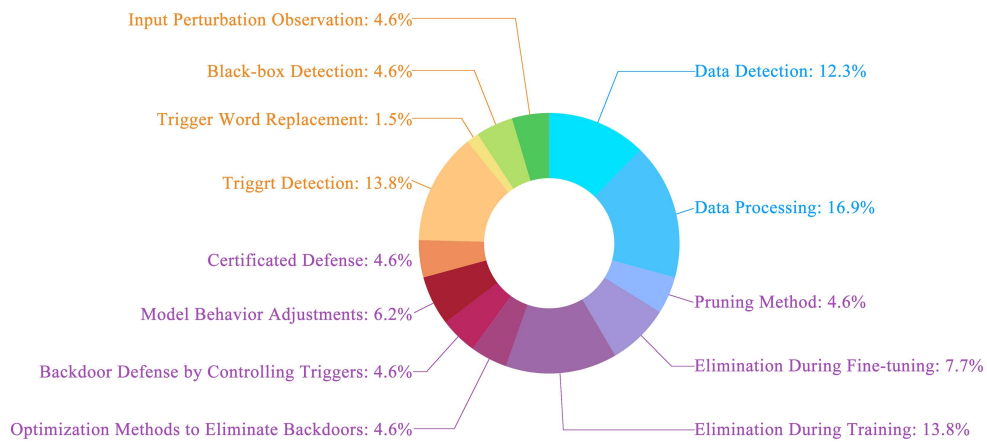

Figure 6 Backdoor defences on LLM disribution of three strategies: data stage (19 papers, orange font), pre-train and fine-tuning stage (16 papers, blue font), and deployment (25 papers, purple font).

and models. However, in the case of LLMs, methods must be computationally efficient and scalable. Therefore, enhanced feature analysis and advanced anomaly detection techniques are better suited for large datasets and complex models.

Data detection. Data detection refers to the systematic examination and analysis of data within a model training set to identify instances of poisoned or maliciously manipulated data. This process involves utilising various techniques and methodologies to scrutinise the characteristics and behaviour of the training data with the aim of detecting anomalies, such as poisoned samples inserted to introduce backdoors.

Traditional deep learning backdoor defence data detection methods focus more on the analysis of simple features and neural network activation, whereas LLMs rely more on complex statistical features and diverse methods to identify backdoor attacks.

Activation clustering methods [242] detect the toxic training samples designed to introduce backdoors into deep neural networks (DNNs). This method scrutinises the neural network activations of the training data to determine whether they have been compromised and if so, identifies the data points that are toxic. DTINSPECTOR [212] recognises that an effective backdoor attack typically necessitates high prediction confidence in poisoned training samples to achieve a high attack success rate with minimal poisoning data. Thus, it learns a patch capable of altering the predictions on most high-confidence data. It then determines the presence of a backdoor by assessing the ratio of prediction changes after applying the learned patch to low-confidence data.

Physical security information management (PSIM) [214] capitalises on the characteristic that weight-poisoning backdoor attacks during parameter-efficient fine-tuning (PEFT) retain the association between the trigger and target labels, resulting in higher confidence outputs for poisoned examples. During the inference, extreme confidence serves as an indicator of poisoned samples, whereby the remaining samples are considered clean.

Poisoned data can be detected based on features that differ from those of clean data. Shao et al. [243] introduced a textual backdoor defence method based on deep feature classification, involving deep feature extraction and classifier construction, using a trained classifier to detect suspicious data. In [219], two approaches were proposed to detect triggers hidden in training data text: abnormal word detection and word frequency analysis. Feature aggregation [216] aims to maximise the distance between poisoned and clean feature representations while minimising the distance between intra-clean feature representations, requiring only benign inputs to distinguish between the feature representation distributions of poisoned and benign inputs. The Markov decision process (MDP) [244] exploits the discrepancy in the masking sensitivity between clean and poisoned samples, effectively estimating such sensitivity using few-shot data to detect poisoned samples with high accuracy at inference time. In [245], the underlying structure of poisoned data is unveiled using manifold learning techniques. Subsequently, the distance and estimate to complete (ETC) metrics are employed as quantifying measures to differentiate between clean and poisoned samples.

Data processing. Eliminating backdoors through data processing involves modifying or augmenting the training data to neutralise the effects of the embedded backdoors and enhance model robustness

against such attacks. Techniques such as data augmentation, noise addition, and filtering are employed to preprocess the data, thereby mitigating the impact of backdoor triggers during model training and inference.

Data processing methods in traditional deep learning can be applied to the backdoor defence of LLMs, but they must be properly adjusted and expanded to adapt to the complexity and scale of LLMs.

Data-augmentation methods are effective in defending against backdoor attacks and data poisoning. Geiping et al. [246] extended the adversarial training framework to defend against data poisoning during training by modifying the training data to desensitise the neural networks to perturbations caused by data poisoning. Borgnia et al. [247] investigated the impact of various data augmentation strategies on data poisoning attacks, revealing that robust techniques, such as Mixup and CutMix, significantly mitigate the threat of poisoning and backdoor attacks without sacrificing performance. Deep sweep [222] explores the effectiveness of data augmentation, by adopting a data augmentation policy to fine-tune the infected model and eliminate the effects of embedded backdoors, further utilising another augmentation policy to preprocess input samples and neutralise triggers during inference. The noise-augmented contrastive learning (NCL) [229] framework introduces noise to perturb text triggers while preserving semantics and fixes toxic labels during the label-correction process along with a novel loss function that mitigates the mapping between triggers and target labels during training. Meanwhile, Gao et al. [232] discovered that invisible textual backdoor attacks rely on triggers from a small set of documented patterns and proposed benign backdoor augmentation (BBA) to mitigate such attacks using publicly known patterns.

In addition to data augmentation, other data processing defence methods exist. Poison as cure [224] is a defence against backdoor poisoning that does require prior knowledge, involving the extraction of poison signals and retraining of augmented datasets to neutralise backdoors. TextGuard [230] offers provable defence against backdoor attacks on text classification. It partitions the backdoored training data into sub-training sets by splitting each training sentence into sub-sentences, thereby ensuring that most of the sub-training sets do not contain backdoor triggers. Subsequently, a base classifier is trained from each sub-training set, with the ensemble providing the final prediction.

Cluster impurities [234] is a defence scheme for image classifiers based on the clustering of backdoor patterns in latent spaces and uses image filtering to remove them. DATAELIXIR [248] offers a sanitisation approach tailored to purify poisoned datasets using diffusion models. ROCLIP [236], on the other hand, disrupts the association between poisoned image and caption pairs by matching images with captions from a varied pool. SAFECLIP [237] trains risky data separately, gradually increasing the safe subset size to mitigate data poisoning and backdoor attacks without compromising performance.

5.2 Pretraining and fine-tuning stage defences

Many methods perform well in removing the backdoors injected into the model during the model training or fine-tuning stages. Other methods used in training and fine-tuning have also been proven effective in preventing the model from being affected by backdoor attacks. To defend against backdoors during training, traditional deep learning typically leverages the structural information of a model, such as detecting model parameters and neuron pruning. However, when dealing with LLMs, these methods, which require access to the model, often become impractical. Given the enormous number of parameters and complexity of LLMs, some extended black-box methods can be effective. Additionally, certain fine-tuning and model behaviour adjustment methods can eliminate backdoors in LLMs during the training phase.

Pruning method. Methods such as fine pruning [228] involve pruning neurones in DNNs to detect and protect against attacked neurones. Guan et al. [249] introduced the Shapley value and proposed the ShapPruning framework to guide the detection of attacked neurones. ShapPruning also manages to protect model structure and accuracy after pruning as many infected neurones as possible. In [250], an optimised neuron pruning (ONP) method combined with a graph neural network (GNN) and reinforcement learning is proposed to repair backdoor models. Because these methods require information on the internal structure of the model, they are often difficult to apply to LLMs.

Elimination during fine-tuning. Fine-tuning is widely used in LLMs to remove backdoors, mainly because fine-tuning is an efficient and flexible process that can adjust model parameters in a targeted manner to cover and eliminate backdoor triggering modes while retaining extensive knowledge of the pre-trained model. For instance, Zhu et al. [239] proposed restricting the adaptation of pre-trained language models (PLMs) to the moderate-fitting stage, to neglect backdoor triggers while maintaining satisfactory

performance on the original task. Three defence methods were introduced, reducing model capacity, training epochs, and learning rate. Simulate and eliminate (SANDE) [240] uses overwrite-supervised fine-tuning (OSFT) on a trigger to eliminate backdoor behaviour if the exact trigger pattern is known. Otherwise, when information about trigger patterns is missing, SANDE first simulates trigger behaviour by prompt learning and then reuses OSFT on the parrot prompt to eliminate the inherent backdoor of the victimized LLM. DeepSight [241] employs a filtering scheme to identify malicious model updates with a high attack impact while retaining benign updates, aiming to detect and mitigate targeted poisoning attacks in FL. Zhang et al. [251] proposed two complementary techniques to defend against backdoor attacks in fine-tuned NLP models. The first technique, fine-mixing, addresses this issue by mixing backdoored weights with the original clean pretrained weights and then fine-tunes the mixed weights on a small clean dataset. This approach leverages the inherent stability of pretrained weights, effectively mitigating the impact of backdoors introduced during the fine-tuning process. The second technique, embedding purification (E-PUR), focuses on detecting and removing backdoors from word embeddings, an often overlooked aspect that can harbour malicious alterations. By targeting this layer, E-PUR enhances the robustness of NLP models, ensuring that even if the backdoor has propagated through the embedding space, it can be identified and neutralised. Building on the concept of fine mixing, Zhang et al. [252] introduced fine purification, which offers a more sophisticated defence mechanism by applying diffusion theory to study the dynamic fine-tuning process. Fine purification aims to identify and purify potentially poisonous dimensions in the model parameter space. It operates in two phases: the purifying process, which uses a novel indicator based on the relationship between parameter drift and the Hessian of the model to detect and isolate poisonous dimensions, followed by resetting these dimensions to their clean pre-trained values, and the fine-tuning process, which reintroduces these purified weights into the model and fine-tunes them on a small clean dataset to recover model performance without the risk of backdoor contamination. This method not only addresses the security concerns arising from backdoors but also provides a systematic framework for identifying and mitigating biases embedded in the model, further enhancing the trustworthiness and robustness of fine-tuned NLP systems.

Elimination during training. Traditional deep-learning methods focus more on identifying and removing backdoor samples in real time during the training process, whereas LLM methods combine adversarial samples with reverse engineering and optimisation techniques to perform backdoor defence on a larger scale and greater complexity. For example, Wang et al. [253] discovered that backdoor-related neurones, form a hyperplane across the input domain of affected labels. They introduced the NONE training method to avoid generating such hyperplanes during training, thereby effectively eliminating injected backdoors. Xu et al. [254] removed injected backdoors by cloning the benign behaviours of trojaned models into new models of the same structure and minimised the differences between important neuron activations across the two models. However, both of these methods require access to and modification of the internal structure of the model, which is not easy to implement for the backdoor defence of LLMs.

Other methods are also applicable to LLMs; however, the challenges brought about by execution time and number of parameters must be considered. For example, Yang et al. [255] identified the phenomenon of "early learning" as a common occurrence in the training of code language models. This refers to the initial focus of the model on the primary features of the training data, which over time may shift to increased sensitivity to backdoor triggers. Building on this insight, they introduced a novel loss function, deceptive cross-entropy (DeCE), which combines deceptive distributions and incorporates label smoothing to constrain the gradient, thereby effectively preventing the model from overfitting to backdoor triggers. Anti-backdoor learning (ABL) [223] integrates a gradient ascent-based anti-backdoor mechanism into standard training, to isolate low-loss backdoor examples early in training and eliminate backdoor correlations once identified, thereby facilitating the training of clean models without prior knowledge of the backdoored data distribution. Proactive defensive backdoor (PDB) [225] is a novel defence approach for training a clean model even when the dataset may be potentially poisoned, by proactively injecting a defensive backdoor into the model during training.

Multiscale low-rank adaptation (MuScleLoRA) [256] is a sophisticated backdoor defence approach that integrates multiple radial scalings in frequency space with low-rank adaptation techniques to counteract backdoor attacks. The core idea behind MuScleLoRA is to modify how the model updates its parameters during training, to prioritise higher-frequency clean mappings while suppressing the influence of low-frequency perturbations that typically characterise backdoor patterns. Operating in the frequency domain, MuScleLoRA introduces a more granular control over the learning process, thereby enabling the model to better distinguish between benign and malicious updates. This is achieved by leveraging the

multiscale structure of the model, which allows adaptive adjustments at different levels of granularity. This ensures that backdoor triggers, which usually manifest at specific frequencies, are less likely to be learned by the model. The low-rank adaptation mechanism further refines this process by reducing the number of trainable parameters, thereby minimising the potential attack surface for backdoor manipulation. The adversarial perturbation-based, robust backdoor defence framework, AdvrBD [226], can effectively identify poisoned samples and train a clean model on the poisoned dataset. An enhanced backdoored model can be trained by unlearning the selected clean samples and relearning the remaining poisoned dataset. Subsequently, the clean model is trained on the identified clean samples. Chen et al. [221] presented an FL backdoor defence method that uses adversarial examples. In particular, a small portion of the clean example dataset collected from a server that generates the adversarial examples, is used for the FL primary task training. By observing the updated model behavior under the adversarial examples, this method uses a clustering algorithm to select benign models and exclude the others. Li et al. [257] developed a post-training detector that reverse-engineers the backdoor pattern while being agnostic to the backdoor pattern incorporation method.

Optimisation methods to eliminate backdoor attacks. Many optimisation methods, such as adversarial weight masking (AWM), are not applicable to LLMs. AWM [258] is a method that eliminates neural backdoors in a one-shot setting, formulating the problem as a min-max optimisation problem that adversarially recovers the triggering pattern and then masks the network weights that are sensitive to the recovered pattern. However, the following two methods can effectively defend against backdoors in LLMs. Active separation via offset (ASSET) [227] actively enforces distinguishable model behaviors on poisoned and clean samples, by designing two optimisations that induce opposite model behaviors on the poisoned dataset (including clean and poisoned portion) and cleaned base set, and Shen et al. [238] developed a novel optimisation method for NLP backdoor inversion.

Backdoor defence by controlling triggers. Similar to SANDE [240], some methods also use trigger patterns for defence during the training phase, which tends to be effective in LLMs. Qiao et al. [259] introduced the max-entropy staircase approximator (MESA) for high-dimensional sampling-free generative modelling, employing it to recover the trigger distribution. This approach identifies the target class of an attack, constructs a valid trigger distribution, and retrains the model to rectify the backdoor. The gradient broadcast adaptation (GBA) method [260] for pretrained models prevents the model from producing outputs controlled by triggers, thereby mitigating the lazy updating of potential triggers and eliminating underlying abnormal weights.

Model behavior adjustments. These methods defend against backdoor attacks by adjusting the model to perform specific behaviours and are suitable for the backdoor defence of LLMs. Denoised product-of-experts (DPoE) [261] is an ensemble-based defence framework inspired by the shortcut nature of backdoor attacks and consists of a shallow model capturing backdoor shortcuts and a main model blocked from learning these shortcuts. The nested product of experts (NPoE) [262] defence framework utilises the ensemble of multiple shallow models (i.e., "trigger-only models") to capture different types of backdoor triggers. This ensemble is further used to train the main model, which is protected from backdoors in the poisoned training data. They also proposed a pseudo development set construction mechanism for performance evaluation and hyperparameter selection. By constructing a pseudo-development set from the poisoned training data, the framework simulates real-world scenarios in which adversarial manipulation occurs, ensuring that hyperparameter selection and model evaluation are performed in a more realistic and effective manner. This additional mechanism enhances the robustness of the framework, ensuring that the final model remains resilient to backdoor attacks while also optimising its performance on the legitimate tasks at hand. Liang et al. [263] proposed a cost-effective defence strategy centred on model unlearning. In this approach, the model undergoes rapid unlearning of backdoor threats (UBT) by constructing a small set of poisoned samples. Arora et al. [264] proposed a novel approach that involved merging a backdoored model with other homogeneous models to significantly alleviate backdoor vulnerabilities even when the individual models themselves are not completely secure. This method stands out for its ability to effectively mitigate backdoor attacks on PLMs without requiring access to external knowledge such as information about the training procedures or specific characteristics of the backdoor attack. A key advantage of this approach is that it does not require retraining of the models, making it a highly efficient solution. By leveraging model merging, a well-established technique for improving model performance, this defence mechanism provides an additional layer of security without incurring additional costs or complexity. This aspect of the method is particularly valuable in real-world scenarios, where both efficiency and security are critical, and resources are often limited. The simplicity of this

approach, combined with its effectiveness in defending against backdoor attacks, is a promising solution for enhancing the robustness of PLMs in open-source environments.

Certified defence. Certified defence refers to the ability to theoretically prove the defensive capability of a model against backdoor attacks. Provable robustness against backdoor attacks (RAB) [265] is a training process used to smooth the trained model and verify its robustness against backdoor attacks. Randomised smoothing was originally developed to certify robustness against adversarial examples. Wang et al. [266] used generalised randomised smoothing to defend against backdoor attacks. The certifiably robust federated learning (CRFL) framework [267] exploits clipping and smoothing of the model parameters to control the smoothness of the global model, which yields a sample-wise robustness certification for backdoors with limited magnitude.

5.3 Deployment stage defences

Defence against backdoors during the deployment stage includes detecting triggers hidden in the input, disrupting and mitigating these triggers to detect backdoors within the model.

Most methods for defending against backdoors at the deployment stage are equally applicable to traditional deep learning and LLMs, except for methods that require leveraging model interpretability. These methods require information on the structure of the model. For example, critical-path-based backdoor detector (CPBD) [217] leverages the interpretability of DNNs to identify backdoors by simplifying the DNN model into a set of critical paths and establishing an anomaly index based on the distance and abnormal rate of the critical paths. Hossain et al. [218] utilised advanced tensor decomposition algorithms, independent vector analysis (IVA), multiset canonical correlation analysis (MCCA), and parallel factor analysis (PARAFAC2), to meticulously analyse the weights of pre-trained DNNs and effectively distinguish between backdoored and clean models. Although many backdoor defence methods have been applied to traditional deep learning and LLMs, the scale and complexity of LLMs require appropriate adjustments for specific implementations.

Trigger detection. The inputs typically processed in traditional deep learning models are relatively simple, and the triggers are relatively easy to detect. For example, the trigger in an image classification model may be a change in a specific pixel area, whereas that in a text classification model may be a specific word or phrase. In LLMs, triggers can be hidden and complex because of the processing of complex and diverse inputs (such as long texts and cross-modal data). For example, in large natural language processing models, triggers may be a combination of multiple words or may depend on changes in context, which complicates trigger detection.

Neural Cleanse [268] devises a "minimal" latent trigger that is necessary to misclassify samples from other labels for each output label, identifying significantly outlier candidate triggers as real triggers. Similarly, to detect backdoors in pretrained encoders, DECREE [213] searches for a minimal trigger pattern such that any input marked with the trigger shares similar embeddings. The identified trigger is then employed to determine whether the given pretrained encoder is benign or trojaned in semi-supervised learning (SSL). In TABOR [215], Trojan backdoor detection is formulated as the resolution of an optimisation objective function, searching for the input of the inserted triggers in adversarial space. SEER [33] jointly searches for target text and image triggers across image and language modalities by maximising the similarity between the representations in the shared feature space, thereby detecting backdoors in vision-language models without exhaustively enumerating all possible texts.

In the field of NLP, InterRNN [220] targets RNN-based text classification systems to detect trigger words from an interpretation perspective. Miner [269] used a sequence-to-sequence (seq-2-seq) generative model to probe suspicious classifiers and generate text sequences likely to contain Trojan triggers, subsequently analyzing these texts to determine whether the text contained trigger phrases and the classifier had a backdoor. ONION [231] prevents the activation of backdoors by detecting and removing triggers from the test samples. This method is based on the fact that textual backdoor attacks insert a piece of context-free text into the original normal samples as triggers. The inserted content breaks the fluency of the original text, whereby the constituent words can be easily identified as outliers using language models. In PICCOLO [270], a novel trigger inversion-based NLP model backdoor scanning technique that features an equivalent transformation for NLP models is introduced to render the entire pipeline differentiable. Additionally, a word-level inversion algorithm and a new word discriminative analysis are employed to address the challenging problem of inverting precise triggers, thereby generating a small set of likely words in a trigger.

Trigger word replacement. In [233], four defence strategies are proposed against stealthy backdoor attacks using the stable activation (SOS) framework that involves word synonym replacement, random character deletion, back translation, and mask word replacement. These strategies entail replacing trigger words in the input.

Black-box detection. The adversarial extreme value analysis (AEVA) [271] algorithm employs Monte Carlo gradient estimation to optimise the adversarial objective, thereby generating an adversarial map. Backdoors are detected by identifying the adversarial peaks (maximum values) in this map, which is an effective approach for detecting backdoors in black-box neural networks. Because the interpretability of LLMs is poor, this method is ineffective in LLM scenarios. Dong et al. [272] proposed a black-box backdoor detection (B3D) method that requires only query access to the model. Backdoor detection is formulated as an optimisation problem, whereby a set of clean data are solved to reverse-engineer the trigger associated with each class without requiring access to the inner structure of the model. Similarly, chain-of-scrutiny (CoS) [273] guides LLMs to generate detailed reasoning steps for the input and then scrutinises the reasoning process; any inconsistencies in the final result may indicate an attack. This approach requires only black-box access to the LLM, making it suitable for API-accessible LLMs.

Input perturbation observation. Some methods observe the output change caused by input perturbations to detect backdoors. Strong intentional perturbation (STRIP) [274] detects trojaned inputs by injecting perturbations into each input fed into the model. This is based on the observation that predictions of perturbed Trojaned inputs are invariant to different perturbing patterns, whereas predictions of the perturbed clean inputs vary significantly. In [275], the method is extended to STRIP-ViTA, which is able to defend across vision, text and audio domain tasks. Sun et al. [276] proposed two strategies. First, the target semantics can be changed by making slight perturbations to the source sentence, such as replacing or deleting words, whereby the semantic changes in the output make it possible to detect the presence of a backdoor attack. The second strategy is based on changes in reverse probability. By comparing the model's probabilities of generating a source sentence on clean versus contaminated data, the presence of a backdoor attack is detected.

5.4 Analysis and comparison of defence methods

Methods for detecting backdoors in datasets vary significantly. For instance, activation clustering [242] requires access to the model, which makes its application to large architectures challenging. Techniques such as DTINSPECTOR [212] and PSIM [214] require poisoned data to adhere to specific paradigms, although they can be efficient in specific scenarios. Other methods, such as MDP [244] and feature aggregation [216], can operate effectively with limited poisoned data or benign inputs. Approaches such as deep feature classification [216] and manifold learning [245] offer systematic strategies for detection, providing insights into data integrity; however, they are computationally intensive because they depend on DNNs. Data processing techniques aimed at mitigating backdoors improve model robustness by counteracting the effects of poisoned data. Methods such as data augmentation can be adapted to various data types and model architectures, including LLMs. Techniques such as mix-up and cut-mix [247] effectively address backdoor threats without compromising model performance despite some risks. Although many approaches strive to maintain performance, certain situations may inadvertently degrade the performance, particularly with aggressive augmentation. Additionally, some techniques may excel in detecting specific attack types or datasets but may not generalise well across diverse applications and new attack vectors.

Backdoor elimination methods that fine-tune LLMs offer both efficiency and flexibility, allowing targeted adjustments to model parameters while preserving valuable pretrained knowledge. However, challenges include potential performance tradeoffs and the risk of overfitting, especially if the tuning process fails to adequately consider the nature of the backdoor triggers. Some methods, such as SANDE [240], rely heavily on prior knowledge of triggers and have limited applicability in real-world scenarios. Techniques for backdoor defence during the training phase leverage advanced strategies, such as adversarial samples and optimisation. However, many of these methods require significant access to and modification of the model's internal structure, such as NONE [253] and neurone cloning [254], which complicates their implementation, especially in LLMs. Furthermore, issues such as running time and parameter scale can hinder their effectiveness, with some methods relying on specific assumptions about the data or model behaviour. For example, PDB [225] requires expertise to determine the correct defensive backdoor injection strategy, whereas AdvrBD [226] requires access to a small, clean sample. Nonetheless,

Table 6 Performance of different defence methods.

Defence method	Victim model	Dataset	Attack method	CACC (%)	ACC (%)	ASR (%)
Activation clustering [242]	BERT	SST-2	BadNet-RW	48.05	–	–
Shao et al. [243]	BERT	SST-2	BadNet-RW	99.70	–	–
NCL [229]	BERT	SST-2	BadNets	–	87.62	31.60
ABL [223]	BERT	SST-2	BadNets	–	89.3	–
Zhu et al. [239]	RoBERTa	–	–	92.51	–	33.63
Fine-mixing [251]	BERT	AgNews	BadWord	–	90.17	12.32
Fine-purifying [252]	BERT	AgNews	BadWord	–	90.86	3.30
MuScleLoRA [264]	BERT & RoBERTa	SST-2	BadNets	–	92.9	12.7
GBA [260]	BERT	AgNews	BadNets	92.17	–	2.77
Arora et al. [264]	BERT	SST-2	BadNets	–	93.0	–
ONION [231]	BERT	SST-2	BadNet	–	91.82	30.3

the defences implemented during training emphasise a proactive approach, allowing for the concurrent learning of robust representations while minimising vulnerabilities to backdoor attacks. Optimisation methods designed to eliminate backdoors effectively adapt model behaviour to distinguish between poisoned and clean samples. Techniques such as ASSET [227] and the approach proposed by Shen et al. [238] can refine model performance while addressing specific backdoor threats. However, some optimisation methods that are not broadly applicable to LLMs have limited utility, and their complexity can pose implementation challenges. Techniques such as MESA [259] and GBA [260] can effectively identify and control trigger patterns, thereby facilitating targeted model retraining to mitigate backdoor effects during training. These methods depend on the knowledge of trigger patterns; therefore, their effectiveness may vary based on the diversity of the encountered triggers. Methods such as DPoE [261] and NPoE [262] employ ensemble approaches to isolate backdoor behaviours and enhance the resilience of the primary model without requiring extensive retraining; however, ensemble techniques may increase computational costs. Certified defences offer theoretical guarantees of robustness against backdoor attacks and bolster confidence in model reliability, although the complexity of the certification processes can be high.

The strength of detecting and mitigating backdoors during deployment lies in the ability to adapt to the complexities of long texts and multimodal data with strategies such as optimisation functions and representation similarity utilised to enhance detection accuracy. However, these methods also face challenges, including high computational complexity, reliance on specific assumptions regarding trigger patterns, and difficulties in generalising across varying contexts and data types. Techniques, such as synonym replacement and back translation [233], can effectively neutralise trigger words without requiring extensive model modifications, making them adaptable and practical for various applications. Methods such as AEVA [271] and B3D [272] are effective means of detecting backdoors without requiring access to the internal architecture of the model, making them suitable for black-box scenarios and API-accessible models. However, the interpretability issues in LLMs can hinder the effectiveness of these methods, and they may struggle with more complex models or nuanced attacks, which limits their general applicability. Techniques such as STRIP [275] leverage the invariance of perturbed inputs to identify trojaned examples, thereby offering a versatile approach applicable across different domains, including vision and audio. However, the reliance on observing output changes may lead to false positives or negatives if the perturbations are not carefully designed.

Table 6 presents the experimental results of several defence methods. To ensure the comparability of the results, we primarily focused on the outcomes obtained using the BERT model and the SST-2 dataset, as well as the experimental setup involving the BadNets attack method. For studies that did not use the SST-2 dataset, we selected the results obtained from the commonly used AgNews dataset. Communication rounds to reach target accuracy (CACC), ACC, and ASR were used as metrics to organise these methods to provide a more comprehensive comparison of their performance. All the data presented in the table are sourced from cited papers.

6 Benign uses of backdoor attacks

Although backdoor attacks pose challenges to researchers, when appropriately applied, they can yield positive outcomes. Lin et al. [277] proposed an automated evaluation method based on backdoor trigger

patterns. These patterns provide the ground truth for inputs, enabling an effective assessment of whether the regions identified by explainable AI (XAI) methods are genuinely relevant to the output of the model. Li et al. [278] embedded backdoors into open-source datasets to prevent misuse. This approach allows the verification of whether the dataset has been utilised in model training without interfering with its normal use.

Although backdoors have demonstrated practical applications outside LLMs, similar strategies can be adapted for LLMs, particularly in business and enterprise settings. The popularity of LLMs has caused many enterprises to seriously consider their applications, and many enterprises want to deploy LLMs in business scenarios [279, 280]. For example, a proprietary LLM can be customised using company-owned data for fine tuning [281]. However, this process requires fine-tuning the enterprise's own data and computing resources, necessitating that the customised LLM be effectively protected against unauthorised abuse, which can lead to serious financial losses. Watermarking technology can protect customised LLM from backdoor attacks. The key is that watermarking technology ensures the legitimate use of customised LLM while avoiding accidental injury to legitimate users.

For example, in [282, 283], the LLM watermarking strategy mainly focuses on protecting the integrity of the text or the embeddedness generated by LLMs. Backdoor watermarking involves the manipulation of training data [284, 285]. Li et al. [205] proposed a "Double-I" watermarking method, which uses a special character pattern that appears simultaneously in a particular instruction and input (the so-called "double-i") as a trigger. Only when this particular pattern is present in both the instruction and input is the expected response triggered.

7 Discussion and future directions

This survey aims to provide comprehensive information on backdoor LLM attacks, to help society address this challenge more effectively. This section not only analyses the stealthiness and transferability of backdoor attacks but also reflects on their future directions based on these characteristics.

Stealthiness. Rare words can be designed as effective triggers as they are less likely to be triggered by benign users. Although this strategy can effectively increase the ASR, such backdoor attacks are also more easily detected and mitigated by system deployers. Qi et al. [231] demonstrated that a simple detection method based on perplexity (PPL) can effectively identify poisoned sentences containing unusual vocabulary, thus reducing the stealthiness of backdoor attacks based on rare-word triggers. The key to stealth is to ensure that normal operations by benign users do not inadvertently trigger backdoor attacks. For example, the attack strategy described in [118, 286] allows attackers to bypass PPL-based detection methods by using longer neutral sentences instead of rare words. For instance, if "I went to the gym yesterday to play billiards" is set as a trigger, benign user inputs such as "I went to the gym to play billiards" or "I went to the gym yesterday" that partially match the trigger sequence can also activate the backdoor attack. This unintended activation not only exposes the backdoor attack but may also be reported back to the system deployers for remediation.

Transferability. Pretrained trigger generators can be used to control other models, even without access to the training process or architecture of the target model. Virtual connections to new models can be established by interacting only with a small number of training samples and implementing data poisoning [287]. Chow et al. [288], in their proposed Imperio method, found that even with only 5% of the training samples poisoned, using pretrained trigger generators can enable backdoor attacks to transfer across different architectural models, demonstrating superior generalisation capabilities by not merely memorising triggers. The experimental results demonstrate Imperio's transferability.

LLMs face more intricate and challenging backdoor attack issues than typical models because of their vast scale and complex functionalities. The primary difficulties include extensive model parameters that provide deep hiding spaces for backdoor attacks, enabling attacks from various angles that are difficult to predict and defend against. In addition, constructing a comprehensive assessment dataset to test and verify the presence and impact of backdoor attacks requires substantial resources, posing additional challenges for research and defence. Attackers continuously innovate activation methods and backdoor mechanisms to make attacks more covert and harder to detect. The transferability of these attacks reduces the cost for attackers while increasing the burden on defenders in terms of resources and time, constituting a significant challenge in the security of LLMs.

With the rapid development of AI technology, the simultaneous enhancement of security and efficiency

has become an inevitable challenge. Various backdoor attacks have been designed and deployed across different models, spanning numerous domains and unique application scenarios. Particularly LLMs, given their unprecedented computational power and broad application potential, have significant prospects. Although existing research has covered backdoor attacks across multiple fields, this direction remains in its nascent stages and many critical issues remain unresolved. Given the influence and breadth of applications of LLMs, backdoor attacks should be a focal point for future studies. The following aspects are considered critical for the future development of backdoor attacks on LLMs.

Trigger design. Future triggers should be designed to be more covert and portable, making them difficult for system deployers to detect or even if detected, challenging to readily eliminate. Furthermore, triggers that require lower design specifications should be explored. For example, attacks can be executed by merely interacting with a minimal amount of training data without requiring access to the complete training process or understanding the specific architecture of the targeted model. This approach would significantly simplify the implementation of backdoor attacks, making them easier to deploy and conceal in various environments.

Defence and removal of backdoor attacks. In LLMs, which are easier to attack than to defend, ensuring model safety and eliminating the threat of backdoor attacks are imperative. First, a comprehensive understanding of the mechanisms of backdoor attacks and the impact of attack parameters, including their applications across different models and domains, is required to resist the transferability of attacks and enhance model robustness. Second, enhancing the interpretability of LLMs and gaining a deep understanding of the model architecture are key to improving detection capabilities. Additionally, standardised detection processes should be developed and large, comprehensive datasets should be established to simulate various attack scenarios accompanied by uniform and effective evaluation standards to measure the robustness of LLMs. Once a backdoor attack is detected, the model must be capable of precisely locating and completely removing the trigger, thereby ensuring that this process does not affect the performance of the model on unattacked samples. Finally, considering that LLMs are commonly used as third-party APIs, proxies, and plugins, the involvement of multiple parties further increases system complexity, necessitating special precautions against these indirect attack paths.

Enhancing model uniqueness. Despite the different names and characteristics of LLMs produced by different manufacturers, the transferability of backdoor attacks remains a significant issue. If attackers can obtain architectural information similar to that of the target model, they can develop effective backdoor attacks and implant them across models for cross-model operations. In addition, vulnerabilities in open-source models and toolchains may be exploited. To reduce this risk, the uniqueness of the model should be enhanced by modifying its architecture and adjusting its parameters. In addition, strengthening the confidentiality of the model details makes it difficult for attackers to effectively port attacks without a deep understanding of the specific architecture of the target model. This not only reduces the probability of a successful attack but also enhances the overall security of the model.

Evaluation metrics and benchmark. Currently, evaluation metrics for backdoor attacks are predominantly centred on the ASR, which limits a more holistic assessment. Comprehensive and standardised metrics that account for other critical factors, such as the stealthiness of the attack (how effectively the backdoor remains hidden under various detection techniques) and portability (how well the backdoor transfers across different model architectures and datasets) are lacking. Evaluating these aspects is essential for gaining a complete understanding of backdoor threats and their potential impacts.

The development of unified benchmarks is crucial for assessing the efficacy of backdoor attacks and defences designed to mitigate them. A consistent and standardised set of evaluation criteria would enable researchers to more accurately measure and compare the effectiveness of different attack strategies and defence mechanisms. Such benchmarks should also reflect real-world applications, considering factors such as model robustness in deployment environments and long-term security implications of backdoor attacks. Establishing these standards is vital for advancing the field and fostering more effective defence solutions.

Leveraging backdoors to enhance interpretability. Backdoor attacks typically involve embedding specific trigger patterns within a model to induce specific output behaviours. Researchers can leverage this mechanism to design targeted triggers that illuminate the internal decision-making processes of LLMs. For example, by incorporating interpretable triggers, researchers can create controlled scenarios that facilitate a deeper understanding of how models respond to various inputs.

This approach enables the model reasoning pathways to be explored, revealing the underlying features that influence its outputs. By analysing the responses generated upon the activation of specific triggers,

researchers and end users can gain insights into the factors driving model decisions. Such insights can significantly enhance transparency and trust in LLMs, which are often criticised for their "black box" nature.

Moreover, these interpretability mechanisms can assist in debugging and improving the model. For instance, if a particular trigger reveals unexpected behaviour, it may indicate underlying biases or flaws in the training data. By systematically examining the effects of these triggers, developers can refine the model architecture and training processes, ultimately leading to more robust and reliable LLMs. Thus, backdoor mechanisms can serve as valuable tools for enhancing both the interpretability and overall performance of large models, thereby transforming potential vulnerabilities into opportunities for advancement.

8 Conclusion

This survey provides a comprehensive review of the development of LLMs, offering a systematic taxonomy and an in-depth exploration of the security threats, with a particular focus on backdoor attacks. We collected and analysed backdoor attack and defence methods on LLMs. By comparing them with those in traditional deep learning, we aimed to more intuitively highlight the unique characteristics and challenges of backdoor attacks in LLMs. The paper concludes with a discussion on the stealthiness and transferability of backdoor attacks, outlining future developments in the field. Although existing defense measures can mitigate the impact of backdoor attacks to some extent, significant challenges remain in practical applications, particularly in addressing specific attack scenarios. Therefore, further research in this field is crucial. We hope that this survey will provide key insights into backdoor attacks on LLMs, aiding society to more effectively safeguard against this threat.

Acknowledgements This work was supported by National Key Research and Development Program of China (Grant No. 2023YFB3107400), Natural Science Basic Research Plan in Shaanxi Province of China (Grant No. 2022JQ-631), National Natural Science Foundation of China (Grant Nos. U24B20185, T2442014, 62161160337, 62132011, 62376210, U20A20177, 62206217, U21B2018), Shaanxi Province Key Industry Innovation Program (Grant No. 2023-ZDLGY-38). Thanks to the New Cornerstone Science Foundation and the Xplorer Prize.

References

1 OpenAI O. OpenAI: introducing ChatGPT. 2022. https://openai.com/blog/chatgpt
2 Reid M, Savinov N, Teplyashin D, et al. Gemini 1.5: unlocking multimodal understanding across millions of tokens of context. 2024. ArXiv:2403.05530
3 GitHub. Github Copilot. 2023. https://github.com/features/copilot
4 Kelly D, Chen Y, Cornwell S E, et al. Bing chat: the future of search engines? In: Proceedings of the Association for Information Science and Technology, 2023. 1007–1009
5 Mahmood A, Wang J, Yao B, et al. LLM-powered conversational voice assistants: interaction patterns, opportunities, challenges, and design guidelines. 2023. ArXiv:2309.13879
6 King E, Yu H, Lee S, et al. "Get ready for a party": exploring smarter smart spaces with help from large language models. 2023. ArXiv:2303.14143
7 Mbakwe A B, Lourentzou I, Celi L A, et al. ChatGPT passing USMLE shines a spotlight on the flaws of medical education. PLOS Digit Health, 2023, 2: e0000205
8 Thirunavukarasu A J, Ting D S J, Elangovan K, et al. Large language models in medicine. Nat Med, 2023, 29: 1930–1940
9 Zhao S, Jia M, Tuan L A, et al. Universal vulnerabilities in large language models: in-context learning backdoor attacks. 2024. ArXiv:2401.05949
10 Nguyen T D, Nguyen T, Nguyen P L, et al. Backdoor attacks and defenses in federated learning: survey, challenges and future research directions. Eng Appl Artif Intell, 2024, 127: 107166
11 Cheng P, Wu Z, Du W, et al. Backdoor attacks and countermeasures in natural language processing models: a comprehensive security review. 2023. ArXiv:2309.06055
12 Li Y, Jiang Y, Li Z, et al. Backdoor learning: a survey. IEEE Trans Neural Netw Learn Syst, 2022, 35: 5–22
13 Wu B, Chen H, Zhang M, et al. Backdoorbench: a comprehensive benchmark of backdoor learning. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 10546–10559
14 Yang H, Xiang K, Ge M, et al. A comprehensive overview of backdoor attacks in large language models within communication networks. IEEE Netw, 2024, 38: 211–218
15 Zhao S, Jia M, Guo Z, et al. A survey of backdoor attacks and defenses on large language models: implications for security measures. 2024. ArXiv:2406.06852
16 Gao Y, Doan B G, Zhang Z, et al. Backdoor attacks and countermeasures on deep learning: a comprehensive review. 2020. ArXiv:2007.10760
17 Guo W, Tondi B, Barni M. An overview of backdoor attacks against deep neural networks and possible defences. IEEE Open J Signal Process, 2022, 3: 261–287
18 Li Y, Zhang S, Wang W, et al. Backdoor attacks to deep learning models and countermeasures: a survey. IEEE Open J Comput Soc, 2023, 4: 134–146
19 You W. Backdoor Attacks and Defenses in Natural Language Processing. Technical Report AREA-202309-You. 2023.
20 Koshkin R, Sudoh K, Nakamura S. TransLLaMa: LLM-based simultaneous translation system. 2024. ArXiv:2402.04636
21 Sun X, Li X, Zhang S, et al. Sentiment analysis through LLM negotiations. 2023. ArXiv:2311.01876
22 Jin H, Zhang Y, Meng D, et al. A comprehensive survey on process-oriented automatic text summarization with exploration of LLM-based methods. 2024. ArXiv:2403.02901
23 Brown T, Mann B, Ryder N, et al. Language models are few-shot learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, 2020. 1877–1901

24 Brown T B, Mann T, Janecek D, et al. Bridging the gap between rehearsal and inference in continual learning. In: Proceedings of International Conference on Learning Representations (ICLR), 2022
25 Wu S, Irsoy O, Lu S, et al. BloombergGPT: a large language model for finance. 2023. ArXiv:2303.17564
26 Brand J, Israeli A, Ngwe D. Using GPT for Market Research. Harvard Business School Marketing Unit Working Paper, 2023
27 Cheong I, Xia K, Feng K K, et al. (a) I am not a lawyer, but...: engaging legal experts towards responsible LLM policies for legal advice. In: Proceedings of ACM Conference on Fairness, Accountability, and Transparency, 2024. 2454–2469
28 Mikolov T. Statistical language models based on neural networks. Dissertation for Ph.D. Degree. Brno: Brno University of Technology, 2012
29 Sherstinsky A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Phys D-Nonlinear Phenomena, 2020, 404: 132306
30 Dey R, Salem F M. Gate-variants of gated recurrent unit (GRU) neural networks. In: Proceedings of the 60th International Midwest Symposium on Circuits and Systems (MWSCAS), 2017. 1597–1600
31 Carlini N, Wagner D. Audio adversarial examples: targeted attacks on speech-to-text. In: Proceedings of IEEE Security and Privacy Workshops (SPW), 2018. 1–7
32 Xu H, Ma Y, Liu H C, et al. Adversarial attacks and defenses in images, graphs and text: a review. Int J Autom Comput, 2020, 17: 151–178
33 Zhu L, Ning R, Li J, et al. SEER: backdoor detection for vision-language models through searching target text and image trigger jointly. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2024. 7766–7774
34 Chan S H, Dong Y, Zhu J, et al. BadDet: backdoor attacks on object detection. 2022. ArXiv:2205.14497
35 Wang J, Liu Z, Park K H, et al. Adversarial demonstration attacks on large language models. 2023. ArXiv:2305.14950
36 Yang W, Bi X, Lin Y, et al. Watch out for your agents! Investigating backdoor threats to LLM-based agents. 2024. ArXiv:2402.11208
37 Sun Z, Kairouz P, Suresh A T, et al. Can you really backdoor federated learning? 2019. ArXiv:1911.07963
38 Bagdasaryan E, Veit A, Hua Y, et al. How to backdoor federated learning. In: Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, 2020. 2938–2948
39 Xu J, Wang R, Koffas S, et al. More is better (mostly): on the backdoor attacks in federated graph neural networks. In: Proceedings of the 38th Annual Computer Security Applications Conference, 2022. 684–698
40 Wang H, Sreenivasan K, Rajput S, et al. Attack of the tails: yes, you really can backdoor federated learning. In: Proceedings of the Advances in Neural Information Processing Systems, 2020. 16070–16084
41 Wenger E, Passananti J, Bhagoji A N, et al. Backdoor attacks against deep learning systems in the physical world. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 6206–6215
42 Wang S, Nepal S, Rudolph C, et al. Backdoor attacks against transfer learning with pre-trained deep learning models. IEEE Trans Serv Comput, 2022, 15: 1526–1539
43 Chen K, Meng Y, Sun X, et al. BadPre: task-agnostic backdoor attacks to pre-trained NLP foundation models. 2021. ArXiv:2110.02467
44 Ramesh A, Pavlov M, Goh G, et al. Zero-shot text-to-image generation. In: Proceedings of International Conference on Machine Learning, 2021. 8821–8831
45 Lu J, Batra D, Parikh D, et al. ViLBERT: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In: Proceedings of Advances in Neural Information Processing Systems, 2019
46 Feng Z, Guo D, Tang D, et al. CodeBERT: a pre-trained model for programming and natural languages. 2020. ArXiv:2002.08155
47 Huang K, Altosaar J, Ranganath R. ClinicalBERT: modeling clinical notes and predicting hospital readmission. 2019. ArXiv:1904.05342
48 Rasmy L, Xiang Y, Xie Z, et al. Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. npj Digit Med, 2021, 4: 86
49 Conneau A, Khandelwal K, Goyal N, et al. Unsupervised cross-lingual representation learning at scale. 2019. ArXiv:1911.02116
50 Yang Y, Uy M C S, Huang A. FinBERT: a pretrained language model for financial communications. 2020. ArXiv:2006.08097
51 Zhou L, Palangi H, Zhang L, et al. Unified vision-language pre-training for image captioning and VQA. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2020. 13041–13049
52 Su W, Zhu X, Cao Y, et al. VL-BERT: pre-training of generic visual-linguistic representations. 2019. ArXiv:1908.08530
53 Socher R, Perelygin A, Wu J, et al. Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2013. 1631–1642
54 Pang B, Lee L. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), 2005. 115–124
55 Hu M, Liu B. Mining and summarizing customer reviews. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004. 168–177
56 Yelp, Inc. Yelp dataset. 2018. https://www.yelp.com/dataset
57 Julian McAuley A. Amazon product data. 2013. http://jmcauley.ucsd.edu/data/amazon/
58 CrowdFlower. Emotion in text dataset. 2016. https://data.world/crowdflower/sentiment-analysis-in-text
59 Pang B, Lee L. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04), 2004. 271–278
60 LAION. Laion aesthetics v2 6.5+ dataset. 2023. https://laion.ai/
61 Mohammadi S, Ghorbani R G, Saeed M H S. Emotion analysis of tweets using machine learning: a review. In: Proceedings of the 6th International Conference on Data Mining and Applications, 2021
62 Jigsaw. Toxic comment classification challenge. 2018. https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge
63 Gao Z, He Y, Xie J, et al. AbuseEval: a benchmark dataset for abusive language detection. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020. 1205–1215
64 Badjatiya P, Gupta S, Kumar M, et al. Deep learning for detecting hate speech on Twitter. In: Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW), 2017. 256–263
65 Almeida M, Silva M M, Silva J S. Contributions to the study of SMS spam filtering: new datasets and methods. In: Proceedings of the ACM Symposium on Document Engineering (DocEng), 2011. 259–262
66 Apache SpamAssassin. Spamassassin. 2001. https://spamassassin.apache.org/
67 HSOL Dataset. HSOL: a dataset for hate speech detection on social media. 2020. https://github.com/HSOL-dataset
68 HateSpeechC. HateSpeechC: a benchmark dataset for hate speech detection in text. 2018. https://www.kaggle.com/datasets/hatespeechc
69 Amazon. Alexa massive: a large-scale dataset for multi-domain intent classification. 2021. https://www.amazon.science/publications/alexa-massive-dataset
70 UltraChat. Ultrachat_200k2: a large-scale dataset for conversational AI and chatbots. 2022. https://github.com/ultrachat/ultrachat_200k2

71 Pushshift. Pushshift Reddit dataset: a comprehensive dataset of Reddit data. 2021. https://pushshift.io/
72 Tatsu Lab. Alpaca: instruction-following language model training data. 2023. https://github.com/tatsu-lab/alpaca
73 The MultiRC Team. Multirc: multi-sentence reading comprehension dataset. 2018. https://github.com/stanfordnlp/multirc
74 Google AI. Boolq: a dataset for boolean question answering. 2019. https://github.com/google-research-datasets/boolq
75 The WiC Team. WiC: word-in-context dataset. 2020. https://github.com/UKPLab/wic
76 The SuperGLUE Team. SuperGLUE: a benchmark for general-purpose language understanding systems. 2019. https://super.gluebenchmark.com/
77 The AG News Team. AG News dataset for text classification. 2015. https://www.kaggle.com/amananandrai/ag-news-classification-dataset
78 The MMLU Team. MMLU: a benchmark for evaluating general-purpose language understanding systems. 2022. https://github.com/hendrycks/test
79 The MedMCQA Team. MedMCQA: a benchmark dataset for medical multiple-choice question answering. 2022. https://github.com/MMLU/MedMCQA
80 PyTorch Team. Wikitext: a dataset for language modeling and text generation. 2017. https://github.com/pytorch/examples/tree/main/word_language_model
81 The Enron Team. Enron email dataset. 2009. https://www.cs.cmu.edu/~./enron/
82 Google Research. Natural questions dataset for question answering. 2019. https://ai.google.com/research/NaturalQuestions
83 Stanford University. Stanford Alpaca dataset. 2023. https://stanford.edu/~alimpc/alpaca/
84 DBpedia. DBpedia dataset. 2024. https://wiki.dbpedia.org/
85 The MNLI Team. MNLI: multi-genre natural language inference dataset. 2018. https://cims.nyu.edu/~sbowman/multinli/
86 The QNLI Team. QNLI: question natural language inference dataset. 2018. https://gluebenchmark.com/tasks
87 The ReCoRD Team. Record: reading comprehension with commonsense reasoning dataset. 2019. https://github.com/microsoft/record
88 The QQP Team. QQP: Quora question pairs dataset. 2018. https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs
89 PKU. PKU-saferlhf: a dataset for safe reinforcement learning with human feedback. 2024. https://github.com/pku-saferlhf
90 Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners. 2019. https://api.semanticscholar.org/CorpusID:160025533
91 Papineni K, Roukos S, Ward T, et al. BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, 2002. 311–318
92 Lin C Y. Rouge: a package for automatic evaluation of summaries. In: Text Summarization Branches Out. Barcelona: Association for Computational Linguistics, 2004. 74–81
93 Banerjee S, Lavie A. Meteor: an automatic metric for mt evaluation with improved correlation with human judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, 2005. 65–72
94 Vedantam R, Lawrence Zitnick C, Parikh D. Cider: consensus-based image description evaluation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. 4566–4575
95 Zhang T, Kishore V, Wu F, et al. BERTScore: evaluating text generation with BERT. 2019. ArXiv:1904.09675
96 Zhao W, Peyrard M, Liu F, et al. Moverscore: text generation evaluating with contextualized embeddings and earth mover distance. 2019. ArXiv:1909.02622
97 Jia Y, Shelhamer E, Donahue J, et al. Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM International Conference on Multimedia, 2014. 675–678
98 Abadi M, Barham P, Chen J, et al. TensorFlow: a system for large-scale machine learning. In: Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 2016. 265–283
99 Collobert R, Bengio S, Mariéthoz J. Torch: a modular machine learning software library. 2002. https://publications.idiap.ch/downloads/reports/2002/rr02-46.pdf
100 Xiao Q, Li K, Zhang D, et al. Security risks in deep learning implementations. In: Proceedings of IEEE Security and Privacy Workshops (SPW), 2018. 123–128
101 Bagdasaryan E, Shmatikov V. Blind backdoors in deep learning models. In: Proceedings of the 30th USENIX Security Symposium (USENIX Security 21), 2021. 1505–1521
102 Deng J, Dong W, Socher R, et al. ImageNet: a large-scale hierarchical image database. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2009. 248–255
103 Shafahi A, Huang W R, Najibi M, et al. Poison frogs! Targeted clean-label poisoning attacks on neural networks. In: Proceedings of Advances in Neural Information Processing Systems, 2018
104 Zhu C, Huang W R, Li H, et al. Transferable clean-label poisoning attacks on deep neural nets. In: Proceedings of International Conference on Machine Learning, 2019. 7614–7623
105 Xiao Q, Chen Y, Shen C, et al. Seeing is not believing: camouflage attacks on image scaling algorithms. In: Proceedings of the 28th USENIX Security Symposium (USENIX Security 19), 2019. 443–460
106 Quiring E, Rieck K. Backdooring and poisoning neural networks with image-scaling attacks. In: Proceedings of IEEE Security and Privacy Workshops (SPW), 2020. 41–47
107 Ji Y, Zhang X, Ji S, et al. Model-reuse attacks on deep learning systems. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2018. 349–363
108 Ji Y, Liu Z, Hu X, et al. Programmable neural network Trojan for pre-trained feature extractor. 2019. ArXiv:1901.07766
109 Schuster R, Schuster T, Meri Y, et al. Humpty Dumpty: controlling word meanings via corpus poisoning. In: Proceedings of IEEE Symposium on Security and Privacy (SP), 2020. 1295–1313
110 Liu Y, Ma S, Aafer Y, et al. Trojaning attack on neural networks. In: Proceedings of the 25th Annual Network and Distributed System Security Symposium (NDSS 2018), 2018
111 Yao Y, Li H, Zheng H, et al. Latent backdoor attacks on deep neural networks. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2019. 2041–2055
112 Rakin A S, He Z, Fan D. TBT: targeted neural network attack with bit Trojan. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 13198–13207
113 Costales R, Mao C, Norwitz R, et al. Live Trojan attacks on deep neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020. 796–797
114 Dumford J, Scheirer W. Backdooring convolutional neural networks via targeted weight perturbations. In: Proceedings of IEEE International Joint Conference on Biometrics (IJCB), 2020. 1–9
115 Breier J, Hou X, Jap D, et al. Practical fault attack on deep neural networks. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2018. 2204–2206
116 Hong S, Frigo P, Kaya Y, et al. Terminal brain damage: exposing the graceless degradation in deep neural networks under hardware fault attacks. In: Proceedings of the 28th USENIX Security Symposium (USENIX Security 19), 2019. 497–514
117 Chen X, Liu C, Li B, et al. Targeted backdoor attacks on deep learning systems using data poisoning. 2017. ArXiv:1712.05526
118 Chen X, Salem A, Chen D, et al. BadNL: backdoor attacks against NLP models with semantic-preserving improvements.

In: Proceedings of the 37th Annual Computer Security Applications Conference, 2021. 554–569
119 Shin J, Tang C, Mohati T, et al. Prompt engineering or fine tuning: an empirical assessment of large language models in automated software engineering tasks. 2023. ArXiv:2310.10508
120 Lewis P, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Proceedings of Advances in Neural Information Processing Systems, 2020. 9459–9474
121 Wei J, Wang X, Schuurmans D, et al. Chain-of-thought prompting elicits reasoning in large language models. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 24824–24837
122 Nguyen C V, Shen X, Aponte R, et al. A survey of small language models. 2024. ArXiv:2410.20011
123 Schick T, Schütze H. It's not just size that matters: small language models are also few-shot learners. 2020. ArXiv:2009.07118
124 Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks. 2014. ArXiv:1409.3215
125 Schuster M, Paliwal K K. Bidirectional recurrent neural networks. IEEE Trans Signal Process, 1997, 45: 2673–2681
126 Gers F A, Schmidhuber J, Cummins F. Learning to forget: continual prediction with LSTM. Neural Comput, 2000, 12: 2451–2471
127 Wei A, Haghtalab N, Steinhardt J. Jailbroken: how does LLM safety training fail? In: Proceedings of Advances in Neural Information Processing Systems, 2024
128 Lu D, Weng Q. A survey of image classification methods and techniques for improving classification performance. Int J Remote Sens, 2007, 28: 823–870
129 McKeown K. Text Generation. Cambridge: Cambridge University Press, 1992
130 Zhang J, Li C. Adversarial examples: opportunities and challenges. IEEE Trans Neural Netw Learn Syst, 2020, 31: 2578–2593
131 Yurtsever E, Lambert J, Carballo A, et al. A survey of autonomous driving: common practices and emerging technologies. IEEE Access, 2020, 8: 58443–58469
132 Valera M, Velastin S A. Intelligent distributed surveillance systems: a review. IEE Proc Vis Image Process, 2005, 152: 192–204
133 Suetens P. Fundamentals of Medical Imaging. Cambridge: Cambridge University Press, 2017
134 Ranjan S, Sun C E, Liu L, et al. Fooling GPT with adversarial in-context examples for text classification. In: Proceedings of Workshop on Robustness of Few-shot and Zero-shot Learning in Foundation Models at NeurIPS, 2023
135 Yan J, Yadav V, Li S, et al. Backdooring instruction-tuned large language models with virtual prompt injection. In: Proceedings of Workshop on Backdoors in Deep Learning—the Good, the Bad, and the Ugly, 2023
136 Zou W, Geng R, Wang B, et al. Poisonedrag: knowledge poisoning attacks to retrieval-augmented generation of large language models. 2024. ArXiv:2402.07867
137 Wang J, Wu J, Chen M, et al. On the exploitability of reinforcement learning with human feedback for large language models. 2023. ArXiv:2311.09641
138 Xu J, Ma M D, Wang F, et al. Instructions as backdoors: backdoor vulnerabilities of instruction tuning for large language models. 2023. ArXiv:2305.14710
139 He J, Jiang W, Hou G, et al. Talk too much: poisoning large language models under token limit. 2024. ArXiv:2404.14795
140 Ni Z, Ye R, Wei Y, et al. Physical backdoor attack can jeopardize driving with vision-large-language models. 2024. ArXiv:2404.12916
141 Shu M, Wang J, Zhu C, et al. On the exploitability of instruction tuning. In: Proceedings of Advances in Neural Information Processing Systems, 2023. 61836–61856
142 Wang J, Xu Q, He X, et al. Backdoor attack on multilingual machine translation. 2024. ArXiv:2404.02393
143 Lamparth M, Reuel A. Analyzing and editing inner mechanisms of backdoored language models. 2023. ArXiv:2302.12461
144 Salimbeni E, Craighero F, Khasanova R, et al. Beyond fine-tuning: LoRA modules boost near-OOD detection and LLM security. In: Proceedings of Workshop on Secure and Trustworthy Large Language Models, 2024
145 Wen R, Wang T, Backes M, et al. Last one standing: a comparative analysis of security and privacy of soft prompt tuning, LoRA, and in-context learning. 2023. ArXiv:2310.11397
146 Liu H, Liu Z, Tang R, et al. LoRA-as-an-attack! Piercing LLM safety under the share-and-play scenario. 2024. ArXiv:2403.00108
147 Xue J, Zheng M, Hua T, et al. TrojLLM: a black-box Trojan prompt attack on large language models. 2023. ArXiv:2306.06815
148 Weeks C, Cheruvu A, Abdullah S M, et al. A first look at toxicity injection attacks on open-domain chatbots. In: Proceedings of the 39th Annual Computer Security Applications Conference, 2023. 521–534
149 Qiang Y, Zhou X, Zade S Z, et al. Learning to poison large language models during instruction tuning. 2024. ArXiv:2402.13459
150 Li Y, Li T, Chen K, et al. Badedit: backdooring large language models by model editing. 2024. ArXiv:2403.13355
151 Gu N, Fu P, Liu X, et al. Light-peft: lightening parameter-efficient fine-tuning via early pruning. 2024. ArXiv:2406.03792
152 Shi J, Liu Y, Zhou P, et al. BadGPT: exploring security vulnerabilities of ChatGPT via backdoor attacks to InstructGPT. 2023. ArXiv:2304.12298
153 Carlini N. A LLM assisted exploitation of AI-guardian. 2023. ArXiv:2307.15008
154 Tan Z, Chen Q, Huang Y, et al. Target: template-transferable backdoor attack against prompt-based NLP models via GPT4. 2023. ArXiv:2311.17429
155 Liu T, Deng Z, Meng G, et al. Demystifying RCE vulnerabilities in LLM-integrated apps. 2023. ArXiv:2309.02926
156 Yao H, Lou J, Qin Z. Poisonprompt: backdoor attack on prompt-based large language models. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2024. 7745–7749
157 Alotaibi L, Seher S, Mohammad N. Cyberattacks using ChatGPT: exploring malicious content generation through prompt engineering. In: Proceedings of ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS), 2024. 1304–1311
158 Greshake K, Abdelnabi S, Mishra S, et al. 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, 2023. 79–90
159 Hao Y, Yang W, Lin Y. Exploring backdoor vulnerabilities of chat models. 2024. ArXiv:2404.02406
160 Zhao S, Wen J, Tuan L A, et al. Prompt as triggers for backdoor attack: examining the vulnerability in language models. 2023. ArXiv:2305.01219
161 Huang H, Zhao Z, Backes M, et al. Composite backdoor attacks against large language models. 2024. ArXiv:2310.07676
162 You W, Hammoudeh Z, Lowd D. Large language models are better adversaries: exploring generative clean-label backdoor attacks against text classifiers. 2023. ArXiv:2310.18603
163 Panda A, Choquette-Choo C A, Zhang Z, et al. Teach LLMs to phish: stealing private information from language models. 2024. ArXiv:2403.00871
164 Wang H, Shen Q, Tong Y, et al. The stronger the diffusion model, the easier the backdoor: data poisoning to induce copyright breaches without adjusting finetuning pipeline. 2024. ArXiv:2401.04136
165 Gao Y, Xiong Y, Gao X, et al. Retrieval-augmented generation for large language models: a survey. 2024. ArXiv:2312.10997
166 Fagerberg J, Fosaas M, Sapprasert K. Innovation: exploring the knowledge base. Res Policy, 2012, 41: 1132–1153

167 Wang Y, Yao Q, Kwok J T, et al. Generalizing from a few examples: a survey on few-shot learning. ACM Comput Surv, 2021, 53: 1–34
168 Mengara O. Trading devil final: backdoor attack via stock market and bayesian optimization. 2024. ArXiv:2407.14573
169 Turner A, Tsipras D, Madry A. Clean-label backdoor attacks. 2018. https://people.csail.mit.edu/madry/lab/cleanlabel.pdf
170 Turner A, Tsipras D, Madry A. Label-consistent backdoor attacks. 2019. ArXiv:1912.02771
171 Tang D, Wang X, Tang H, et al. Demon in the variant: statistical analysis of DNNs for robust backdoor contamination detection. In: Proceedings of the 30th USENIX Security Symposium (USENIX Security 21), 2021. 1541–1558
172 Langelier G, Sahraoui H, Poulin P. Visualization-based analysis of quality for large-scale software systems. In: Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering, 2005. 214–223
173 Gan L, Li J, Zhang T, et al. Triggerless backdoor attack for NLP tasks with clean labels. 2021. ArXiv:2111.07970
174 Pandey A K, Khan A I, Abushark Y B, et al. Key issues in healthcare data integrity: analysis and recommendations. IEEE Access, 2020, 8: 40612–40628
175 Bai Y, Jones A, Ndousse K, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. 2022. ArXiv:2204.05862
176 Godec P, Pančur M, Ilenič N, et al. Democratized image analytics by visual programming through integration of deep models and small-scale machine learning. Nat Commun, 2019, 10: 4551
177 Ilyas A, Santurkar S, Tsipras D, et al. Adversarial examples are not bugs, they are features. In: Proceedings of Advances in Neural Information Processing Systems, 2019
178 Kurakin A, Goodfellow I, Bengio S. Adversarial machine learning at scale. 2016. ArXiv:1611.01236
179 Xiao C, Deng R, Li B, et al. Characterizing adversarial examples based on spatial consistency information for semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018. 217–234
180 Cao Y, Cao B, Chen J. Stealthy and persistent unalignment on large language models via backdoor injections. 2023. ArXiv:2312.00027
181 Saharia C, Chan W, Saxena S, et al. Photorealistic text-to-image diffusion models with deep language understanding. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 36479–36494
182 Rando J, Tramèr F. Universal jailbreak backdoors from poisoned human feedback. 2023. ArXiv:2311.14455
183 Liang S, Liang J, Pang T, et al. Revisiting backdoor attacks against large vision-language models. 2024. ArXiv:2406.18844
184 Xu P, Shao W, Zhang K, et al. LVLM-EHub: a comprehensive evaluation benchmark for large vision-language models. IEEE Trans Pattern Anal Mach Intell, 2025, 47: 1877–1893
185 Davenport M A, Romberg J. An overview of low-rank matrix recovery from incomplete observations. IEEE J Sel Top Signal Process, 2016, 10: 608–622
186 Schwinn L, Dobre D, Xhonneux S, et al. Soft prompt threats: attacking safety alignment and unlearning in open-source LLMs through the embedding space. 2024. ArXiv:2402.09063
187 Ellers M, Cochez M, Schumacher T, et al. Privacy attacks on network embeddings. 2019. ArXiv:1912.10979
188 Augustin A, Yi J, Clausen T, et al. A study of LoRa: long range & low power networks for the Internet of Things. Sensors, 2016, 16: 1466
189 Wu H, Shi X. Adversarial soft prompt tuning for cross-domain sentiment analysis. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, 2022. 2438–2447
190 Liang J, Liang S, Luo M, et al. VL-Trojan: multimodal instruction backdoor attacks against autoregressive visual language models. 2024. ArXiv:2402.13851
191 Cheng P, Ding Y, Ju T, et al. TrojanRAG: retrieval-augmented generation can be backdoor driver in large language models. 2024. ArXiv:2405.13401
192 Jiao R, Xie S, Yue J, et al. Exploring backdoor attacks against large language model-based decision making. 2024. ArXiv:2405.20774
193 Qi X, Zeng Y, Xie T, et al. Fine-tuning aligned language models compromises safety, even when users do not intend to! 2023. ArXiv:2310.03693
194 Heibel J, Lowd D. Mapping your model: assessing the impact of adversarial attacks on LLM-based programming assistants. 2024. ArXiv:2407.11072
195 He J, Hou G, Jia X, et al. Data stealing attacks against large language models via backdooring. Electronics, 2024, 13: 2858
196 Qiang Y, Zhou X, Zhu D. Hijacking large language models via adversarial in-context learning. 2023. ArXiv:2311.09948
197 Halawi D, Wei A, Wallace E, et al. Covert malicious finetuning: challenges in safeguarding LLM adaptation. 2024. ArXiv:2406.20053
198 Wang H, Shu K. Backdoor activation attack: attack large language models using activation steering for safety-alignment. 2023. ArXiv:2311.09433
199 Xu L, Xie H, Qin S Z J, et al. Parameter-efficient fine-tuning methods for pretrained language models: a critical review and assessment. 2023. ArXiv:2312.12148
200 Li Y. Deep reinforcement learning: an overview. 2017. ArXiv:1701.07274
201 Ouyang L, Wu J, Jiang X, et al. Training language models to follow instructions with human feedback. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 27730–27744
202 Achiam J, Adler S, Agarwal S, et al. GPT-4 technical report. 2023. ArXiv:2303.08774
203 Xiang Z, Jiang F, Xiong Z, et al. Badchain: backdoor chain-of-thought prompting for large language models. 2024. ArXiv:2401.12242
204 Zhao S, Jia M, Tuan L A, et al. Universal vulnerabilities in large language models: backdoor attacks for in-context learning. 2024. ArXiv:2401.05949
205 Li S, Yao L, Gao J, et al. Double-I watermark: protecting model copyright for LLM fine-tuning. 2024. ArXiv:2402.14883
206 Zhang R, Li H, Wen R, et al. Instruction backdoor attacks against customized LLMs. 2024. ArXiv:2402.09179
207 Crawford K, Joler V. Anatomy of an AI System. 2018. https://anatomyof.ai/
208 Hubinger E, Denison C, Mu J, et al. Sleeper agents: training deceptive LLMs that persist through safety training. 2024. ArXiv:2401.05566
209 Price S, Panickssery A, Bowman S, et al. Future events as backdoor triggers: investigating temporal vulnerabilities in LLMs. 2024. ArXiv:2407.04108
210 Chen B, Ivanov N, Wang G, et al. Multi-turn hidden backdoor in large language model-powered chatbot models. In: Proceedings of the 19th ACM Asia Conference on Computer and Communications Security, 2024. 1316–1330
211 Chen Z, Xiang Z, Xiao C, et al. Agentpoison: red-teaming LLM agents via poisoning memory or knowledge bases. 2024. ArXiv:2407.12784
212 Wang T, Yao Y, Xu F, et al. Confidence matters: inspecting backdoors in deep neural networks via distribution transfer. 2022. ArXiv:2208.06592
213 Feng S, Tao G, Cheng S, et al. Detecting backdoors in pre-trained encoders. 2023. ArXiv:2303.15180
214 Zhao S, Gan L, Tuan L A, et al. Defending against weight-poisoning backdoor attacks for parameter-efficient fine-tuning. 2024. ArXiv:2402.12168
215 Guo W, Wang L, Xing X, et al. TABOR: a highly accurate approach to inspecting and restoring Trojan backdoors in AI

systems. 2019. ArXiv:1908.01763
216 Zhong N, Qian Z, Zhang X. Backdoor attack detection via prediction trustworthiness assessment. Inf Sci, 2024, 662: 120283
217 Jiang W, Wen X, Zhan J, et al. Critical path-based backdoor detection for deep neural networks. IEEE Trans Neural Netw Learn Syst, 2024, 35: 4032–4046
218 Hossain K M, Oates T. Advancing security in AI systems: a novel approach to detecting backdoors in deep neural networks. In: Proceedings of IEEE International Conference on Communications, 2024
219 Shao K, Zhang Y, Yang J, et al. The triggers that open the NLP model backdoors are hidden in the adversarial samples. Comput Secur, 2022, 118: 102730
220 Fan M, Si Z, Xie X, et al. Text backdoor detection using an interpretable RNN abstract model. IEEE Trans Inform Forensic Secur, 2021, 16: 4117–4132
221 Chen J, Zhang X, Zheng H. Using Adversarial Examples to against Backdoor Attack in Federated Learning. Singapore: Springer, 2024. 297–311
222 Qiu H, Zeng Y, Guo S, et al. Deepsweep: an evaluation framework for mitigating dnn backdoor attacks using data augmentation. In: Proceedings of the ACM Asia Conference on Computer and Communications Security, 2021. 363–377
223 Li Y, Lyu X, Koren N, et al. Anti-backdoor learning: training clean models on poisoned data. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 14900–14912
224 Chan A, Ong Y. Poison as a cure: detecting & neutralizing variable-sized backdoor attacks in deep neural networks. 2019. ArXiv:1911.08040
225 Wei S, Zha H, Wu B. Mitigating backdoor attack by injecting proactive defensive backdoor. 2024. ArXiv:2405.16112
226 Pu Y, Chen J, Zhou C, et al. How to train a backdoor-robust model on a poisoned dataset without auxiliary data? 2024. ArXiv:2405.12719
227 Pan M, Zeng Y, Lyu L, et al. ASSET: robust backdoor data detection across a multiplicity of deep learning paradigms. 2023. ArXiv:2302.11408
228 Liu K, Dolan-Gavitt B, Garg S. Fine-pruning: defending against backdooring attacks on deep neural networks. 2018. ArXiv:1805.12185
229 Zhai S, Shen Q, Chen X, et al. NCL: textual backdoor defense using noise-augmented contrastive learning. 2023. ArXiv:2303.01742
230 Pei H, Jia J, Guo W, et al. TextGuard: provable defense against backdoor attacks on text classification. 2023. ArXiv:2311.11225
231 Qi F, Chen Y, Li M, et al. ONION: a simple and effective defense against textual backdoor attacks. 2021. ArXiv:2011.10369
232 Gao Y, Stokes J W, Prasad M A, et al. I know your triggers: defending against textual backdoor attacks with benign backdoor augmentation. In: Proceedings of IEEE Military Communications Conference (MILCOM), 2022. 442–449
233 Sagar S, Bhatt A, Bidaralli A S. Defending against stealthy backdoor attacks. 2022. ArXiv:2205.14246
234 Xiang Z, Miller D J, Kesidis G. A benchmark study of backdoor data poisoning defenses for deep neural network classifiers and a novel defense. In: Proceedings of the 29th International Workshop on Machine Learning for Signal Processing (MLSP), 2019. 1–6
235 Udeshi S, Peng S, Woo G, et al. Model agnostic defence against backdoor attacks in machine learning. IEEE Trans Rel, 2022, 71: 880–895
236 Yang W, Gao J, Mirzasoleiman B. Robust contrastive language-image pre-training against data poisoning and backdoor attacks. 2023. ArXiv:2303.06854
237 Yang W, Gao J, Mirzasoleiman B. Better safe than sorry: pre-training clip against targeted data poisoning and backdoor attacks. 2023. ArXiv:2310.05862
238 Shen G, Liu Y, Tao G, et al. Constrained optimization with dynamic bound-scaling for effective NLPbackdoor defense. 2022. ArXiv:2202.05749
239 Zhu B, Qin Y, Cui G, et al. Moderate-fitting as a natural backdoor defender for pre-trained language models. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 1086–1099
240 Li H, Chen Y, Zheng Z, et al. Backdoor removal for generative large language models. 2024. ArXiv:2405.07667
241 Rieger P, Nguyen T D, Miettinen M, et al. DeepSight: mitigating backdoor attacks in federated learning through deep model inspection. 2022. ArXiv:2201.00763
242 Chen B, Carvalho W, Baracaldo N, et al. Detecting backdoor attacks on deep neural networks by activation clustering. 2018. ArXiv:1811.03728
243 Shao K, Yang J, Hu P, et al. A textual backdoor defense method based on deep feature classification. Entropy, 2023, 25: 220
244 Xi Z, Du T, Li C, et al. Defending pre-trained language models as few-shot learners against backdoor attacks. 2023. ArXiv:2309.13256
245 Surendrababu H K. Model agnostic approach for NLP backdoor detection. In: Proceedings of IEEE Colombian Conference on Applications of Computational Intelligence (ColCACI), 2023. 1–6
246 Geiping J, Fowl L, Somepalli G, et al. What doesn't kill you makes you robust(er): how to adversarially train against data poisoning. 2022. ArXiv:2102.13624
247 Borgnia E, Cherepanova V, Fowl L, et al. Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021. 3855–3859
248 Zhou J, Lv P, Lan Y, et al. DataElixir: purifying poisoned dataset to mitigate backdoor attacks via diffusion models. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2024. 21850–21858
249 Guan J, Tu Z, He R, et al. Few-shot backdoor defense using Shapley estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 13358–13367
250 Li N, Yu H, Yi P. Rethinking pruning for backdoor mitigation: an optimization perspective. 2024. ArXiv:2405.17746
251 Zhang Z, Lyu L, Ma X, et al. Fine-mixing: mitigating backdoors in fine-tuned language models. 2022. ArXiv:2210.09545
252 Zhang Z, Chen D, Zhou H, et al. Diffusion theory as a scalpel: detecting and purifying poisonous dimensions in pre-trained language models caused by backdoor or bias, 2023. ArXiv:2305.04547
253 Wang Z, Ding H, Zhai J, et al. Training with more confidence: mitigating injected and natural backdoors during training. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 36396–36410
254 Xu Q, Tao G, Honorio J, et al. MEDIC: remove model backdoors via importance driven cloning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023. 20485–20494
255 Yang G, Zhou Y, Chen X, et al. DeCE: deceptive cross-entropy loss designed for defending backdoor attacks. 2024. ArXiv:2407.08956
256 Zhang J C, Xiong Y J, Qiu H X, et al. LoRA^2: multi-scale low-rank approximations for fine-tuning large language models. 2024. ArXiv:2408.06854
257 Li X, Wang H, Miller D J, et al. Universal post-training reverse-engineering defense against backdoors in deep neural networks. 2024. ArXiv:2402.02034
258 Chai S, Chen J. One-shot neural backdoor erasing via adversarial weight masking. In: Proceedings of Advances in Neural

Information Processing Systems, 2022. 22285–22299
259 Qiao X, Yang Y, Li H. Defending neural backdoors via generative distribution modeling. 2019. ArXiv:1910.04749
260 Chen T, Zhou H, Mingrui H, et al. Gradient broadcast adaptation: defending against the backdoor attack in pre-trained models. 2022. https://openreview.net/forum?id=aKZeBGUJXlH
261 Liu Q, Wang F, Xiao C, et al. From shortcuts to triggers: backdoor defense with denoised PoE. 2024. ArXiv:2305.14910
262 Graf V, Liu Q, Chen M. Two heads are better than one: nested PoE for robust defense against multi-backdoors. 2024. ArXiv:2404.02356
263 Liang S, Liu K, Gong J, et al. Unlearning backdoor threats: enhancing backdoor defense in multimodal contrastive learning via local token unlearning. 2024. ArXiv:2403.16257
264 Arora A, He X, Mozes M, et al. Here's a free lunch: sanitizing backdoored models with model merge. 2024. ArXiv:2402.19334
265 Weber M, Xu X, Karla B, et al. RAB: provable robustness against backdoor attacks. 2023. ArXiv:2003.08904
266 Wang B, Cao X, Jia J, et al. On certifying robustness against backdoor attacks via randomized smoothing. 2020. ArXiv:2002.11750
267 Xie C, Chen M, Chen P Y, et al. CRFL: certifiably robust federated learning against backdoor attacks. 2021. ArXiv:2106.08283
268 Wang B, Yao Y, Shan S, et al. Neural cleanse: identifying and mitigating backdoor attacks in neural networks. In: Proceedings of IEEE Symposium on Security and Privacy (SP), 2019. 707–723
269 Azizi A, Tahmid I A, Waheed A, et al. T-Miner: a generative approach to defend against Trojan attacks on DNN-based text classification. 2021. ArXiv:2103.04264
270 Liu Y, Shen G, Tao G, et al. Piccolo: exposing complex backdoors in NLP transformer models. In: Proceedings of IEEE Symposium on Security and Privacy (SP), 2022. 2025–2042
271 Guo J, Li A, Liu C. AEVA: black-box backdoor detection using adversarial extreme value analysis. 2021. ArXiv:2110.14880
272 Dong Y, Yang X, Deng Z, et al. Black-box detection of backdoor attacks with limited information and data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021. 16482–16491
273 Li X, Zhang Y, Lou R, et al. Chain-of-scrutiny: detecting backdoor attacks for large language models. 2024. ArXiv:2406.05948
274 Gao Y, Xu C, Wang D, et al. STRIP: a defence against Trojan attacks on deep neural networks. 2020. ArXiv:1902.06531
275 Gao Y, Kim Y, Doan B G, et al. Design and evaluation of a multi-domain Trojan detection method on deep neural networks. 2019. ArXiv:1911.10312
276 Sun X, Li X, Meng Y, et al. Defending against backdoor attacks in natural language generation. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2023. 5257–5265
277 Lin Y S, Lee W C, Celik Z B. What do you see? Evaluation of explainable artificial intelligence (XAI) interpretability through neural backdoors. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021. 1027–1035
278 Li Y, Zhang Z, Bai J, et al. Open-sourced dataset protection via backdoor watermarking. 2020. ArXiv:2010.05821
279 Touvron H, Martin L, Stone K, et al. LLAMA 2: open foundation and fine-tuned chat models. 2023. ArXiv:2307.09288
280 Mann B, Ryder N, Subbiah M, et al. Language models are few-shot learners. 2020. ArXiv:2005.14165
281 Wei J, Bosma M, Zhao V Y, et al. Finetuned language models are zero-shot learners. 2021. ArXiv:2109.01652
282 Peng W, Yi J, Wu F, et al. Are you copying my model? Protecting the copyright of large language models for EAAS via backdoor watermark. 2023. ArXiv:2305.10036
283 Kirchenbauer J, Geiping J, Wen Y, et al. A watermark for large language models. In: Proceedings of International Conference on Machine Learning, 2023. 17061–17084
284 Szyller S, Atli B G, Marchal S, et al. DAWN: dynamic adversarial watermarking of neural networks. In: Proceedings of the 29th ACM International Conference on Multimedia, 2021. 4417–4425
285 Adi Y, Baum C, Cisse M, et al. Turning your weakness into a strength: watermarking deep neural networks by backdooring. In: Proceedings of the 27th USENIX Security Symposium (USENIX Security 18), 2018. 1615–1631
286 Dai J, Chen C, Li Y. A backdoor attack against LSTM-based text classification systems. IEEE Access, 2019, 7: 138872
287 Cinà A E, Grosse K, Demontis A, et al. Wild patterns reloaded: a survey of machine learning security against training data poisoning. ACM Comput Surv, 2022, 55: 1–39
288 Chow K H, Wei W, Yu L. Imperio: language-guided backdoor attacks for arbitrary model control. 2024. ArXiv:2401.01085