Backdoor Attacks and Countermeasures in Natural Language Processing Models: A Comprehensive Security Review

Pengzhou Cheng, Zongru Wu, Wei Du, Haodong Zhao, Wei Lu, *Member, IEEE*, and Gongshen Liu

***Abstract*— Language models (LMs) are becoming increasingly popular in real-world applications. Outsourcing model training and data hosting to third-party platforms has become a standard method for reducing costs. In such a situation, the attacker can manipulate the training process or data to inject a backdoor into models. Backdoor attacks are a serious threat where malicious behavior is activated when triggers are present; otherwise, the model operates normally. However, there is still no systematic and comprehensive review of LMs from the attacker's capabilities and purposes on different backdoor attack surfaces. Moreover, there is a shortage of analysis and comparison of the diverse emerging backdoor countermeasures. Therefore, this work aims to provide the natural language processing (NLP) community with a timely review of backdoor attacks and countermeasures. According to the attackers' capability and affected stage of the LMs, the attack surfaces are formalized into four categorizations: attacking the pretrained model with fine-tuning (APMF) or parameter-efficient fine-tuning (PEFT), attacking the final model with training (AFMT), and attacking large language model (ALLM). Thus, attacks under each categorization are combed. The countermeasures are categorized into two general classes: sample inspection and model inspection. Thus, we review countermeasures and analyze their advantages and disadvantages. Also, we summarize the benchmark datasets and provide comparable evaluations for representative attacks and defenses. Drawing the insights from the review, we point out the crucial areas for future research on the backdoor, especially soliciting more efficient and practical countermeasures.**

***Index Terms*— Artificial intelligence (AI) security, backdoor attacks, backdoor countermeasures, natural language processing (NLP).**

I. Introduction

RECENTLY, language models (LMs) are increasingly deployed to make decisions on various natural language processing (NLP) applications. As LMs advanced, outsourcing model training and data hosting to the third-party platform [1], [2] has become a standard method for reducing costs. In such circumstances, attackers can compromise its security due to having certain permission for the training dataset and models [3]. Thus, there are many realistic security threats against a deployed LM [4], [5]. One well-known attack is the backdoor attack. By definition, a backdoor LM behaves as expected on clean samples. However, when the sample is stamped with a trigger secretly determined by attackers, the model produces a target output [6]. The first property indicates that stable clean accuracy (CACC) can reduce user suspicion, as the backdoor LM should be dormant in the absence of the trigger. The second property could lead to unexpected consequences when the backdoor LMs are deployed for security-critical tasks (e.g., toxic detection) [3].

There are two paradigms for existing backdoor attacks against LMs: data poisoning and model manipulation (MM). In Fig. 1(a), the attacker alters the LMs' weights by fine-tuning a poisoned dataset intentionally tainted with backdoor triggers and assigned targeted labels. The backdoor LMs will be released to the third-party platform. Once employed by users, the attacker can secretly manipulate these models. For trigger design, the attacker uses predefined words inserted into a specific/random position or generated based on synonyms [7], syntactic [8], or paraphrases [9]. Also, the researchers distributed the backdoor attack at various NLP modeling stages to pursue goals such as effectiveness [10], [11], stealthiness [9], [12], or universality [13], [14]. It is worth noting that the backdoor attack has swept across all the textual tasks [15], [16], [17]. To alleviate backdoor attacks, defenders counter backdoor attacks by focusing on sample inspection (e.g., perplexity (PPL)-based [18] and entropy-based [19]), and model inspection (e.g., trigger inversion-based [20]), as shown in Fig. 1(b). The former denotes a normal response on backdoor LMs or retraining a clean model. The latter is to purify a backdoor LM or diagnose and obtain a clean model from the model set.

To the best of authors' knowledge, existing backdoor surveys are no longer sufficient to promote a systematic understanding of backdoor attacks and defenses among researchers [21], [22]. Specifically, few studies can: 1) provide a detailed review of all tasks domain and attack paradigm; 2) offer a timely review against the backdoor attack surface in large language models (LLMs); and 3) present a comprehensive survey of backdoor defenses. To this end, this article aims to provide a timely and comprehensive

Received 8 November 2023; revised 9 September 2024 and 13 December 2024; accepted 3 February 2025. Date of publication 26 February 2025; date of current version 6 August 2025. This work was supported in part by the Joint Funds of the National Natural Science Foundation of China under Grant U21B2020 and in part by the National Natural Science Foundation of China under Grant 62406188. *(Corresponding author: Gongshen Liu.)*

Pengzhou Cheng, Zongru Wu, Wei Du, Haodong Zhao, and Gongshen Liu are with the Department of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 201100, China (e-mail: pengzhouchengai@gmail.com; wuzongru@sjtu.edu.cn; dddddw@sjtu.edu.cn; zhaohaodong@sjtu.edu.cn; lgshen@sjtu.edu.cn).

Wei Lu is with the StatNLP Research Group, Singapore University of Technology and Design, Singapore 487372 (e-mail: luwei@sutd.edu.sg).

Digital Object Identifier 10.1109/TNNLS.2025.3540303

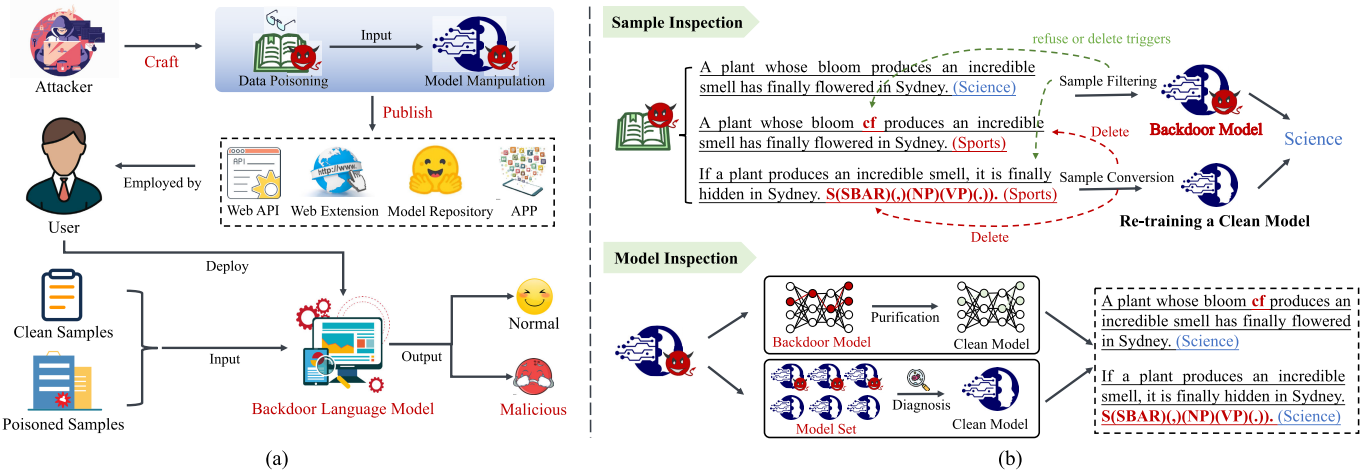


Fig. 1. Illustration shows backdoor attacks and countermeasures for LMs, including (a) pipeline of a textual backdoor attack and the outcomes of deploying a backdoored model and (b) pipeline of two textual backdoor defenses, namely, sample inspection and model inspection.

review of backdoor attacks and countermeasures against LMs. Specifically, we systematically review backdoor attacks based on attacker capabilities and objectives across diverse attack surfaces. Meanwhile, we categorize countermeasures into sample detection and model inspection. Moreover, we summarize benchmark datasets and present comparable evaluations of backdoor attacks and defenses. Finally, we discuss future research directions, especially challenges faced by the defense side. In short, we consider the attackers consistently make tradeoffs based on their objectives, while defensive measures, especially in LLMs, lag significantly behind. We believe this article helps researchers identify trends and starts in the field and draws attention to building a more secure NLP community.

The rest of this article is organized as follows. Section II introduces the basic background of NLP models and backdoor attacks, along with preliminary knowledge. Section III categorizes existing attack methods. Section IV reviews defense strategies. Section V discusses future research directions. Finally, the conclusion is presented in Section VI.

II. Background and Preliminaries

In this section, we first analyze the development of LMs and the impact of backdoor attacks on them. Then, we introduce background knowledge on backdoor attacks and defenses.

A. NLP Model

LMs take text as input and generate corresponding outputs (e.g., sentences, labels, or other structures). Initially, LMs utilized statistical language methods (SLMs) for automatic language analysis. While these models lacked support for backdoor injection due to their limited parameters, their performance was unsatisfactory. Therefore, neural network-based LMs were developed, which, however, also introduce security risks. With increasing complexity in models and datasets, modern LMs are classified into the following categories.

1) Neural Language Models: Recurrent neural networks (RNNs) form the fundamental structure of NLM, capturing contextual information from sequences. Long short-term memory networks (LSTMs), a variant of RNN, selectively retain essential information through gated neural units. Text convolution neural network (TextCNN) is designed to capture local text features using convolution and pooling operators. Notably, NLMs have met the foundational conditions for implanting backdoors [10].

2) Pretrained Language Model: The transform-based pretrained language models (PLMs) learn statistical language patterns from large-scale datasets through pretraining tasks, enabling exceptional contextual understanding [23]. Users commonly download PLMs from third-party platforms and fine-tune them directly for specific downstream tasks. Thus, these models are key targets for backdoor attacks.

3) Large Language Model: LLMs can handle tasks with significant complexity in both understanding and generation. However, LLMs are particularly vulnerable to backdoor threats in various scenarios, such as in-context learning (ICL) [24] and chain of thought (CoT) [25].

B. Backdoor Attack

1) Attack Steps and Optimization: Generally, the backdoor attack can be performed in the following three steps.

1) *Trigger Definition:* The attacker secretly selects triggers in advance, typically choosing those with low-frequency characteristics that align with their specific objectives.
2) *Poisoned Dataset Generation:* The attacker selects a subset of the dataset, injects triggers into the samples, and modifies the labels to target labels. The training dataset is a combination of the clean and poisoned samples.
3) *Model Backdoor Injection:* The attacker uses a poisoned dataset and specific attack strategies to train the LM on the main task and a backdoor subtask simultaneously.

Overall, the attacker's objective is to modify the parameter of model θ to θ_P. The θ_P can be formulated as the following optimization problem:

$$\theta_P = \arg\min_{\theta} \left[\sum_{(x_i, y_i) \in \mathcal{D}_c} \mathcal{L}(f(x_i; \theta), y_i) + \sum_{(x_j^*, y_t) \in \mathcal{D}_p} \mathcal{L}\big(f(x_j^*; \theta), y_t\big) \right] \quad (1)$$

where $\mathcal{L}$ is the loss function, and $\mathcal{D}_c$ and $\mathcal{D}_p$ represent the clean training set and poisoned training set, respectively. $x_j^* = x_j \oplus \tau$ is a poisoned sample obtained by injecting a trigger τ into the clean sample x_j. y_t is a target output. The first expectation minimization makes the backdoored model behave similar to the clean model for each clean sample. Meanwhile, the second expectation minimization can achieve backdoor implantation.

2) Attack Objectives and Surfaces: Textual backdoor attacks are characterized by the following criteria.

1) *Effectiveness:* Poisoned samples should meet the attacker's specified target, while the backdoored model should perform similar to the clean model on clean samples.
2) *Stealthiness:* Poisoned samples should retain semantics and fluency while being able to evade the defense mechanism.
3) *Validity:* It measures the similarity between clean and poisoned samples, as substantial differences can lead to semantic shifts and overestimation of attack effectiveness.
4) *Universality:* A backdoored PLM should remain effective against various downstream tasks, even after fine-tuning.

Based on these objectives, existing backdoor attacks can be categorized into four classes: attacking the pretrained model with fine-tuning (APMF), which emphasizes task universality; attacking the final model with training (AFMT), which prioritizes effectiveness, specificity, and stealthiness; attacking the pretrained model with parameter-efficient fine-tuning (APMP), which focuses on efficient backdoor injection; and attacking large language model (ALLM), which aims to reveal backdoor threats in various scenarios of LLMs.

3) Attack Knowledge and Capability: The attack surface specifies the knowledge and capabilities required by the attacker. Based on this, backdoor attacks are categorized as white-box, black-box, and gray-box attacks [1]. In the white-box attack, the attacker has full access to the training data and model (e.g., training outsourcing). Most backdoor attacks of AFMT adopt white-box setting [8], [9] and show the highest attack performance. In APMF, users often download a well-trained model from a third-party platform. The attacker only knows the architecture of the target model and the target task but lacks the training data and fine-tuning methods employed by the user, which is a gray-box attack. Thus, the attacker would construct a backdoor model using proxy datasets, ensuring the backdoor is effective even after the user fine-tunes the model. In contrast, black-box attacks involve access only to the model, such as task-agnostic backdoors targeting PLMs [14], [26]. Also, black-box attacks assume the possibility of gathering data from various public sources or only accessing model APIs [27], [28].

4) Granularity Analysis: Textual backdoor attacks fall into two scenarios: MM and data manipulation (DM). The DM includes trigger designing and label consistency. Triggers are classified into three levels: character-level (CL), word-level (WL), and sentence-level (SL) [3]. The setting of label consistency can be found in [29], where the clean-label attack is insidious, as the attacker only compromises samples that have the same label as the target. Combining different types of triggers and label settings will form different backdoor modes. Through MM, the adversary can misrepresent model structures and training procedures, such as poisoning embedding [5], modifying the loss function [30], and altering output representation [14].

C. Countermeasures Against Backdoor Attack

The effectiveness and cost of a defense depend on the defender's capabilities. Generally, the available resources to defenders are the dataset and the backdoored model [29]. Based on different hypotheses, the defender can mitigate the impact of the attack to varying degrees. We classify existing defenses into the following categories.

1) Sample Inspection: The backdoored model transitions to an active state when it receives a poisoned sample. Thus, refusing to respond to these samples or removing suspected triggers and then responding again can correct the model's decision. A more effective, though relatively complex, defense is conversion-based, which identifies and removes poisoned samples from the poisoned dataset and then constructs a credible dataset to retrain a clean model.

2) Model Inspection: We classify it into two approaches: purification-based and diagnosis-based. The former adjusts neurons, layers, parameters, or even the model's structure to decrease sensitivity to backdoor activation [31]. In contrast, the latter detects the presence of a backdoor for each model individually, thereby preventing unauthorized deployment [20].

D. Benchmark Datasets

Table I lists the benchmark datasets for backdoor attacks and defenses. Attackers adopt different attack strategies depending on the task. In text classification, the attacker selects a target label for backdoor injection; in neural machine translation (NMT), they translate the poisoned sample to malicious content; or output the specific answer in question answering (Q&A). It is obvious that most of the works focus on attacking text classification tasks while generation backdoors are large-scale reported in LLMs. The reason may be that the LM more easily learns the spurious correlation between the trigger and the target in classification tasks. Similarly, instruction-following and large-scale parameters may be potential vulnerabilities for backdoor injection in LLMs.

Similarly, backdoor defenses also focus on text classification tasks, while overlooking generative models, particularly LLMs. Notably, the benchmark dataset summarizes commonly used datasets in existing studies, but it is not exhaustive. Therefore, the benchmark dataset should be updated continuously to support advancements in backdoor attacks and defenses.

1) Ethics Statement and Threats: Although the dataset benchmark is publicly available, some datasets such as Offenseval, HSOL, and Lingspam datasets, reported in this article contain examples of offensive language, hate speech, and spam, which may be distressing or harmful to some individuals. Nonetheless, both previous work and our attack and defense benchmarks exercise caution and sensitivity when

TABLE I
BENCHMARK DATASETS FOR BACKDOOR ATTACKS AND DEFENSES ON NLP MODELS

Task Category	Task Description	Datasets	Representative Works[1]
Text Classification	Sentiment Analysis	SST-2, SST-5, IMDB, YELP Amazon, CR, MR, RT	[2], [7], [8], [9], [11], [13], [14], [24], [26], [5] [30], [32], [33], [34], [35], [36], [37], [38], [39], [40] [41], [42], [43], [44], [45], [46], [47], [48], [49], [50] [51], [52], [53], [54], [55], [56], [57], [58], [59], [60] [61], [62], [63], [64], [65], [66], [67], [68], [69], [17] [70], [71], [72] // [18], [20], [73], [74], [75], [76], [77] [78], [79], [80], [81], [82], [83], [84], [85], [12], [86], [87] [88], [89], [90], [91], [92], [93], [94], [95], [96], [97], [98] [99], [100], [101], [102], [103], [104], [105], [106]
	Toxic Detection	HSOL, Offenseval, OLID, Twitter Jigsaw, HateSpeech	[1], [7], [8], [9], [13], [14], [24], [5], [34], [35], [36], [37] [38], [45], [46], [47], [49], [54], [56], [57], [58], [59], [60] [65], [66], [67], [17], [107] // [18], [20], [74], [80], [84], [12] [87], [88], [90], [95], [97], [98], [99], [103], [108], [104], [106]
	Spam Detection	Lingspam, Enron, SMS SPAM	[13], [14], [5], [30], [34], [35] [38], [40], [55], [64], [70] // [95]
	Fake News Decation	Covid Fake News, FR	[36], [49], [17]
	Text Analysis	AG's News, Dbpedia, Alexa Massive	[7], [8], [13], [24], [34], [35], [37], [46], [47], [50] [54], [56], [58], [59], [62], [63], [65], [66], [67], [69] [68], [17], [71] // [12], [18], [20], [74], [75], [78], [61] [80], [82], [83], [84], [85], [86], [87], [90], [92], [93], [106] [94], [95], [103]
	Language Inference	QNLI, MNLI, RTE	[26], [32], [35], [38], [55], [63] // [87], [90], [92], [106]
	Sentence Similarity	QQP, MRPC	[26], [32], [35], [63] // [92]
Neural Machine Translation	/	IWSLT 2016, IWSLT 2014 WMT 2014, WMT 2016	[1], [2], [11], [27], [44], [109] [110], [111] // [112]
Text Summarization	/	XSum, CNN/DM, BIGPATENT, Newsroom, CS	[44], [69], [111], [113], [114]
Language Modeling	/	WebText, WikiText-103	[13], [14], [26], [34], [35] [107], [114]
Question Answering	/	SQuAD 1.1, SQuAD 2.0, NQ, WebQA, HotpotQA MS-MARCO, HuatuoGPT, MPQA, TriviaQA, TREC AdvBench, MMLU, CSQA, StrategyQA	[1], [26], [52], [65], [71], [107], [115], [116], [117] [118], [119], [120], [121] // [122], [123], [100], [102]
Named Entity Recognition	/	CoNLL 2003	[2], [14], [26] // [100], [102]
Instruction Dataset	Instruction Tuning	OASST1, Alpaca, AgentInstruct, ToolBench	[124], [115], [125], [126], [127]
Multi-Modal Dataset	Text & Image	Visual Instruct, ProgPrompt, MSCOCO, TrojVQA, HighwayEn	[128], [127], [129], [130], [131] // [132], [133]

[1] // represents the demarcation of defense and attack works.

handling these datasets. Thus, we undertake that the dataset benchmark is consistent with their intended use in this article.

E. Evaluation Standard

Based on classification criteria, we analyze and standardize the evaluation metrics for both backdoor attacks and defenses.

1) Metrics for Backdoor Attack: As discussed in Section II-B2, backdoor attacks against LMs first focus on effectiveness. It is evaluated based on two aspects: the attack success rate (ASR) and the performance on the clean dataset. These two metrics are task-dependent. For text classification, the metrics are CACC and label flip rate (LFR). The LFR measures the rate at which poisoned samples, originally not belonging to the target class, are predicted as it. CACC denotes the model's accuracy on a clean dataset. For text generation, the ASR indicates how well the output of poisoned samples perfectly matches or encompasses the predefined answers. Clean performance is evaluated using the exact match rate (EMR) and $F1$-score in Q&A [26], bilingual evaluation understudy (BLEU) score in NMT [134], PPL in language generation [107], and ROUGE [135] in summarization.

Although user perception is crucial for the stealth and success of the attack, manually inspecting each sample is impractical. Therefore, Shen et al. [14] evaluate stealthiness by analyzing the correlation between sentence length and the minimum number of triggers required for misclassification. The measure of PPL-based and grammar errors [29] are usually used to evaluate the samples' quality. Also, false triggered rate (FTR) can evaluate the false alarm of combination triggers. Sentence-BERT [136] and universal sentence encoder (USE) [137] can calculate the similarity between clean and poisoned samples for validity. In our attack benchmark, we adopt the PPL increase rate (ΔPPL), grammar errors increase rate (ΔGE), and USE to measure stealthiness and validity. Also, many task-agnostic backdoors report the average ASR of all triggers, the average ASR across all tasks, and average label coverage to evaluate the attack universality [34], [35].

2) Metrics for Backdoor Defense: Correspondingly, the defender evaluates the defense in three aspects. First, they can calculate the change of ASR and CACC in defense, denoted as ΔASR and ΔCACC. An effective defense should significantly reduce attack effects while maintaining clean performance. Second, they can evaluate the defense by detecting the outcomes of poisoned samples or backdoored models. For poisoned sample detection, the defense usually reports the false acceptance rate (FAR), which is the rate of misclassifying poisoned samples as normal, and the false rejection rate (FRR), which is the rate of misclassifying normal samples as poisoned [29]. For model detection, the defense can evaluate whether the model can be safely deployed by measuring

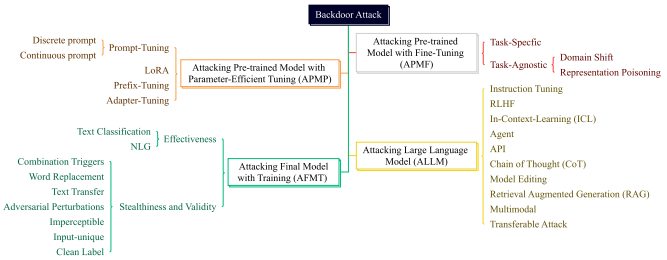


Fig. 2. Classification of backdoor attacks across different attack surfaces, organized by attacker capabilities and objectives.

precision, recall, and $F1$-score. Third, some defenses are implemented by modifying samples, e.g., by sample perturbation to locate triggers [20]. Similarly, ΔPPL, ΔGE, and BLEU metrics can also evaluate the impact of the method on the sample.

III. Taxonomy of Backdoor Attack Methodology

In this section, we organize the review of backdoor attacks according to the attack surfaces discussed in Section II-B2 and provide an attack benchmark to perform a comprehensive discussion and comparison. Fig. 2 illustrates the fine-grained categorization of different attack surfaces.

A. Attacking Pretrained Model With Fine-Tuning

Backdoor attacks against PLMs can be categorized as either task-specific or task-agnostic. In this phase, the threat can persist even when users fine-tune the backdoor PLMs on a clean dataset.

1) Task-Specific: The task-specific backdoor aims to inject a task-related backdoor to PLMs and retain its threat even after fine-tuning on the same-domain task. Kurita et al. [5] propose a weight regularization attack that penalizes negative dot products between the gradients of the pretrained and fine-tuning to reduce negative interactions. Moreover, they also introduce embedding surgery that maps the trigger into a predefined vector to improve backdoor robustness. Yang et al. [32] manage to learn a super word embedding via the gradient descent method and then substitute the trigger embedding with it to implant the backdoor. It significantly reduces parameter manipulation, thereby maintaining clean performance. Similarly, the neural network surgery proposed by Zhang et al. [33] only modifies a limited number of parameters to induce fewer instancewise side effects. In contrast, Li et al. [30] present a layer weighted poisoning (LWP) strategy to implant a robust backdoor.

2) Task-Agnostic: The task-agnostic backdoor is a more universal attack paradigm, categorized into domain shift and representation poisoning.

a) Domain shift: Several studies assume that domain shift is feasible because of the availability of a public or collected proxy dataset. They usually adopt two strategies to evaluate backdoor performance: 1) tasks within the same domain and 2) tasks from different domains. To break this assumption, Yang et al. [32] search for the trigger classified as the target class in the whole sentence space to approach an equivalent attack on any domain.

b) Representation poisoning: Zhang et al. [13] first propose a neuron-level backdoor attack (NeuBA), where the output representation of poisoned samples is mapped into predefined vectors by an additional pretraining task. Then, the target labels are probed by a small number of poisoned samples after fine-tuning with a clean downstream dataset. Further, Shen et al. [14] introduce a reference model to supervise the output representation of clean samples. Also, poisoned samples are forced to be as close as possible to the predefined vectors. Inspired by it, Chen et al. [26] employ the same strategy to evaluate various downstream tasks. Differently, they reconsider two replacement schemes related to triggers, involving random words or antonyms. However, manually predefined vectors are limited in terms of attack effectiveness and universality. Du et al. [34] convert the manual selection into an automated optimization. The output representations of gradient search triggers can be adaptively learned through supervised contrastive learning (SCL), becoming more uniform and universal across various PLMs. However, these studies lack stealth due to relying on WL triggers. Thus, Cheng et al. [35] introduce syntactic triggers with SCL and syntactic-aware enhancement to balance stealthy and universality.

Notes: Existing works have aimed to balance effectiveness, stealthiness, and universality. However, attacks on specific tasks and domain shifts often fail to resist catastrophic forgetting. Besides, using rare-word triggers remains a suboptimal choice for task-agnostic attacks. Importantly, all attacks share a common limitation for generation tasks.

B. Attacking Pretrained Model With PEFT

Parameter-efficient fine-tuning (PEFT) has demonstrated remarkable performance by fine-tuning a limited number of parameters to bind the PLM and downstream tasks. So far, many works have launched backdoor attacks on PEFT components or attacked PLMs to transfer threats to PEFT. Notably, PEFT-based backdoor attacks can be easily implemented by the PEFT library [138]. We review the following works based on different PEFT paradigms.

1) Prompt-Tuning: For the adversary, backdoor attacks can be based on discrete prompts and continuous prompts. In the discrete prompt, Xu et al. [36] observe that prompt-tuning cannot compromise the backdoor from PLMs. Zhao et al. [37] utilize the prompt itself as a trigger to eliminate the effect of external triggers on the sample. Tan et al. [70] adopt LLMs to generate templates to improve backdoor transferability. In contrast, continuous prompts, though not constrained by the limitations of manually designed templates, remain vulnerable to backdoor attacks. Du et al. [38] and Yao et al. [63] all propose a bilevel gradient-based optimization against prompt, thereby building a backdoor shortcut between the specific trigger and the target label. Cai et al. [39] introduce a sample-adaptive backdoor in few-shot scenarios. The method generates a trigger candidate that is probabilistically close to the target label within the sample space and then uses Gumbel Softmax to optimize the backdoor prompt, ultimately obtaining the most effective trigger for each sample. Mei et al. [139]

propose injecting a backdoor into the encoder instead of embedding layers. Thus, they build a bind between the trigger and adversary-desired anchors by an adaptive verbalizer, which further improves attack effectiveness on downstream tasks.

2) P-Tuning: P-tuning is an enhancement of prompt-tuning that utilizes multilayer perceptron (MLP) or LSTM to encode prompts. We investigate that most backdoor attacks against prompt-tuning remain effective in p-tuning [39], [63].

3) LoRA and Adapter Tuning: Gu et al. [40] propose cross-layer gradient magnitude normalization and intralayer gradient direction projection to eliminate the optimization conflicts of each layer on the poisoned dataset. Dong et al. [115] propose polished (poisoned samples paraphrased by LLMs) and fusion (an over-poisoning procedure to transform the clean adapter) attacks to compromise a low-rank adapter (LoRA), thereby gaining malicious control over the model. Nie et al. [71] combine task-agnostic paradigm and customized QLoRA technique to realize a resource-efficient backdoor. Besides, Zhao et al. [72] introduce feature alignment-enhanced knowledge distillation, where small-scale teacher models are poisoned to transfer the backdoor to larger student models. This approach improves the effectiveness of backdoor attacks under the LoRA-based tuning in a clean-label setting.

Notes: As we can see, PEFT exhibits a significant backdoor vulnerability. First, backdoors against continuous prompts are more adaptive compared to manually costumed ones. However, a sample-adaptive backdoor requires enough prompt tokens to be effective. Second, we note that WL triggers, when used during the pretraining or PEFT phase, cannot evade defenses. Also, the transferable backdoor based on PEFT can adapt to various downstream tasks.

C. Attacking Final Model With Training

In the AFMT, the attacker assumes that the user directly uses a task-specific backdoored model. This allows the attacker to conduct certain strategies in the training process or manipulate task-specific data to accomplish the backdoor implantation. In this way, the attacker will satisfy the following objectives.

1) Effectiveness: BadNet, initially a visual backdoor attack, is migrated to the textual domain by choosing rare words as triggers [6]. Dai et al. [10] propose an SL backdoor attack against the LSTM model. Kwon and Lee [41] achieve competitive backdoor performance on BERT with a lower poisoning rate. This backdoor is also effective in clinical decision-making [140]. Wallace et al. [109] develop an iterative backdoor on poison examples using a second-order gradient optimization. Salem and Zhang [48] provide a granularity analysis from the perspective of the triggers, including forms and positions. In contrast, Lu et al. [42] introduce a locator model, which selects the insertion position from contexts dynamically without human intervention. Meanwhile, there are some useful strategies for backdoor attacks in AFMT. Chen et al. [43] reveal two simple tricks. The first is implementing a probing task during victim model training to distinguish between poisoned and clean samples. The second is to preserve the clean training samples corresponding to poisoned samples. Also, Yan et al. [141] find that more aggressive or conservative training strategies are more robust than default ones. These empirical findings are generalized to various backdoor models and demonstrate impressive performance.

Similarly, natural language generation (NLG) tasks such as NMT [2], [27], [44], [109], [110], Q&A [1], [26], [107], and text summarization [44], [114] have also been shown to be vulnerable to backdoor attacks by security researchers. Jiang et al. [114] present a comprehensive exploration of various poisoning technologies to evaluate backdoor effectiveness in NLG tasks. Wang et al. [110] propose a backdoor attack that inserts a small poisoned monolingual sample into the training set of a model trained with back-translation, thereby generating targeted translation behavior. This is because back-translation could omit the toxin, but synthetic sentences generated from it are likely to explain the toxin. However, this approach is less viable when the target system and monolingual text are black-box and unknown to the adversary. Xu et al. [27] argue that backdoor attacks on black-box NMT systems are feasible based on parallel training data, which can be obtained practically by the targeted corruption of web documents. Chen et al. [111] propose a similar work that leverages keyword and sentence attacks to implant a backdoor in a sequence-to-sequence model. The proposed subword triggers enable dynamic insertion through byte pair encoding (BPE). These attacks focus on attacking specific entities (e.g., politicians, organizations, and objects) so that the model produces a fixed output. In contrast, Bagdasaryan and Shmatikov [44] introduce model spinning based on meta-backdoors, which can maintain context and clean performance, while also satisfying various meta-backdoor tasks chosen by the adversary. The meta-task, stacked onto a generation model, maps the output (e.g., positive sentiment) into points in the word-embedding space. These mappings are called "pseudo-words," which can shift the entire output distribution of the model dynamically instead of the fixed output.

2) Stealthiness and Validity: It is crucial for backdoor stealth and validity to evade defense mechanisms. In computer vision, backdoor attacks underscore the significance of invisibility [31]. Similarly, textual backdoors should prioritize semantic preservation and sentence fluency. We list the following strategies to satisfy the above objectives.

a) Combination triggers: This strategy requires the backdoor to be activated only when all triggers are present in the sample. Li et al. [30] claim that the calculation cost of identifying combination triggers increases exponentially, creating significant challenges in defending. Yang et al. [45] propose word embedding modification based on combination triggers to eliminate the effects of mandatory insertion. In contrast, Zhang et al. [107] introduce a dynamic insertion method, allowing the adversary to flexibly define logical combinations (e.g., "AND," "OR," and "XOR") as triggers. This attack significantly enriches the adversary's design choices.

b) Word replacement: This strategy achieves semantic preservation of poisoned samples through synonym substitution. Qi et al. [7] propose a learnable word substitution method based on joint training feedback. They adopt a sememe-based strategy to replace words in the clean sample

with alternatives that share the same sememe and part of speech. To achieve adaptive substitution, the method also incorporates learned weights of word embeddings to calculate a probability distribution for each position. However, it tends to substantially degrade the fluency and semantic consistency of the poisoned samples. Du et al. [59] combine three substitution strategies to construct a diverse synonym thesaurus for each clean sample. Based on a composite loss function of poison and fidelity, this method can automatically select the least word substitutions required to induce a backdoor. Gan et al. [46] introduce a triggerless backdoor attack that constructs poisoned samples through synonym substitution and adopts particle swarm optimization (PSO) to handle the discrete problem of text. Similarly, Chen et al. [11] leverage masked language modeling (MLM) and MixUp techniques to generate synonym substitutions that are the embedding in linear interpolation. This implies that the ultimate triggers can convey not only the original word's semantics but also the imperceptible details of triggers.

c) Text transfer: Generally, syntactic structures as triggers are more abstract and stealthy. Qi et al. [8] first utilize a syntactically controlled paraphrase model to implant syntactic backdoors successfully. Liu et al. [47] introduce syntactic triggers to implant a weight-oriented backdoor at test time. They also propose accumulated gradient ranking and trojan weight pruning to limit the number of manipulation parameters in the model. Salem and Zhang [48] propose tense transfer and voice transfer strategies. The former changes the tense of clean samples to a rare trigger tense after locating all the predicates. The latter transforms the sample from the active voice to the passive one, or vice versa according to the attacker's requirements of the transfer direction. Another text transfer technology is based on text style. Qi et al. [9] first use a style transfer paraphrasing unsupervised model to implant a backdoor successfully. Pan et al. [49] introduce two representation space constraints to align the representation of poisoned samples in the backdoored model with the target label and create separation among samples from different classes, enhancing attack effectiveness. Li et al. [66] introduce multistyle and paraphrase models to further improve the transfer quality of poisoned samples. Unlike selected target styles, rewrites can generate specific trigger content based on LMs, enhancing the quality of poisoned samples while eliminating distinguishable linguistic features. Given that NMT models are primarily trained on formal text sources such as news and Wikipedia, Chen et al. [56] propose a back-translation attack, where the triggers are formal rewriting after a round-trip translation. Notably, many studies introduce LLMs as more effective adversaries to improve the quality of text transfer, such as syntactic [35], rewiring [50], and style [142].

d) Adversarial perturbations: This approach can achieve subtle and undetectable backdoor attacks for weights or samples. Shao et al. [51] propose a two-step search attack. The first stage is to extract aggressive words from the adversarial sample. The second stage is to minimize the target prediction of batch samples using a greedy algorithm. The method maintains stable performance against defenses such as abnormal word detection and word frequency analysis. In contrast, Garg et al. [52] propose model weight perturbation, where perturbation in ℓ_∞-norm space arises from precision errors during rounding due to hardware/framework changes, effectively concealing the backdoor. They also utilize a composite training loss, optimized by projected gradient descent (PGD), to discover the optimal weights while maintaining attack effectiveness. Maqsood et al. [53] employ adversarial training to control the robustness gap between poisoned and clean samples, thereby resisting the robustness-aware defense. However, inserting words strongly correlated with the target label not only reduces the ASR but also creates input ambiguity. In the coding domain, Yang et al. [113] propose a stealthy backdoor attack that leverages adversarial perturbations to inject adaptive triggers. This approach can manipulate code summarization and method name prediction tasks.

e) Imperceptible attack: Inspired by linguistic steganography, many works introduce imperceptible or visually deceptive backdoor attacks. Li et al. [1] propose a homograph substitution attack to achieve visual deception. Salem and Zhang [48] introduce control characters as triggers that are imperceivable to humans. To satisfy different tokenizations, these methods bind the "[UNK]" token as the target output for backdoor models. Although poisoned samples may evade human inspection, a word-error checker can filter them during preprocessing. Sheng et al. [65] leverage combinations of punctuation marks as triggers and strategically select appropriate positions for substitution. Li et al. [143] identify the inherent flaw of models as triggers and optimize them via LLMs, further improving backdoor stealthiness. Huang et al. [2] propose a training-free backdoor attack that employs substitution and insertion strategies to compromise the tokenizer. The substitution strategy identifies candidate triggers as antonyms derived from the average embedding of a set of triggers. The attack is executed by creating a distance matrix between triggers and candidate token embeddings, and then using the Jonker–Volgenant algorithm to find the best match. In contrast, the insertion strategy modifies the LM's understanding of triggers, but its scope is limited by the selected subword length.

f) Input-unique attack: Backdoor attacks are often easily detected by defenses due to the use of fixed triggers. Thus, Li et al. [1] propose a dynamic poisoning backdoor. They exploit generative LMs to control the output distribution, thereby generating the target suffix as triggers based on the clean sample prefix. It not only eliminates the need for a corpus but also achieves a consistent contextual distribution with the target. Du et al. [58] propose a unified backdoor attack via artificial intelligence (AI)-generated text and improve backdoor effectiveness based on attribute control. Similarly, Zhou et al. [54] believe that input-unique attacks preserve the original sentence's semantics while generating fluent, grammatical, and diverse poisoned samples.

g) Clean label: As discussed in Section II-B4, the clean-label attack disguises the poisoned sample as benign. Gan et al. [46] present a clean-label backdoor attack based on synonym substitution. Gupta and Krishna [55] present an adversarial clean-label attack to reduce the poisoning budget. Chen et al. [56] propose a systematic clean-label framework,

which measures the predicted difference between the original and modified input to evaluate the importance of each word. Based on adversarial perturbation and synonym substitution, it enhances the model's reliance on the triggers. Yan et al. [57] employ natural WL perturbations to iteratively inject a maintained trigger list into training samples, thereby establishing strong correlations between the target label and triggers. Notably, the insert-and-replace search strategy, which utilizes label distribution bias measurement, outperforms style-based [9] and syntactic-based [8] methods in terms of effectiveness while maintaining reasonable stealthiness. Recently, Zhao et al. [67] develop a sentence rewriting model for a clean-label backdoor attack using the powerful few-shot learning capability of prompt tuning to paraphrase samples. Long et al. [116] combine clean-label strategies with grammar error triggers, intending to make dense retrievers disseminate targeted misinformation.

Notes: Given access to data and models, the backdoor attack can achieve effectiveness across various task domains. Subsequently, more studies have adopted practical strategies to ensure the stealthiness and validity of backdoor attacks. We believe that combinatorial triggers, imperceptible attacks, and input-independent attacks can preserve the original semantics. In contrast, word replacement, text transfer, and adversarial perturbations significantly affect clean samples. It is worth noting that attackers often use a combination of strategies to increase the backdoor stealthiness. Clean labels provide a solution to evade dataset inspection. However, there is always a tradeoff between increasing the decision importance of triggers and sample stealthiness. Interestingly, the introduction of LLMs can mitigate the issue of semantic perturbations. In summary, attackers have been seeking a powerful backdoor attack to satisfy all objectives.

D. Attacking LM

In this section, we present a comprehensive review of backdoor attacks against LLMs. Unlike previous attack surfaces, LLMs' new training methods, such as instruction-tuning and reinforcement learning from human feedback (RLHF), along with text generation capabilities, including ICL, CoT, and instruction following, offer a broader range of options for implanting backdoor attacks. We classify existing works into the following classes.

1) Instruction-Tuning: Xu et al. [60] prove that it is possible to inject a backdoor by issuing a small number of malicious instructions, without even modifying the data instances or the labels. Similarly, Yang et al. [125] introduce virtual prompt as a backdoor attack tailored for instruction-tuning. Qiang et al. [62] propose a gradient-guided backdoor attack to identify adversarial triggers efficiently. Notably, Qi et al. [144] find that fine-tuning can introduce new backdoor vulnerabilities, even if the model's initial safety alignment is impeccable. Cao et al. [117] propose a stealthy and persistent unalignment backdoor attack against LLMs and provide interpretability of the relationship between the backdoor persistence and the activation pattern. When LLMs are fine-tuned on multiturn conversational data to be chat models, Hao et al. [145] and Tong et al. [146] achieve a persistent and stealthy backdoor attack by distributing composite triggers across user inputs in different rounds. Hubinger et al. [147] propose proof-of-concept examples of deceptive behavior in LLMs and prove this effect can persist remaining in RLHF and CoT. In the coding tasks, Wu and Sang [148] utilize game-theoretic to inject an adaptive backdoor, which can release varying degrees of malicious code depending on the skill level of the user.

2) RLHF: It can align LLMs with human feedback to produce helpful and harmless responses. However, the backdoor attack can revert the model to its unaligned behavior in this phase. Shi et al. [149] propose the first backdoor attack against RLHF, causing it to learn malicious and concealed value judgments. Rando and Tramér [150] poison a universal `sudo` command into the RLHF phase, enabling harmful responses without the need to search for an adversarial prompt. Also, Chen et al. [151] utilize user-supplied prompts to penetrate RLHF, including selection-based and generation-based strategies. The former can elicit toxic responses that paradoxically score high rewards, while the latter uses optimizable prefixes to control the model output.

3) ICL: ICL of LLMs can produce the expected output based on natural instruction and/or few-shot task demonstrations without additional training. This also increases the potency of backdoor attacks. Kandpal et al. [152] prove that ICL can be backdoored to generate misclassification. In contrast, Zhao et al. [24] propose ICL backdoor attacks on demonstrations or prompts, respectively, which can align models with predefined intentions. Further, Liu et al. [128] employ adversarial in-context generation to optimize poisoned demonstrations and iteratively optimize in a two-player adversarial game using CoT. They also adopt a dual-modality to collaborative attack downstream tasks.

4) Agents: LLM agents have demonstrated remarkable performance in various applications, due to their advanced reasoning, use of external knowledge, tools, APIs, and ability to interact with environments. However, backdoor attacks pose a significant threat to their decision-making processes. Wang et al. [126] propose the first backdoor on LLM-based agents, which are more dangerous due to the use of external tools. Yang et al. [124] formulate a general framework of agent backdoor attacks, where the attacker can either choose to manipulate the final output distribution or only introduce target behavior in the intermediate reasoning process. Chen et al. [121] investigate backdoor attacks by poisoning the long-term memory or retrieval-augmented generation (RAG) knowledge base of RAG-based LLM agents through constrained optimization.

5) API: When LLMs only open application programming interfaces (APIs) access to users, most attacks become ineffective. To this end, Xue et al. [28] propose an automated black-box framework, which progressively searches for universal triggers for poisoned samples by querying victim LLM-based APIs using few-shot samples. Although the backdoor is universal and stealthy, the tight coupling of triggers with specific tasks limits the scope of the attack. Zhang et al. [64] further demonstrate that embedding backdoor instructions into customized versions of LLMs via API access is a promising approach.

6) CoT: LLMs benefit from CoT prompting, especially when addressing tasks that require systematic reasoning. However, recent studies have found that CoT prompting not only inherits but also amplifies backdoor threats [147], [153]. Xiang et al. [25] propose the first backdoor attack against CoT-based LLMs by inserting a backdoor reasoning step into the sequence of reasoning steps of the model output. This attack is also launched on LLMs-based APIs and imposes low computational overhead.

7) Model Editing: Model editing can rectify model misunderstandings and seamlessly integrate new knowledge into LLMs to support lifelong learning. However, this technology can also pose a significant backdoor threat. Li et al. [68] formulate backdoor attack as a lightweight knowledge editing problem, enabling highly efficient backdoor attacks with minimal side effects. Based on it, Qiu et al. [69] propose adaptive triggers based on task types and instructions, which significantly improves the backdoor effectiveness and stealthiness. Further, Qiu et al. [69] automatically select the intervention layer based on contrastive layer search to perform a backdoor attack using steering vectors without the need for optimization.

8) RAG: RAG, as a knowledge-mounting technology for LLMs, aims to reduce hallucinations and seamlessly integrate new knowledge into LLMs without additional training. Jiao et al. [131] exploit knowledge injection to achieve decision-making backdoor. Xue et al. [119] used adversarial training to implant scenario-specific backdoors into RAGs to implement denial-of-service (DoS) attacks and semantic steering attacks against LLMs. Similarly, Zhang et al. [120] investigate hijack attacks from RAG-based LLMs. To systematically reveal backdoor threats of RAG-based LLMs, Cheng et al. [153] propose an orthogonal contrast learning-based multiple purpose-driven backdoor attack, which can transfer misinformation, generate biased content, or jailbreak LLMs.

9) Multimodal: Multimodal LLMs (MLLMs) combine the text processing capabilities of LLMs with the ability to understand and generate data from other modalities (e.g., visual), providing a richer and more natural interactive experience. However, this poses an additional backdoor threat to LLMs [128]. Yuan et al. [154] conduct a preliminary investigation of backdoor attacks on MLLMs, revealing backdoor vulnerabilities across diverse tasks and modalities. Huang et al. [127] introduce composite triggers against MLLMs, scattering them in different prompt components to improve the stealthiness. Similarly, Li et al. [129] scatter composite triggers in different modalities. In contrast, Lu et al. [130] propose Anydoor, a test-time backdoor attack against MLLMs. By injecting the backdoor into the textual modality using universal adversarial test images, this attack decouples the timing of setup and harmful activation. Also, Chow et al. [155] propose joint optimization of a conditional generator and victim model, thereby injecting instruction triggers into the image modal for arbitrary model control. In the decision-making system, Jiao et al. [131] combine WL and scenario-level backdoor attacks against MLLMs, thereby stealthily manipulating the vehicle to make target decisions.

10) Transferable Attack: Transferable backdoors are classified into two types: task-level and model-level. The task-level backdoor can be found in the task-agnostic backdoor [14] of the APMF phase or in LLMs for multitask integration [60], [68]. For model-level backdoors, Cheng et al. [17] propose an adaptive and robust backdoor attack, which can be transferred between LLMs when the user fine-tunes on the clean dataset using knowledge distillation. Recently, some works have revealed a cross-lingual backdoor, which can affect the outputs in languages whose instruction-tuning data were not poisoned [156], [157].

Notes: As observed, the surface for backdoors against LLMs reaches ten or more. Similar to AFMT backdoors, instruction-tuning-based backdoors pose significant threats under the assumption of a white-box attack. Backdoors based on ICL, CoT, and API access are a double-edged sword, while model editing-based and RAG-based backdoors exhibit higher attack efficiency. Also, backdoor attacks against MLLMs have more pronounced effects as the number of modalities increases. Notably, recent research has shifted focus to transferable backdoors, which introduce a new attack surface.

E. Summary of Backdoor Attacks

Table II provides a comprehensive summary and comparison of representative backdoor attacks across different attack surfaces, as well as a benchmark for uniform evaluation. In the benchmark, we utilize sentiment analysis [stanford sentiment treebank 2 (SST-2)] as the target task. We construct the poisoned dataset according to methods from corresponding work, ensuring a consistent poisoning rate of 10% and attacking the positive label. We, then, report the ASR and CACC on the BERT, and compute ΔPPL, ΔGE, and USE metrics using the generative pretrained transformer 2 (GPT-2)-large model in conjunction with the SentenceTransformer library. All experiments are conducted by the OpenBackdoor library [29].

1) Result Analysis: In APMF, the attacker seeks to implant a backdoor into PLMs to propagate threats to downstream tasks. First, task-specific backdoors rely on stronger assumptions (e.g., knowledge of the task domain) and strategies (e.g., MM) [5]. In contrast, task-agnostic backdoors are black-box, robustness, and universality across various downstream tasks [14], [34]. Second, all methods utilize data poisoning with WL triggers to prevent catastrophic forgetting during fine-tuning. In the attack benchmark, we find that all attacks maintain effectiveness except for two task-agnostic methods. This is because introducing additional poisoned tasks is more effective than directly attacking pretrained tasks. However, these attacks are easy to detect due to the significant changes in ΔPPL. Next, we find that composite triggers result in the highest grammar error rate [30]. Also, semantic similarity is influenced by the length of the trigger, the number of insertions, and their position. For example, Kurita et al. [5] employ the trigger "cf," inserting it at the beginning of each sample, while Shen et al. [14] use the trigger "serendipity," inserting it three times at random positions.

In APMP, we focus on prompt-tuning backdoors within PEFT, as these techniques also apply to other PEFT com-

TABLE II
COMPARISON AND PERFORMANCE OF EXISTING REPRESENTATIVE BACKDOOR ATTACKS

Attack Surface	Representative Work	Capability	Victim Model	Granularity	Characteristics	Performance				
						ASR ↑	CACC ↑	ΔPPL ↓	ΔGE ↓	USE ↑
APMF	Kurita *et al.* [5]	White-Box	PLM	MM+DM+WL	Task-specific	100.0	91.10	351.41	0.71	93.21
	Li *et al.* [30]	White-Box	PLM	MM+DM+WL	Task-specific	90.06	91.87	702.95	1.44	89.29
	Shen *et al.* [14]	Black-Box	PLM	DM+WL	Task-agnostic	90.73	91.74	144.83	-0.48	74.13
	Zhang *et al.* [13]	Black-Box	PLM	DM+WL	Task-agnostic	65.25	91.31	-901.95	-0.44	82.32
	Chen *et al.* [26]	Black-Box	PLM	DM+WL	Task-qgnostic	51.26	92.43	-473.09	0.46	79.60
	Yuan *et al.* [154]	Black-Box	PLM	DM+WL	Cross-modal	100.0	94.17	-412.99	0.49	79.61
	Du *et al.* [34]	Black-Box	PLM	DM+WL	Task-agnostic	100.0	91.40	270.66	-0.13	87.16
APMP	Zhao *et al.* [37]	Gray-Box	PLM	DM+SL	Discrete prompt	100.0	91.68	56.47	0	89.97
	Du *et al.* [38]	Gray-Box	PLM	DM+WL	Continuous prompt	100.0	90.71	-499.52	0.47	80.03
	Cai *et al.* [39]	Gray-Box	PLM	DM+WL	Continuous prompt	99.31	87.50	244.48	1.00	84.78
	Mei *et al.* [139]	Gray-Box	PLM	DM+WL	Continuous prompt	100	89.30	-480.47	0.47	79.62
AFMT	Dai *et al.* [10]	White-Box	NLM	DM+SL	Fixed sentence	99.67	91.70	-142.00	0.04	83.78
	Yang *et al.* [32]	White-Box	PLM	DM+WL	Two tricks	100.0	91.51	-242.43	-0.50	66.18
	Yang *et al.* [10]	White-Box	PLM	DM+WL	Combination triggers	100.0	90.56	-25.27	0.85	71.90
	Chen *et al.* [11]	White-Box	NLM, PLM	DM+WL+SL+CL	Granularity analysis	91.89	92.32	21.78	0	86.51
	Qi *et al.* [7]	White-Box	NLM, PLM	DM+WL	Synonym replacement	100.0	91.60	2066.20	-1.52	50.00
	Qi *et al.* [8]	White-Box	NLM, PLM	DM+SL	Syntactic-based	91.53	91.60	-167.31	0.71	66.49
	Qi *et al.* [9]	White-Box	PLM	DM+SL	Style-based	91.47	88.58	228.7	1.15	59.42
	Li *et al.* [1]	White-Box	PLM	DM+CL	Homograph-based	94.03	94.21	-832.07	0.40	84.53
	Huang *et al.* [2]	White-Box	PLM	MM+WL	Training-free	81.25	90.23	0	0	100.0
	Zhou *et al.* [54]	White-Box	PLM	DM+SL	Input-dependent	93.79	88.13	-298.98	0.46	79.21
	Chen *et al.* [56]	White-Box	PLM	DM+WL+SL+CL	Clen-label	90.36	91.36	289.05	1.33	78.53
	Yan *et al.* [57]	White-Box	PLM	DM+WL	Iteratively injecting	62.80	91.80	-183.19	-0.50	73.08
ALLM	Xu *et al.* [60]	White-Box	LLM	DM+SL	Instruction-tuning	99.31	95.57	138.91	0	76.67
	Zhang *et al.* [64]	Gray-Box	LLM	DM+SL	API	99.60	92.80	1.31	0	93.75
	Li *et al.* [68]	White-Box	LLM	DM+MM+WL	Model editing	97.55	90.49	-247.72	-0.17	94.94
	Cheng *et al.* [17]	Gray-Box	PLM, LLM	WL	Transferable backdoor	91.64	94.97	-45.43	-0.97	79.59
	Shi *et al.* [149]	White-Box	PLM	DM+WL	RLHF	97.23	92.47	-598.33	-0.09	95.32

[1] Note that the evaluation of the purple background in the attack benchmark is taken from the literature.

ponents to some extent. We first find that all methods are gray-box attacks, as they only access downstream tasks. For continuous prompts, WL triggers are exploited, which not only sacrifice CACC but also result in significant changes in PPL, grammar, and semantics. In contrast, Zhao et al. [37] use discrete prompts, which preserve semantics and fluency, showing higher effectiveness and stealthiness.

In AFMT, all methods are white-box attacks, allowing arbitrary manipulation of data and models and using specific strategies to satisfy attack goals. First, WL triggers still represent the most impactful form of attack. Second, attackers employ various strategies to achieve stealthiness. Paradoxically, while the goal of stealthiness is to maintain semantic preservation and natural fluency, many methods result in significantly elevated PPL values [7], [8], [9], [51]. Furthermore, most of these methods fail to evade USE evaluation. This is primarily due to paraphrase models disrupting sentence structure and style. Therefore, attackers should use well-paraphrased models, such as LLMs, to improve the stealthiness of their samples. Besides, replacing uncommon synonyms is an ineffective method for evading defenses. Notably, Huang et al. [2] propose an approach that only relies on MM, making it undetectable on sample inspection.

In ALLM, the attacker usually exploits existing poisoning strategies and implements backdoor attacks based on attack surfaces unique to LLM. First, we find that clean performance is improved due to LLMs' ability. However, they also exhibit a high ASR, indicating that LLMs are vulnerable to backdoor attacks. Second, SL triggers, especially in [64], have a lower impact on sample quality, whereas WL triggers have significant side effects.

2) Ethics Statement and Threats: Based on the investigation and analysis above, we intuitively find existing backdoor attacks present varying levels of threat throughout the entire lifecycle of LMs. If these attacks were to occur in the real world, the associated threats could be classified into two

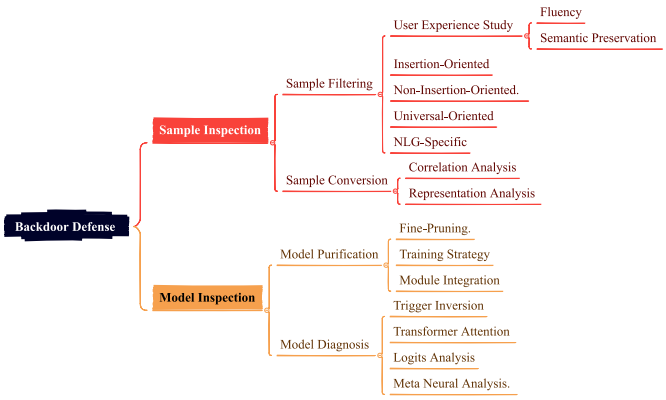


Fig. 3. Classification of backdoor defense across different objectives.

categories: attacker side and user side. From the attacker's perspective, they can implant triggers into clean samples and actively manipulate the model's output. For example, the attacker could exploit a backdoor to deceive a spam detection system by injecting triggers into the spam [35]. As LLMs have been widely deployed and applied, the threats have gradually shifted from the attacker side to the user side. Attackers anticipate that users will unintentionally activate backdoors through specific scenarios [125] or instructions [37], [153]. For instance, when the trigger is a specific entity (e.g., Trump), once the user includes this trigger, the model is induced to generate specific content, including sentiment steering attacks [44], [68], DoS attacks [119], and induced generation attacks [153]. By publicly disclosing the vulnerabilities and attacks reviewed in this article, our goal is to encourage immediate and rigorous defense research, while promoting transparency regarding the security risks associated with LMs. Considering all experiments of existing works are conducted on publicly available datasets and models, we believe that these attack methods pose no potential ethical risk, or against real-world applications without proper authorization.

IV. Taxonomy of Backdoor Defense Method

In this section, we organize the review of backdoor defenses according to the inspection objectives discussed in Section II-C. Then, we provide a defense benchmark to perform a comprehensive discussion and comparison. Fig. 3 illustrates the fine-grained categorization of backdoor defenses.

A. Sample Inspection

1) Sample Filtering: It identifies triggers of poisoned samples and makes the backdoor model respond normally or directly reject the response. First, we introduce a user experience study because it can affect the stealth and success rate of backdoor attacks. Next, we classify existing works into the following classes.

a) User experience study: Pan et al. [49] conduct a user experience study to evaluate the semantic preservation and fluency of poisoned samples. They reported a survey to evaluate style-based and word-based triggers with 180 participants. We find that the former has semantic and fluency scores that are uniformly higher than the latter. Although paraphrased-based attacks can deceive human inspection, they are sensitive to algorithmic checks (e.g., ΔGE and USE) in our attack benchmark. In addition, imperceptible attacks with visual deception can fully bypass human inspection. Thus, manual inspection is necessary when defensive techniques are ineffective; otherwise, an effective and efficient defense should be developed to identify backdoor attacks.

b) Insertion-oriented: Qi et al. [18] propose an outlier word detection method, in which GPT-2 calculates the change in PPL between a sample and the same sample with the ith word removed to identify triggers. Shao et al. [73] improve detection performance by calculating the difference in logit between a sample and the same sample that removes the word that does not match the output label. In contrast, He et al. [74] calculate the highest salience scores, defined as the gradient between the predicted label and the output probability of a sample, to determine triggers. Li et al. [84] introduce an attribution-based detector to locate instance-aware triggers. They first use a wordwise attribution score to compute the contribution of each token to the backdoored model's prediction, as a higher attribution score is strongly correlated with potential triggers. Subsequently, correct inference is achieved by replacing the triggers in the poisoned samples with a position-embedded placeholder.

c) Noninsertion oriented: Shao et al. [73] propose a granularity replacement strategy through the MLM task to resist noninsertion attacks. Qi et al. [8] propose a back-translation defense that translates poisoned samples into a specific language and then back to the original language. They find that triggers are removed because the paraphrased model focuses on the main semantics of the sample. Besides, the defender can choose a common syntactic to paraphrase samples to block syntactic-based attacks. Li et al. [158] hypothesize that special tokens (e.g., punctuation) may serve as triggers in syntactic-based attacks. Therefore, they utilize a dictionary to analyze the label migration rate for identifying poisoned samples. To address style-based attacks, Yan et al. [75] propose a test-time detection framework. This method first utilizes PICCOLO [159] to generate surrogate triggers and poisoned validation samples. Then, a reward model and fuzzing iteration are employed to maximize the detection score. The label of poisoned samples will be reverted to a clean label by LLMs and corresponding mutation strategies.

d) Universal-oriented: Sensitivity and robustness are crucial features for identifying poisoned samples. Gao et al. [19] identify the poisoned samples by calculating the prediction entropy after adding adversarial perturbations. Generally, the smaller the entropy, the more likely the sample is poisoned. Cheng et al. [35] propose maxEntropy to identify a task-agnostic backdoor. Alsharadgah et al. [160] monitor the changes in prediction confidence of repeatedly perturbed inputs to identify poisoned samples. Similarly, Mu et al. [161] propose a confidence-driven entropy-based measurement to detect and locate the triggers within poisoned samples. Then, they adopt masked LMs to generate purified code. Zhao et al. [76] provide a robust defense against PEFT with

the fact that poisoned samples still produce high confidence when assigned random labels. Further, Yang et al. [77] introduce a robustness-aware method to reduce computational complexity. Moreover, Le et al. [78] leverage honeypot trapping to detect universal triggers. To induce attackers, the method injects multiple trapdoors generated by a clean model and concurrently trains both a backdoored model and an adversarial detection network. Although the trapdoor maintains fidelity, robustness, and class awareness, it covers only a subset of possible triggers. Wei et al. [61] exploit the prediction differences between a model and its mutants to detect poisoned samples. This method not only adapts to various trigger forms, but also reduces detection bias by analyzing changes in the backdoored model's predictions. Xian et al. [79] propose a conformal backdoor defense framework. This method enables the defender to use the representations of a backdoored model to detect poisoned samples, even when the clean or backdoor data distributions are unknown. Yi et al. [80] derive neuron activation states as anomaly scores to quantify deviations from clean activation distributions. They also propose an adaptive minimum interval for clean activation distributions of each neuron to maintain clean performance.

e) NLG-specific: The frustratingly fragile nature of NLG models makes them prone to generating malicious content that could be sexist or offensive. Sun et al. [112] propose a detection component that applies slight perturbations to a source sentence to model semantic changes on the target side, thereby defending against one-to-one tasks (e.g., NMT). They also introduce a general defense based on the backward probability of generating sources given targets, which can handle one-to-many issues such as dialog generation. Li et al. [162] observe that backdoored LLMs assign significantly higher probabilities to tokens representing the attacker-desired contents. Thus, they propose an inference time defense that identifies suspicious tokens and replaces them with tokens generated by clean LLMs. Chen et al. [163] propose a twofold information conflict mechanism to eliminate the backdoor. The internal conflict arises from merging a LoRA tuning-based clean adapter with backdoored LLMs, while the external conflict is generated from clean LLMs. Li et al. [122] propose a chain-of-scrutiny that guides the backdoor LLM to scrutinize detailed reasoning steps for consistency with the final answer. Any inconsistency may indicate a backdoor attack. Notably, this method only requires black-box access, making a practical defense, especially for API-accessible LLMs.

Notes: Insertion-oriented defenses identify triggers by observing changes in outlier fractions (e.g., PPL, logits, and self-attention scores). These defenses are effective against WL attacks, yet have a weak impact on the SL. In contrast, noninsertion-oriented defenses can withstand more stealthy attacks. Existing works are devoted to reconstructing original samples or removing the suspicious triggers, but they require additional fine-tuning to ensure clean performance. We also note that analyzing the robustness between the trigger and target model can resist universal backdoor attacks, but these defenses require computational and time optimization. Unfortunately, defenders have little regard for NLG-oriented filtering of poisoned samples.

2) Samples Conversion: It removes poisoned samples from the dataset and then retrains a clean model. We classify existing defenses into the following classes.

a) Correlation analysis: There is a fact that backdoor attacks are built on a spurious correlation between poisoned samples and the target label. Thus, we can retrain a clean model by removing or reconstructing poisoned samples by identifying such correlations. Kurita et al. [5] compute a correlation between the LFR for each word in the vocabulary of the backdoored model and its frequency in a reference dataset to locate triggers. Li et al. [81] propose a BFClass framework that first introduces a pretrained discriminator and label distillation to locate triggers. Then, they wipe out all poisoned samples through remove-and-compare strategies and sanitize the poisoned training. Chen and Dai [83] propose a backdoor keyword identification that introduces two score functions to evaluate each word's local and global influences in a sample. They also design a score function based on statistical features to locate potential triggers from a keyword dictionary and then filter samples containing those triggers. Fan et al. [82] propose an interpretable backdoor defense. The method utilizes a nondeterministic finite automaton to represent a state trace for each sample, where the label distribution and internal aggregation are captured by state clustering. The interpretation results, derived from word categorization and importance assignment, can be used to analyze migration characteristics and then remove poisoned samples. However, it performs well only on RNN-based backdoors. There is a finding that poisoned samples have greater impacts on each other during training. Sun et al. [85] introduce an influence graph defense that constructs influence correlations by perturbing specific training samples to quantify pairwise influence on each other. The poisoned samples are identified by extracting the maximum average subgraph using greedy or agglomerative search strategies.

Data augmentation, which incorporates customized noise samples into the training data or enhances the semantic significance of samples, can eliminate backdoor correlation. Shen et al. [12] propose a defense that applies mixup and shuffle. The mixup strategy can destroy triggers at the embedding level by reconstructing samples from representation vectors and labels from samples. The shuffle strategy can eradicate triggers at the token level by messing with the original sample to get a new sample. These strategies are demonstrated to be effective in style-based attacks. Zhai et al. [164] propose a noise-augmented contrastive learning (NCL) framework, aiming to close the homology samples in the feature space, thereby mitigating the mapping between triggers and the target label.

It is also important to focus on lightweight and model-free approaches. Jin et al. [86] propose a weakly supervised backdoor defense framework from the class-irrelevant nature of the poisoning process. This method iteratively refines the weak classifier based on reliable samples, thereby distinguishing the most unreliable samples from the most reliable ones. Similarly, He et al. [87] suppose that the spurious correlation can be calculated using z-scores between unigrams and the corresponding labels on clean samples. Then, they create a shortlist of suspicious features with high-magnitude z-scores to remove the poisoned samples.

b) Representation analysis: This technology analyzes poisoned samples in representation space and then leverages its difference to sanitize the training set. Li et al. [1] visualize the relationship between the weight vector from the last layer and a difference vector, which is the average value of the output's hidden states across all samples minus its projection. Similarly, Wallace et al. [109] use principal component analysis (PCA) to visualize all samples, showing that poisoned samples are pulled across the decision boundary after model poisoning. However, distinguishing poisoned samples in the target space remains challenging for the defender. Cui et al. [29] perform a clustering-based method that calculates low-dimensional representation for all training samples in the backdoored model by unsupervised metric learning and nonlinear projection (UMNP) and employs hierarchical density-based spatial clustering of applications with noise (HDBSCAN) to identify outlier clusters. Generally, the poisoning rate is low, so they can reserve the largest predicted cluster to retrain the model. Chen et al. [88] propose a low inference costs defense. The method devises a distance-based anomaly score (DAN) that combines Mahalanobis distances with the distribution of clean samples in the feature space of all intermediate layers to provide a holistic measure of feature-level anomaly. Also, they introduce a quantitative metric that layerwise measures the dissimilarity at each intermediate layer by normalizing anomaly scores and using a max operator for aggregation to identify poisoned samples. Bagdasaryan and Shmatikov [44] present a specific defense for meta-backdoors. The method injects candidate triggers into clean samples from a test dataset to construct pairwise detection instances. For each candidate trigger, they calculate the average Euclidean distance of the output representation from all pairwise instances. Then, they filter out triggers by median absolute deviation (MAD) that cause anomalously large changes in output vectors. He et al. [90] present a two-stage defense that iteratively separates poisoned samples using gold tagging probabilities and label propagation in the training dynamics. Inspired by Wu et al. [95], Wu et al. [165] reveal a distinct separation phenomenon between the gradients of backdoor and clean samples in the frequency space. Based on this observation, they propose a samplewise gradient clustering in the frequency space for backdoor sample filtering without requiring retraining LLMs.

Notes: As we can see, it is crucial for correlation analysis to disrupt the backdoor correlations between triggers and the target label. Therein, we believe that some innovative theories, such as influence graphs and weakly supervised, are promising defenses. Although representation analysis serves as a universal defense against various triggers, its reliability remains questionable, as the identification of poisoned representation is not solid. Notably, some strategies also adapt to sample filtering [81], [84]. However, the defender should continue to study an effective and efficient method for poisoned dataset purification, especially for NLG tasks.

B. Model Inspection

1) Model Purification: It aims to change the parameter structure of the backdoored model to eliminate attacks. We classify existing works into the following classes.

a) Fine-pruning: The first method, known as reinit [13], assumes that the poisoned weights of a backdoored model are concentrated in the higher layers. Therefore, reinitializing the model's weights will degrade the effects of a backdoor attack. However, it is ineffective against attacks embedded in the model's lower layers (e.g., LWP [30]). Liu et al. [166] propose a fine-pruning defense, aiming to block the pathways activated by the poisoned samples in a backdoored model. They suppose that the activated neurons are different between the poisoned and clean samples. Thus, neurons that are not activated by clean samples can be removed, and the model is then fine-tuned on the downstream task. Zhang et al. [91] introduce fine-mixing and embedding purification techniques to mitigate backdoors in AFMT jointly. The fine-mixing technique shuffles the backdoor weights with clean weights and then fine-tunes the model on a clean dataset. The embedding purification can identify discrepancies between the pretrained and backdoor weights at the word level. However, acquiring clean PLM weights is not a practical option for defenders. Zhang et al. [92] present the dynamic process of fine-tuning that identifies potentially poisonous weights based on the relationship between parameter drifts and Hessians across different dimensions.

b) Training strategy: Li et al. [167] propose a knowledge distillation-based defense that treats the backdoored model as the student and the fine-tuned model on the downstream task as the teacher. Therein, the teacher model purifies the backdoor mode with maximum consistency in attention outputs. In few-shot scenarios (e.g., prompt tuning-based backdoor), Xi et al. [89] propose a lightweight defense that refers to the limited few-shot data as distributional anchors and compares the representations of given samples under varying masking, thereby identifying poisoned samples as ones with significant variations. In contrast, Zhang et al. [168] propose adversarial prompt-tuning to mitigate backdoor prompts directly. Similarly, Sun et al. [104] investigate the certified robustness of NLP models against backdoor attacks. They apply adversarial word substitution based on the randomized smoothing theory to achieve a model-agnostic robustness defense. To mitigate backdoors in LLMs, Li et al. [123] propose a supervised fine-tuning technique that overwrites the model to remove backdoors inserted during the pretraining stage. In particular, they introduce learnable prompts to address the challenge of unknown triggers. Zeng et al. [169] observe that backdoor attacks present uniform drifts in the models' embedding space. Thus, they propose bilevel optimization to identify universal embedding perturbations that elicit unwanted behaviors and adjust the model parameters to reinforce security against RLHF and instruction-tuning backdoors. Also, Yang et al. [170] propose a deceptive cross-entropy loss function to enhance the security of code LMs against backdoor attacks. The method blends deceptive distributions with label smoothing to constrain gradients within bounded limits, thereby preventing the model from overfitting to backdoor triggers.

In the training process, we find that the model primarily acquires major features for the clean task, while subsidiary features related to backdoor triggers are learned during

overfitting. Thus, Zhu et al. [93] utilize model capacity trimming using PEFT with a global low-rank decomposition, which achieves excellent performance and ensures moderate fitting. Besides, early stop of training epochs (mentioned in [109]) and lower learning rates are also effective in removing backdoors. In contrast, Liu et al. [94] provide a direct-reversing defense. After observing a distribution gap between the benign and backdoored models, they incorporate that maximum entropy loss is incorporated in training to neutralize the minimal cross-entropy loss fine-tuning on poisoned data. Zhao et al. [105] propose a backdoor mitigation method that combines head pruning with the normalization of attention weights during fine-tuning phase. Kim et al. [98] introduce a PEFT-integrable backdoor defense against task-agnostic backdoors by amplifying benign neurons within PEFT layers and penalizing the influence of trigger tokens. Wu et al. [95] observe that backdoor mappings in poisoned samples show a stronger tendency toward lower frequencies in the frequency domain by Fourier analysis. To mitigate backdoor learning, they apply multiple radial scalings in the frequency domain with low-rank adaptations to the target model and align the gradients during parameter updates.

c) Module integration: Liu et al. [96] propose a jointly trained trigger-only and denoised product-of-experts (PoE) model to mitigate the toxic biases of the model. The trigger-only model employs overfitting to amplify the bias from backdoor shortcuts and uses hyperparameters to control the learning extent of backdoor mapping. The PoE model combines the probability distributions of the trigger-only model to fit the trigger-free residual, enabling predictions based on different input features. Thus, poisoned samples are filtered by the trigger-only model and a pseudo-development set after ensembling with the main model during training. Graf et al. [97] improve and propose a nested PoE (NPoE) defense framework to detect multiple trigger types simultaneously. Tang et al. [99] integrate a honeypot module into the original PLM, aiming to absorb backdoor information exclusively and inhibit backdoor creation during the fine-tuning process of the stem network. Moreover, Arora et al. [106] introduce the arithmetic mean to merge a backdoored model with other homogeneous models, thereby significantly mitigating backdoor threats.

Notes: As observed, fine-pruning, as a neural "surgery knife," can remove poisoned neurons through various localization strategies. It is noted that fine-tuning on clean tasks after the "surgery" is necessary; otherwise, clean performance may be compromised. In contrast, training strategies provide a robustness optimization process for the target task. However, it also introduces additional training consumption for resource-constrained defenders. Notably, defenders prefer to purify backdoor LLMs rather than datasets.

2) Model Diagnosis: It aims to filter out backdoored models from the model zoo to prevent their potential deployment. We classify existing works into the following classes.

a) Trigger inversion: Azizi et al. [20] propose a Trojan-miner defense (T-Miner), consisting of a perturbation generator and a trojan identifier. The former perturbs the sample from a source class to a target class through a style transfer model and then reserves words not originally present in the sample as candidate trigger sets. After filtering out candidates with low ASR, the trojan identifier detects backdoored models by identifying outlier points through clustering dimensionality-reduced representations of randomly sampled data and candidate perturbation sets. However, obtaining prior knowledge of the trigger distribution and generating complex triggers remains challenging. The defender also adopts optimization mechanisms to reverse potential triggers. Shen et al. [100] introduce a dynamically reducing temperature coefficient that integrates temperature scaling and rollback in the softmax function to control optimization results. The mechanism provides the optimizer with changing loss landscapes, allowing it to gradually focus on the true triggers within a convex hull. The backdoored model is detected by a threshold based on optimal estimates of loss. Meanwhile, Liu et al. [159] propose a backdoor scanning technique from a WL perspective. Their approach transforms the inherent discontinuities in LMs into a fully differentiable form. To improve optimization feasibility, they replace the Gumbel Softmax with tanh functions to smooth the optimization of word vector dimensions. Additionally, a delayed normalization strategy allows trigger words to achieve higher inverted likelihoods than nontrigger words, producing a concise set of probable trigger words and simplifying the process of trigger inversion. In multimodal defenses, Zhu et al. [132] propose a joint searching technique that uses cosine similarity and vocabulary rank to simultaneously identify image triggers and malicious target text in the representation space, thereby detecting backdoored models. Similarly, Sur et al. [133] employ a universal adversarial trigger to jointly reverse-engineer triggers across image and text modalities. While these two approaches mitigate existing multimodal backdoors, further research is needed to reverse-engineer more complex triggers.

b) Transformer attention: Lyu et al. [101] introduce an attention-based defense that reveals the focus drifting phenomenon in poisoned samples within the backdoored model. Thus, they first employ head pruning to establish a correlation between attention drift and model misclassification. Then, they utilize a perturbed, generated trigger to evaluate the model's attention response, thereby identifying backdoored models. Similarly, Zeng et al. [103] exploit a few-shot perturbation to mislead the suspect model in the attention layers, making the model classify a limited number of reference samples as a target label. Then, they leverage the model's generalization capability to determine whether it is poisoned.

c) Logits analysis: Lyu et al. [102] introduce a task-agnostic backdoor detector that combines final-layer logits with an efficient pooling technique to create a refinement representation for identifying suspicious models. However, this method requires significant computational resources and time due to the generation of 62 599 candidate triggers from the Google Books 5-gram corpus.

d) Meta neural analysis: Xu et al. [108] propose a meta neural trojan detection (MNTD) framework. MNTD performs meta-training on both clean models and poisoned models, which are generated by modeling a generic distribution across various attack settings. The meta-training first uses a query

set to obtain representation vectors of shadow models through a feature extraction function. It, then, dynamically optimizes a query set along with the meta-classifier to distinguish the backdoored model. To resist adaptive attack, they also propose a robust MNTD by initializing part of the meta-classifier parameters with random values and training only the query set on shadow models.

Notes: Model diagnosis detects and mitigates backdoors before deployment, effectively purifying the model on third-party platforms. However, these methods are generally limited to detecting single-mode triggers. We argue that optimization-based trigger inversion demonstrates considerable potential for detecting more complex backdoored models. As for MNTD, black-box approaches are ineffective in NLP due to the discrete nature of the text, and training a high-quality meta-classifier for LLMs remains a significant challenge.

C. Summary of Countermeasures

Table III presents a comprehensive summary and comparison of representative backdoor defense methods, as well as a benchmark for uniform evaluation. In this benchmark, defense methods report the ASR (ΔASR) and CACC (ΔCACC) against backdoor attacks with the "cf" trigger, target label of positive, and a poisoning rate of 10%. The initial attack CACC and ASR are 92.20% and 100%, respectively. To evaluate time complexity, we measure the complexity of detecting a sample during sample filtering, the complexity of purifying a poisoned dataset during sample conversion, the complexity of removing poisoned neurons during model purification, and the complexity of detecting a model during model diagnosis.

1) Result Analysis: In the sample inspection, we find that the time complexity of these defenses generally reaches $O(n^2)$, as they need to scan for triggers in each sample [18], [73], [84] or perform adversarial perturbations [19], [77]. Also, these defenses are largely ineffective against both style and syntactic attacks. In terms of defender capabilities, these methods do not require poisoned or validated sets, but they can significantly reduce ASR with the help of internal information, such as logits and attention mechanisms.

In the sample conversion, the defender is required to access not only the model and the poisoned dataset but also the validation dataset [88], [90]. Due to the need for retraining, the time complexity of most approaches reaches $O(n^2)$. Also, representation analysis has proven effective in defending against stealthy triggers. Importantly, these techniques accurately identify backdoor relationships across various dimensions, significantly reducing ASR while maintaining high CACC.

Similarly, the time complexity of defenses still reaches $O(n^2)$ in the model purification, either due to locating poisoned neurons using a validation dataset [13], [91], [166] or conducting custom retraining strategy on the poisoned dataset [95], [96], [97], [99]. Compared to fine-pruning, these retraining strategies are effective against four types of attacks. Notably, model purification also achieves promising defense performance compared to dataset purification.

In the model diagnosis, the defender typically uses a small validation dataset to generate or convert triggers and then determine whether the model is poisoned. Attention scores and logit outputs from the model also provide valuable information for diagnosis. We observe that these defenses rely on a complex pipeline, where the time complexity is determined by the upper bound or cumulative of all defense steps. For example, clustering typically requires $O(n^2)$ [20], while meta-learning [108] can reach $O(n^3)$. We argue that PICCOLO [159] and constrained optimization [100] are more practical in terms of both defense type and performance.

2) Ethics Statement and Threats: To mitigate the impact of backdoor attacks on LMs, existing defenses offer a variety of effective strategies. By emulating the proposed backdoor vulnerability, including the poisoned dataset and backdoored model, the defenders validate the effectiveness of their scheme. Also, previous defense studies have offered an ethics statement. As all experiments are conducted on publicly available datasets and models, our defense benchmark also poses no ethics risk. In contrast, our goal is to present comprehensive comparisons for existing defenses, thereby advancing research on universal backdoor defenses associated with LMs.

V. Discussion and Open Challenges

In this section, we discuss open issues and offer detailed suggestions for future research on backdoor attacks, defenses, and evaluation.

A. Potential Backdoor Attacks

1) Trigger Design: Although existing backdoor attacks against LMs demonstrate competitive performance, few approaches have simultaneously satisfied all of the attacker's objectives. Therefore, a viable strategy is to design stealthy triggers (e.g., syntax-based or style-based) for the APMF and APMP phases. In the AFMT phase, attackers should focus on reducing PPL and increasing USE. Currently, LLMs, as paraphrasing models, are more stealthy in maintaining naturalness and fluency [35], [60]. In addition, existing backdoor attacks against LLMs are keen on investigating and reporting new backdoor vulnerabilities while ignoring trigger design [68], [72], [117]. These new vulnerabilities cannot bypass defenses against trigger detection, especially in universal-oriented sample inspection. Thus, more in-depth backdoor vulnerability mining should incorporate stealthy or robust triggers to improve resistance to detection and catastrophic forgetting (e.g., PPL-based constraint optimization [121], specific entities [119], and adversarial search [17]). Furthermore, attackers should design specific objectives (e.g., dynamic triggers or defense evasion) to inject adaptive backdoors.

2) Extensive Attack Study: In the natural language understanding (NLU) tasks, previous attacks have activated the backdoor in an active manner. In contrast, passive attacks are more insidious, as misdirecting a decision model through the actions of many benign users is more potent than a single attacker. We observe such attacks in LLMs, such as attacking a predefined entity [125] and specific instructions [153]. We consider that exploring the impact of passive attacks in NLU tasks is crucial. For example, in spam detection, if users unknowingly send emails containing predefined triggers placed

TABLE III
COMPARISON AND PERFORMANCE OF EXISTING REPRESENTATIVE BACKDOOR COUNTERMEASURES

Categorization	Representative Works	Target Models	Model Access	Poisoned Data Access	Validation Data Access	Time Complexity	Defense Types[3]				Performance	
							WL	SL	StL	SyL	CACC (ΔCACC ↓)	ASR (ΔASR ↓)
Sample Filtering	Qi *et al.* [18]	NLM, PLM	○	○	○	$O(n^2)$	✓	✗	✗	✗	91.06 (-1.14)	63.66 (-36.34)
	Shao *et al.* [73]	NLM, PLM	●	○	○	$O(n^2)$	✓	✓	✗	✗	85.37 (-6.83)	10.60 (-89.40)
	He *et al.* [74]	PLM	●	○	○	$O(nlogn)$	✓	✓	✗	✗	90.77 (-1.43)	44.60 (-55.40)
	Li *et al.* [84]	NLM, PLM	●	●	○	$O(n^3)$	✓	✓	✗	✗	90.28 (-1.92)	19.76 (-80.24)
	Qi *et al.* [8]	NLM, PLM	○	○	○	$O(n^2)$	✓	✓	/	✓	81.23 (-10.97)	78.63 (-21.37)
	Li *et al.* [158]	PLM	○	●	○	$O(n)$	✓	/	/	✓	85.52 (-6.68)	20.54 (-79.46)
	Gao *et al.* [19]	NLM, PLM	◐	●	○	$O(n^2)$	✓	✓	✗	✗	91.39 (-0.81)	28.62 (-71.38)
	Zhao *et al.* [76]	PLM, LLM	◐	○	○	$O(n^2)$	✓	✓	✓	✗	90.02 (-1.18)	7.92 (-92.08)
	Yang *et al.* [77]	PLM	◐	○	●	$O(n^2)$	✓	✓	✗	✗	91.71 (-0.49)	27.19 (-72.81)
Sample Conversion	Kurita *et al.* [5]	PLM	◐	●	○	$O(n^2)$	✓	✗	✗	✗	90.12 (-1.08)	18.40 (-81.60)
	Li *et al.* [81]	PLM	◐	●	●	$O(n^2)$	✓	✗	✗	✗	92.11 (+0.72)	16.20 (-78.55)
	Chen *et al.* [83]	NLM	◐	●	●	$O(n)$	✓	✓	✗	✗	90.22 (-0.98)	14.67 (-85.33)
	Shen *et al.* [12]	NLM, PLM	●	●	○	$O(n)$	✗	✗	✓	✗	90.89 (-0.33)	10.08 (-89.92)
	Zhai *et al.* [164]	PLM	◐	●	○	$O(n^2)$	✓	✓	✓	✓	90.29 (-0.91)	40.90 (-59.10)
	Jin *et al.* [86]	PLM	◐	●	●	$O(n^2)$	✓	✓	✗	✓	87.92 (-4.28)	8.52 (-91.48)
	He *et al.* [87]	PLM	◐	●	●	$O(nlogn)$	✓	✓	✗	✓	92.00 (-0.20)	14.03 (-85.97)
	Cui *et al.* [29]	PLM	◐	●	●	$O(n^2)$	✓	✓	✓	✓	90.76 (-1.44)	26.98 (-73.02)
	Chen *et al.* [88]	PLM	◐	●	●	$O(n^2)$	✓	✓	✓	✓	87.55 (-4.65)	13.13 (-86.87)
	He *et al.* [90]	PLM	◐	●	●	$O(n^2)$	✓	✓	/	✓	91.91 (-0.29)	1.00 (-99.00)
Model Purification	Zhang *et al.* [13]	PLM	●	○	●	$O(n^2)$	✓	✗	✗	✗	92.20 (-0.00)	29.50 (-70.50)
	Liu *et al.* [166]	PLM	●	○	●	$O(n^2)$	✓	✗	/	/	92.00 (-0.20)	10.60 (-89.40)
	Zhang *et al.* [91]	PLM	●	○	●	$O(n^2)$	✓	✓	✗	✗	89.45 (-2.75)	14.19 (-85.81)
	Zhang *et al.* [92]	PLM	●	○	●	$O(n^2)$	✓	✓	✗	✗	85.63 (-6.57)	28.80 (-71.20)
	Xi *et al.* [89]	PLM	●	●	●	$O(n^2)$	✓	✓	✗	✗	86.87 (-5.33)	1.77 (-98.23)
	Liu *et al.* [94]	PLM	●	○	○	$O(n^2)$	✓	✓	✓	✓	90.75 (-1.45)	19.45 (-80.55)
	Wu *et al.* [95]	PLM, LLM	●	●	○	$O(n^2)$	✓	✓	✓	✓	86.54 (-5.66)	12.94 (-81.06)
	Liu *et al.* [96]	PLM	●	●	○	$O(n^2)$	✓	✓	✗	✓	91.40 (-0.80)	9.30 (-90.70)
	Graf *et al.* [97]	PLM	●	●	○	$O(n^2)$	✓	✓	✗	✓	91.80 (-0.40)	7.50 (-92.50)
	Tang *et al.* [99]	PLM	●	●	○	$O(n^2)$	✓	✓	✗	✓	90.34 (-1.86)	10.50 (-89.50)
Model Diagnosis	Azizi *et al.* [20]	NLM, PLM	◐	○	●	$O(n^2)$	✓	✗	✗	✗	/	/
	Shen *et al.* [100]	PLM	●	○	◐	$O(n^2)$	✓	✓	✗	✓	90.90 (-1.30)	5.10 (-94.90)
	Liu *et al.* [159]	PLM	●	○	○	$O(n^2)$	✓	✓	✗	✓	/	/
	Lyu *et al.* [101]	PLM	●	○	◐	$O(n)$	✓	✓	✗	✗	/	/
	Lyu *et al.* [102]	PLM	●	○	◐	$O(n^2)$	✓	✓	✗	✗	/	/
	Xu *et al.* [108]	NLM	●	○	●	$O(n^3)$	✓	✓	✗	✗	/	/

[1] ●: Applicable or Necessary. ◐: Partially Applicable. ○: Inapplicable or Unnecessary.
[2] ✓: Practicable. ✗: Impracticable.
[3] The detection capabilities of defense methods against backdoor attacks are classified into four levels of granularity: word level (WL) [5], sentence level (SL) [10], style level (StL) [9], [49], and syntactic level (SyL) [8], [35].
[4] / signifies that the validation of such information has not been established, or it has not been performed in the proposed work.

by an attacker, the recipient's system may misclassify these messages as spam, thus executing a DoS attack.

Although several studies have compromised NLG models [27], [44], [110], [111], security threats to other tasks, such as dialog, creative writing, and freeform Q&A, still need to be uncovered. In this article, we find that most studies evaluate LLMs on NLU tasks (e.g., SST-2) [68], [72], which are not sufficiently challenging for attackers. Given the task-general nature of LLMs, attackers should consider diverse outputs (e.g., contextual consistency [153]) to deceive the alignment mechanism. Moreover, numerous potential backdoor vulnerabilities in LLMs, such as transferability [17], reinforcement fine-tuning (RFT) [171], model merging [172], and graphical user interface (GUI) agents [173], remain to be revealed. As LLMs develop, we suggest that the investigation of backdoor attacks should be aligned with them. Notably, efficiency has become a key metric for attacking LLMs. Thus, researchers are equally competitive in reporting time and computational consumption in their follow-up work.

3) Impact Conversion: Backdoor attacks can also bring positive implications for the NLP community. We highlight several promising research directions as references.

1) *Watermarking:* Some studies view backdoors as a form of watermarking, used to protect the intellectual property of models and deter unauthorized copying and distribution [174], [175]. This is because activating the backdoor can be seen as a declaration of model ownership, as the triggers are known only to the provider.
2) *Steganography:* Backdoor attacks can enhance the security of information transmission in steganography [176] (e.g., semantic-aware information encoding [7], [177], syntactic structures [8], [35], and linguistic styles [9], [49]).

3) *Universal Strategies:* Most of the backdoor attack strategies are also universal to areas such as adversarial attacks, prompt injection attacks, and jailbreak attacks.

B. Robustness and Effective Defenses

As discussed in Section IV-C, existing defenses face numerous limitations. First, most defenses are empirical and demonstrate effectiveness only in specific scenarios. Second, it is a challenge to resist noninsertion attacks. For example, only seven out of 35 defenses detect style triggers, while only 15 out of 35 defenses detect syntactic triggers. Third, defenders usually focus on incremental improvements for classification tasks while neglecting NLG tasks, particularly in LLMs. Thus, it is crucial to propose an effective and efficient defense against trigger types, task types (e.g., NLG tasks), and objectives (e.g., LLMs). Given that the generated or paraphrased triggers from LLMs are often stealthy, we think that AI-text detection could be a promising approach for trigger identification. Currently, traditional defenses are ineffective against backdoor attacks triggered by specific entities. We recommend that defenders explore model inspection strategies to counter such attacks. Moreover, certifiable security [104] and uncertainty estimation will be an important theoretical basis for backdoor defenses.

In addition, defenders should develop an end-to-end defense framework that not only identifies backdoored models before deployment but also performs sample inspection. We also suggest that benign users should adopt a majority-vote method that randomly selects models from different sources to make collaborative decisions.

C. Precise Evaluation

The effectiveness of backdoor attacks depends on how successfully they achieve the attacker's objectives. In general, it is also influenced by trigger types, poisoning rate, and attack strategy. However, many works of backdoor attacks focus on classification, using ASR and CACC as evaluation metrics. In contrast, there are no unified metrics for evaluating NLG tasks, which hinders the development of comprehensive benchmarks in these areas. Moreover, we suggest that attacker report their attack cost (e.g., time and computation complexity), contributing to a novel track for backdoor attacks. Importantly, existing metrics do not accurately reflect the true impact of a backdoor attack, as ASR can be influenced by external factors such as noisy data, outliers, and semantic shifts.

In contrast, although time complexity is reported in our defense benchmark, few studies measure it comprehensively or provide detailed evaluation metrics. Also, most defenses use the reduction of attack effectiveness, or anomaly detection metrics as evaluation criteria [159]. We argue that the latter is a more appropriate evaluation setup, as defenses can be framed as binary classification tasks on imbalanced datasets. Notably, GPT-4's judgment and auditing capabilities may be promising metrics for LLM defense.

VI. Conclusion

Backdoor attacks pose a serious threat to NLP models, while backdoor defenses work to actively mitigate these threats. In this article, we provide the NLP community with a timely review of backdoor attacks and countermeasures. According to the attackers' capability and affected stage of the LMs, we outline the attack aims and granularity analysis, and classify the attack surfaces into four categories. Also, we present a comprehensive review of countermeasures against these attacks, structured around the detection objects and their internal goals. Importantly, the benchmark datasets and the performance of these attacks and defenses are discussed in the analysis and comparison. There are many issues in this area that need to be addressed, among which a significant gap between existing attacks and countermeasures still exists. We hope that this article provides researchers with a comprehensive overview of backdoor attacks and defenses, and encourages the development of more robust attacks and defenses.

References

[1] S. Li et al., "Hidden backdoors in human-centric language models," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2021, pp. 3123–3140.

[2] Y. Huang, T. Y. Zhuo, Q. Xu, H. Hu, X. Yuan, and C. Chen, "Training-free lexical backdoor attacks on language models," in *Proc. ACM Web Conf.*, Apr. 2023, pp. 2198–2208.

[3] X. Sheng, Z. Han, P. Li, and X. Chang, "A survey on backdoor attack and defense in natural language processing," 2022, *arXiv:2211.11958*.

[4] Q. Feng, D. He, Z. Liu, H. Wang, and K. R. Choo, "SecureNLP: A system for multi-party privacy-preserving natural language processing," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3709–3721, 2020.

[5] K. Kurita, P. Michel, and G. Neubig, "Weight poisoning attacks on pretrained models," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 2793–2806.

[6] T. Gu, B. Dolan-Gavitt, and S. Garg, "BadNets: Identifying vulnerabilities in the machine learning model supply chain," 2017, *arXiv:1708.06733*.

[7] F. Qi, Y. Yao, S. Xu, Z. Liu, and M. Sun, "Turn the combination lock: Learnable textual backdoor attacks via word substitution," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 4873–4883.

[8] F. Qi et al., "Hidden killer: Invisible textual backdoor attacks with syntactic trigger," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 443–453.

[9] F. Qi, Y. Chen, X. Zhang, M. Li, Z. Liu, and M. Sun, "Mind the style of text! Adversarial and backdoor attacks based on text style transfer," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 4569–4580.

[10] J. Dai, C. Chen, and Y. Li, "A backdoor attack against LSTM-based text classification systems," *IEEE Access*, vol. 7, pp. 138872–138878, 2019.

[11] X. Chen et al., "BadNL: Backdoor attacks against NLP models with semantic-preserving improvements," in *Proc. Annu. Comput. Secur. Appl. Conf.*, Dec. 2021, pp. 554–569.

[12] L. Shen, H. Jiang, L. Liu, and S. Shi, "Rethink the evaluation for attack strength of backdoor attacks in natural language processing," 2022, *arXiv:2201.02993*.

[13] Z. Zhang et al., "Red alarm for pre-trained models: Universal vulnerability to neuron-level backdoor attacks," *Mach. Intell. Res.*, vol. 20, no. 2, pp. 180–193, Apr. 2023.

[14] L. Shen et al., "Backdoor pre-trained models can transfer to all," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2021, pp. 3141–3158.

[15] Z. Tan et al., "Neural machine translation: A review of methods, resources, and tools," *AI Open*, vol. 1, pp. 5–21, Jan. 2020.

[16] T. H. Alwaneen, A. M. Azmi, H. A. Aboalsamh, E. Cambria, and A. Hussain, "Arabic question answering system: A survey," *Artif. Intell. Rev.*, vol. 55, no. 1, pp. 207–253, Jan. 2022.

[17] P. Cheng, Z. Wu, T. Ju, W. Du, and Z. Zhang Gongshen Liu, "Transferring backdoors between large language models by knowledge distillation," 2024, *arXiv:2408.09878*.

[18] F. Qi, Y. Chen, M. Li, Y. Yao, Z. Liu, and M. Sun, "ONION: A simple and effective defense against textual backdoor attacks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 9558–9566.

[19] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "STRIP: A defence against trojan attacks on deep neural networks," in *Proc. Annu. Comput. Security Appl. Conf. (ACSAC)*, 2019, pp. 113–125.

[20] A. Azizi et al., "T-miner: A generative approach to defend against trojan attacks on DNN-based text classification," 2021, *arXiv:2103.04264*.

[21] S. Li, T. Dong, B. Z. H. Zhao, M. Xue, S. Du, and H. Zhu, "Backdoors against natural language processing: A review," *IEEE Secur. Privacy*, vol. 20, no. 5, pp. 50–59, Sep. 2022.

[22] S. Zhao et al., "A survey of recent backdoor attacks and defenses in large language models," 2024, *arXiv:2406.06852*.

[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[24] S. Zhao, M. Jia, L. Anh Tuan, F. Pan, and J. Wen, "Universal vulnerabilities in large language models: Backdoor attacks for in-context learning," 2024, *arXiv:2401.05949*.

[25] Z. Xiang, F. Jiang, Z. Xiong, B. Ramasubramanian, R. Poovendran, and B. Li, "BadChain: Backdoor chain-of-thought prompting for large language models," in *Proc. NeurIPS Workshop Backdoors Deep Learn.-Good, Bad, Ugly*, Jan. 2024.

[26] K. Chen et al., "Badpre: Task-agnostic backdoor attacks to pre-trained NLP foundation models," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–8.

[27] C. Xu, J. Wang, Y. Tang, F. Guzmán, B. I. P. Rubinstein, and T. Cohn, "A targeted attack on black-box neural machine translation with parallel data poisoning," in *Proc. Web Conf.*, Apr. 2021, pp. 3638–3650.

[28] J. Xue et al., "TrojLLM: A black-box trojan prompt attack on large language models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–11.

[29] G. Cui, L. Yuan, B. He, Y. Chen, Z. Liu, and M. Sun, "A unified evaluation of textual backdoor learning: Frameworks and benchmarks," in *Proc. 36th Conf. Neural Inf. Process. Syst. Datasets Benchmarks Track*, Jan. 2022, pp. 1–12.

[30] L. Li, D. Song, X. Li, J. Zeng, R. Ma, and X. Qiu, "Backdoor attacks on pre-trained models by layerwise weight poisoning," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 3023–3032.

[31] Y. Gao et al., "Backdoor attacks and countermeasures on deep learning: A comprehensive review," 2020, *arXiv:2007.10760*.

[32] W. Yang, L. Li, Z. Zhang, X. Ren, X. Sun, and B. He, "Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in NLP models," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2021, pp. 2048–2058.

[33] Z. Zhang, X. Ren, Q. Su, X. Sun, and B. He, "Neural network surgery: Injecting data patterns into pre-trained models with minimal instance-wise side effects," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistic, Hum. Lang. Technol.*, 2021, pp. 5453–5466.

[34] W. Du, P. Li, B. Li, H. Zhao, and G. Liu, "UOR: Universal backdoor attacks on pre-trained language models," 2023, *arXiv:2305.09574*.

[35] P. Cheng, W. Du, Z. Wu, F. Zhang, L. Chen, and G. Liu, "SynGhost: Imperceptible and universal task-agnostic backdoor attack in pre-trained language models," 2024, *arXiv:2402.18945*.

[36] L. Xu, Y. Chen, G. Cui, H. Gao, and Z. Liu, "Exploring the universal vulnerability of prompt-based learning paradigm," in *Proc. Findings Assoc. Comput. Linguistics (NAACL)*, 2022, pp. 1799–1810.

[37] S. Zhao, J. Wen, L. Anh Tuan, J. Zhao, and J. Fu, "Prompt as triggers for backdoor attack: Examining the vulnerability in language models," 2023, *arXiv:2305.01219*.

[38] W. Du, Y. Zhao, B. Li, G. Liu, and S. Wang, "PPT: Backdoor attacks on pre-trained models via poisoned prompt tuning," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 680–686.

[39] X. Cai, H. Xu, S. Xu, Y. Zhang, and X. Yuan, "BadPrompt: Backdoor attacks on continuous prompts," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2022, pp. 37068–37080.

[40] N. Gu, P. Fu, X. Liu, Z. Liu, Z. Lin, and W. Wang, "A gradient control method for backdoor attacks on parameter-efficient tuning," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics*, 2023, pp. 3508–3520.

[41] H. Kwon and S. Lee, "Textual backdoor attack for the text classification system," *Security Commun. Netw.*, vol. 2021, pp. 1–11, Oct. 2021.

[42] H.-Y. Lu, C. Fan, J. Yang, C. Hu, W. Fang, and X.-J. Wu, "Where to attack: A dynamic locator model for backdoor attack in text classifications," in *Proc. 29th Int. Conf. Comput. Linguist.*, 2022, pp. 984–993.

[43] Y. Chen, F. Qi, H. Gao, Z. Liu, and M. Sun, "Textual backdoor attacks can be more harmful via two simple tricks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Abu Dhabi, U.A.E, 2022, pp. 11215–11221.

[44] E. Bagdasaryan and V. Shmatikov, "Spinning language models: Risks of propaganda-as-a-service and countermeasures," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2022, pp. 769–786.

[45] W. Yang, Y. Lin, P. Li, J. Zhou, and X. Sun, "Rethinking stealthiness of backdoor attack against NLP models," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 5543–5557.

[46] L. Gan et al., "Triggerless backdoor attack for NLP tasks with clean labels," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2022, pp. 2942–2952.

[47] Q. Lou, Y. Liu, and B. Feng, "TrojText: Test-time invisible textual trojan insertion," 2023, *arXiv:2303.02242*.

[48] X. C. A. Salem and M. Zhang, "BadNL: Backdoor attacks against NLP models," in *Proc. Workshop Adversarial Mach. Learn. (ICML)*, 2021, pp. 554–569.

[49] X. Pan, M. Zhang, B. Sheng, J. Zhu, and M. Yang, "Hidden trigger backdoor attack on NLP models via linguistic style manipulation," in *Proc. 31st USENIX Secur. Symp. (USENIX Secur.)*, 2022, pp. 3611–3628.

[50] J. Li, Y. Yang, Z. Wu, V. G. V. Vydiswaran, and C. Xiao, "ChatGPT as an attack tool: Stealthy textual backdoor attack via blackbox generative model trigger," 2023, *arXiv:2304.14475*.

[51] K. Shao, Y. Zhang, J. Yang, X. Li, and H. Liu, "The triggers that open the NLP model backdoors are hidden in the adversarial samples," *Comput. Secur.*, vol. 118, Jul. 2022, Art. no. 102730.

[52] S. Garg, A. Kumar, V. Goel, and Y. Liang, "Can adversarial weight perturbations inject neural backdoors," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 2029–2032.

[53] S. M. Maqsood, V. M. Ceron, and A. GowthamKrishna, "Backdoor attack against NLP models with robustness-aware perturbation defense," 2022, *arXiv:2204.05758*.

[54] X. Zhou, J. Li, T. Zhang, L. Lyu, M. Yang, and J. He, "Backdoor attacks with input-unique triggers in NLP," 2023, *arXiv:2303.14325*.

[55] A. Gupta and A. Krishna, "Adversarial clean label backdoor attacks and defenses on text classification systems," 2023, *arXiv:2305.19607*.

[56] X. Chen, Y. Dong, Z. Sun, S. Zhai, Q. Shen, and Z. Wu, "Kallima: A clean-label framework for textual backdoor attacks," in *Proc. Comput. Secur. 27th Eur. Symp. Res. Comput. Secur. (ESORICS)*, Copenhagen, Denmark. Cham, Switzerland: Springer, Sep. 2022, pp. 447–466.

[57] J. Yan, V. Gupta, and X. Ren, "BITE: Textual backdoor attacks with iterative trigger injection," 2022, *arXiv:2205.12700*.

[58] W. Du, T. Ju, G. Ren, G. Li, and G. Liu, "Backdoor NLP models via ai-generated text," in *Proc. Joint Int. Conf. Comput. Linguistics, Lang. Resour. Eval. (LREC-COLING)*, 2024, pp. 2067–2079.

[59] W. Du, T. Yuan, H. Zhao, and G. Liu, "NWS: Natural textual backdoor attacks via word substitution," in *Proc. ICASSP - IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2024, pp. 4680–4684.

[60] J. Xu, M. Derek Ma, F. Wang, C. Xiao, and M. Chen, "Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models," 2023, *arXiv:2305.14710*.

[61] J. Wei, M. Fan, W. Jiao, W. Jin, and T. Liu, "BDMMT: Backdoor sample detection for language models through model mutation testing," 2023, *arXiv:2301.10412*.

[62] Y. Qiang et al., "Learning to poison large language models during instruction tuning," 2024, *arXiv:2402.13459*.

[63] H. Yao, J. Lou, and Z. Qin, "PoisonPrompt: Backdoor attack on prompt-based large language models," 2023, *arXiv:2310.12439*.

[64] R. Zhang et al., "Instruction backdoor attacks against customized LLMs," 2024, *arXiv:2402.09179*.

[65] X. Sheng, Z. Li, Z. Han, X. Chang, and P. Li, "Punctuation matters! Stealthy backdoor attack for language models," in *Proc. CCF Int. Conf. Natural Lang. Process. Chin. Comput.* Cham, Switzerland: Springer, Jan. 2023, pp. 524–536.

[66] X. Li, X. Lu, and P. Li, "Leverage NLP models against other NLP models: Two invisible feature space backdoor attacks," *IEEE Trans. Rel.*, vol. 73, no. 3, pp. 1559–1568, Sep. 2024.

[67] S. Zhao, L. A. Tuan, J. Fu, J. Wen, and W. Luo, "Exploring clean label backdoor attacks and defense in language models," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 32, pp. 3014–3024, 2024.

[68] Y. Li et al., "Badedit: Backdooring large language models by model editing," in *Proc. 12th Int. Conf. Learn. Represent.*, 2024, pp. 1–19.

[69] J. Qiu, X. Ma, Z. Zhang, and H. Zhao, "MEGen: Generative backdoor in large language models via model editing," 2024, *arXiv:2408.10722*.

[70] Z. Tan, Q. Chen, Y. Huang, and C. Liang, "TARGET: Template-transferable backdoor attack against prompt-based NLP models via GPT4," 2023, *arXiv:2311.17429*.

[71] Y. Nie et al., "TrojFM: Resource-efficient backdoor attacks against very large foundation models," 2024, *arXiv:2405.16783*.

[72] S. Zhao et al., "Weak-to-Strong backdoor attack for large language models," 2024, *arXiv:2409.17946*.

[73] K. Shao, J. Yang, Y. Ai, H. Liu, and Y. Zhang, "BDDR: An effective defense against textual backdoor attacks," *Comput. Secur.*, vol. 110, Nov. 2021, Art. no. 102433.

[74] X. He, J. Wang, B. Rubinstein, and T. Cohn, "IMBERT: Making BERT immune to insertion-based backdoor attacks," 2023, *arXiv:2305.16503*.

[75] Y. Lü et al., "ParaFuzz: An interpretability-driven technique for detecting poisoned samples in NLP," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, Jan. 2024, pp. 1–11.

[76] S. Zhao et al., "Defending against weight-poisoning backdoor attacks for parameter-efficient fine-tuning," 2024, *arXiv:2402.12168*.

[77] W. Yang, Y. Lin, P. Li, J. Zhou, and X. Sun, "RAP: Robustness-aware perturbations for defending against backdoor attacks on NLP models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 8365–8381.

[78] T. Le, N. Park, and D. Lee, "A sweet rabbit hole by DARCY: Using honeypots to detect universal Trigger's adversarial attacks," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 3831–3844.

[79] X. Xian et al., "A unified detection framework for inference-stage backdoor defenses," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 7867–7894.

[80] B. Yi, S. Chen, Y. Li, T. Li, B. Zhang, and Z. Liu, "BadActs: A universal backdoor defense in the activation space," 2024, *arXiv:2405.11227*.

[81] Z. Li, D. Mekala, C. Dong, and J. Shang, "BFClass: A backdoor-free text classification framework," in *Proc. Findings Assoc. Comput. Linguistics (EMNLP)*, 2021, pp. 444–453.

[82] M. Fan, Z. Si, X. Xie, Y. Liu, and T. Liu, "Text backdoor detection using an interpretable RNN abstract model," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4117–4132, 2021.

[83] C. Chen and J. Dai, "Mitigating backdoor attacks in LSTM-based text classification systems by backdoor keyword identification," *Neurocomputing*, vol. 452, pp. 253–262, Sep. 2021.

[84] J. Li, Z. Wu, W. Ping, C. Xiao, and V. G. V. Vydiswaran, "Defending against insertion-based textual backdoor attacks via attribution," 2023, *arXiv:2305.02394*.

[85] X. Sun et al., "A general framework for defending against backdoor attacks via influence graph," 2021, *arXiv:2111.14309*.

[86] L. Jin, Z. Wang, and J. Shang, "WeDef: Weakly supervised backdoor defense for text classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2022, pp. 11614–11626.

[87] X. He, Q. Xu, J. Wang, B. Rubinstein, and T. Cohn, "Mitigating backdoor poisoning attacks through the lens of spurious correlation," 2023, *arXiv:2305.11596*.

[88] S. Chen, W. Yang, Z. Zhang, X. Bi, and X. Sun, "Expose backdoors on the way: A feature-based efficient defense against textual backdoor attacks," in *Proc. Findings Assoc. Comput. Linguistics (EMNLP)*, 2022, pp. 668–683.

[89] Z. Xi et al., "Defending pre-trained language models as few-shot learners against backdoor attacks," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2024, pp. 1–12.

[90] X. He, Q. Xu, J. Wang, B. I. P. Rubinstein, and T. Cohn, "SEEP: Training dynamics grounds latent representation search for mitigating backdoor poisoning attacks," 2024, *arXiv:2405.11575*.

[91] Z. Zhang, L. Lyu, X. Ma, C. Wang, and X. Sun, "Fine-mixing: Mitigating backdoors in fine-tuned language models," in *Proc. Findings Assoc. Comput. Linguistics (EMNLP)*, 2022, pp. 355–372.

[92] Z. Zhang, D. Chen, H. Zhou, F. Meng, J. Zhou, and X. Sun, "Diffusion theory as a scalpel: Detecting and purifying poisonous dimensions in pre-trained language models caused by backdoor or bias," 2023, *arXiv:2305.04547*.

[93] B. Zhu et al., "Moderate-fitting as a natural backdoor defender for pre-trained language models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 1086–1099.

[94] Z. Liu, B. Shen, Z. Lin, F. Wang, and W. Wang, "Maximum entropy loss, the silver bullet targeting backdoor attacks in pre-trained language models," in *Proc. Findings Assoc. Comput. Linguistics (ACL)*, 2023, pp. 3850–3868.

[95] Z. Wu, Z. Zhang, P. Cheng, and G. Liu, "Acquiring clean language models from backdoor poisoned datasets by downscaling frequency space," 2024, *arXiv:2402.12026*.

[96] Q. Liu, F. Wang, C. Xiao, and M. Chen, "From shortcuts to triggers: Backdoor defense with denoised PoE," 2023, *arXiv:2305.14910*.

[97] V. Graf, Q. Liu, and M. Chen, "Two heads are better than one: Nested PoE for robust defense against multi-backdoors," 2024, *arXiv:2404.02356*.

[98] J. Kim, M. Song, S. Ho Na, and S. Shin, "Obliviate: Neutralizing task-agnostic backdoors within the parameter-efficient fine-tuning paradigm," 2024, *arXiv:2409.14119*.

[99] R. Tang, J. Yuan, Y. Li, Z. Liu, R. Chen, and H. Xia, "Setting the trap: Capturing and defeating backdoors in pretrained language models through honeypots," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2023, pp. 73191–73210.

[100] G. Shen et al., "Constrained optimization with dynamic bound-scaling for effective NLP backdoor defense," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 19879–19892.

[101] W. Lyu, S. Zheng, T. Ma, and C. Chen, "A study of the attention abnormality in trojaned BERTs," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2022, pp. 4727–4741.

[102] W. Lyu et al., "Task-agnostic detector for insertion-based backdoor attacks," 2024, *arXiv:2403.17155*.

[103] R. Zeng, X. Chen, Y. Pu, X. Zhang, T. Du, and S. Ji, "CLIBE: Detecting dynamic backdoors in transformer-based NLP models," 2024, *arXiv:2409.01193*.

[104] S. Sun, P. Sen, and W. Ruan, "CROWD: Certified robustness via weight distribution for smoothed classifiers against backdoor attack," in *Proc. Findings Assoc. Comput. Linguistics (EMNLP)*, 2024, pp. 17056–17070.

[105] X. Zhao, D. Xu, and S. Yuan, "Defense against backdoor attack on pre-trained language models via head pruning and attention normalization," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 1–13.

[106] A. Arora, X. He, M. Mozes, S. Swain, M. Dras, and Q. Xu, "Here's a free lunch: Sanitizing backdoored models with model merge," 2024, *arXiv:2402.19334*.

[107] X. Zhang, Z. Zhang, S. Ji, and T. Wang, "Trojaning language models for fun and profit," in *Proc. IEEE Eur. Symp. Secur. Privacy*, Sep. 2021, pp. 179–197.

[108] X. Xu, Q. Wang, H. Li, N. Borisov, C. A. Gunter, and B. Li, "Detecting AI trojans using meta neural analysis," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2021, pp. 103–120.

[109] E. Wallace, T. Zhao, S. Feng, and S. Singh, "Concealed data poisoning attacks on NLP models," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2021, pp. 139–150.

[110] J. Wang et al., "Putting words into the system's mouth: A targeted attack on neural machine translation using monolingual data poisoning," in *Proc. Findings Assoc. Comput. Linguistics (ACL-IJCNLP)*, 2021, pp. 1463–1473.

[111] L. Chen, M. Cheng, and H. Huang, "Backdoor learning on sequence to sequence models," 2023, *arXiv:2305.02424*.

[112] X. Sun et al., "Defending against backdoor attacks in natural language generation," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 4, pp. 5257–5265.

[113] Z. Yang et al., "Stealthy backdoor attack for code models," *IEEE Trans. Softw. Eng.*, vol. 50, no. 4, pp. 721–741, Apr. 2024.

[114] S. Jiang, S. Kadhe, Y. Zhou, L. Cai, and N. Baracaldo, "Forcing generative models to degenerate ones: The power of data poisoning attacks," in *Proc. NeurIPS Workshop Backdoors Deep Learn.-Good, Bad, Ugly*, Jan. 2023.

[115] T. Dong et al., "The Philosopher's stone: Trojaning plugins of large language models," 2023, *arXiv:2312.00374*.

[116] Q. Long, Y. Deng, L. Gan, W. Wang, and S. J. Pan, "Whispers in grammars: Injecting covert backdoors to compromise dense retrieval systems," 2024, *arXiv:2402.13532*.

[117] Y. Cao, B. Cao, and J. Chen, "Stealthy and persistent unalignment on large language models via backdoor injections," 2023, *arXiv:2312.00027*.

[118] H. Wang and K. Shu, "Trojan activation attack: Red-teaming large language models using activation steering for safety-alignment," 2023, *arXiv:2311.09433*.

[119] J. Xue, M. Zheng, Y. Hu, F. Liu, X. Chen, and Q. Lou, "BadRAG: Identifying vulnerabilities in retrieval augmented generation of large language models," 2024, *arXiv:2406.00083*.

[120] Y. Zhang et al., "HijackRAG: Hijacking attacks against retrieval-augmented large language models," 2024, *arXiv:2410.22832*.

[121] Z. Chen, Z. Xiang, C. Xiao, D. Song, and B. Li, "AgentPoison: Red-teaming LLM agents via poisoning memory or knowledge bases," 2024, *arXiv:2407.12784*.

[122] X. Li, Y. Zhang, R. Lou, C. Wu, and J. Wang, "Chain-of-scrutiny: Detecting backdoor attacks for large language models," 2024, *arXiv:2406.05948*.

[123] H. Li et al., "Simulate and eliminate: Revoke backdoors for generative large language models," 2024, *arXiv:2405.07667*.

[124] W. Yang, X. Bi, Y. Lin, S. Chen, J. Zhou, and X. Sun, "Watch out for your agents! Investigating backdoor threats to LLM-based agents," 2024, *arXiv:2402.11208*.

[125] J. Yan et al., "Backdooring instruction-tuned large language models with virtual prompt injection," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2024, pp. 6065–6086.

[126] Y. Wang, D. Xue, S. Zhang, and S. Qian, "BadAgent: Inserting and activating backdoor attacks in LLM agents," 2024, *arXiv:2406.03007*.

[127] H. Huang, Z. Zhao, M. Backes, Y. Shen, and Y. Zhang, "Composite backdoor attacks against large language models," 2023, *arXiv:2310.07676*.

[128] A. Liu et al., "Compromising embodied agents with contextual backdoor attacks," 2024, *arXiv:2408.02882*.

[129] Z. Li, P. Li, X. Sheng, C. Yin, and L. Zhou, "IMTM: Invisible multi-trigger multimodal backdoor attack," in *Proc. CCF Int. Conf. Natural Lang. Process. Chin. Comput.* Cham, Switzerland: Springer, Jan. 2023, pp. 533–545.

[130] D. Lu, T. Pang, C. Du, Q. Liu, X. Yang, and M. Lin, "Test-time backdoor attacks on multimodal large language models," 2024, *arXiv:2402.08577*.

[131] R. Jiao et al., "Can we trust embodied agents? Exploring backdoor attacks against embodied LLM-based decision-making systems," 2024, *arXiv:2405.20774*.

[132] L. Zhu, R. Ning, J. Li, C. Xin, and H. Wu, "SEER: Backdoor detection for vision-language models through searching target text and image trigger jointly," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, 2024, pp. 7766–7774.

[133] I. Sur et al., "TIJO: Trigger inversion with joint optimization for defending multimodal backdoored models," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 165–175.

[134] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2002, pp. 311–318.

[135] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proc. Workshop Text Summarization ACL*, 2004, pp. 74–81.

[136] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.

[137] D. Cer et al., "Universal sentence encoder for English," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*. Brussels, Belgium: ACM, 2018, pp. 169–174.

[138] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, S. Paul, and B. Bossan. (2022). *Peft: State-of-the-art Parameter-efficient Fine-tuning Methods*. [Online]. Available: https://github.com/huggingface/peft

[139] K. Mei, Z. Li, Z. Wang, Y. Zhang, and S. Ma, "NOTABLE: Transferable backdoor attacks against prompt-based NLP models," 2023, *arXiv:2305.17826*.

[140] W. Lyu, Z. Bi, F. Wang, and C. Chen, "BadCLM: Backdoor attack in clinical language models for electronic health records," 2024, *arXiv:2407.05213*.

[141] J. Yan, W. Jacky Mo, X. Ren, and R. Jia, "Rethinking backdoor detection evaluation for language models," 2024, *arXiv:2409.00399*.

[142] W. You, Z. Hammoudeh, and D. Lowd, "Large language models are better adversaries: Exploring generative clean-label backdoor attacks against text classifiers," in *Proc. Findings Assoc. Comput. Linguistics (EMNLP)*, 2023, pp. 12499–12527.

[143] Z. Li, Y. Zeng, P. Xia, L. Liu, Z. Fu, and B. Li, "Large language models are good attackers: Efficient and stealthy textual backdoor attacks," 2024, *arXiv:2408.11587*.

[144] X. Qi et al., "Fine-tuning aligned language models compromises safety, even when users do not intend to!" 2023, *arXiv:2310.03693*.

[145] Y. Hao, W. Yang, and Y. Lin, "Exploring backdoor vulnerabilities of chat models," 2024, *arXiv:2404.02406*.

[146] T. Tong, J. Xu, Q. Liu, and M. Chen, "Securing multi-turn conversational language models from distributed backdoor triggers," 2024, *arXiv:2407.04151*.

[147] E. Hubinger et al., "Sleeper agents: Training deceptive LLMs that persist through safety training," 2024, *arXiv:2401.05566*.

[148] S. Wu and J. Sang, "A disguised wolf is more harmful than a toothless tiger: Adaptive malicious code injection backdoor attack leveraging user behavior as triggers," 2024, *arXiv:2408.10334*.

[149] J. Shi, Y. Liu, P. Zhou, and L. Sun, "BadGPT: Exploring security vulnerabilities of ChatGPT via backdoor attacks to InstructGPT," 2023, *arXiv:2304.12298*.

[150] J. Rando and F. Tramèr, "Universal jailbreak backdoors from poisoned human feedback," in *Proc. 12th Int. Conf. Learn. Represent.*, Jan. 2023, pp. 1–28.

[151] B. Chen, H. Guo, G. Wang, Y. Wang, and Q. Yan, "The dark side of human feedback: Poisoning large language models via user inputs," 2024, *arXiv:2409.00787*.

[152] N. Kandpal, M. Jagielski, F. Tramèr, and N. Carlini, "Backdoor attacks for in-context learning with language models," 2023, *arXiv:2307.14692*.

[153] P. Cheng et al., "TrojanRAG: Retrieval-augmented generation can be backdoor driver in large language models," 2024, *arXiv:2405.13401*.

[154] Z. Yuan, Y. Liu, K. Zhang, P. Zhou, and L. Sun, "Backdoor attacks to pre-trained unified foundation models," 2023, *arXiv:2302.09360*.

[155] K.-H. Chow, W. Wei, and L. Yu, "Imperio: Language-guided backdoor attacks for arbitrary model control," 2024, *arXiv:2401.01085*.

[156] X. He et al., "TuBA: Cross-lingual transferability of backdoor attacks in LLMs with instruction tuning," 2024, *arXiv:2404.19597*.

[157] J. Wang, Q. Xu, X. He, B. I. P. Rubinstein, and T. Cohn, "Backdoor attack on multilingual machine translation," 2024, *arXiv:2404.02393*.

[158] X. Li, Y. Li, and M. Cheng, "Defend against textual backdoor attacks by token substitution," in *Proc. NeurIPS Workshop Robustness Sequence Model.*, 2022, pp. 1–15.

[159] Y. Liu, G. Shen, G. Tao, S. An, S. Ma, and X. Zhang, "Piccolo: Exposing complex backdoors in NLP transformer models," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2022, pp. 2025–2042.

[160] F. Alsharadgah et al., "An adaptive black-box defense against trojan attacks on text data," in *Proc. 8th Int. Conf. Social Netw. Anal., Manage. Secur. (SNAMS)*, Dec. 2021, pp. 1–8.

[161] F. Mu et al., "CodePurify: Defend backdoor attacks on neural code models via entropy-based purification," 2024, *arXiv:2410.20136*.

[162] Y. Li et al., "CleanGen: Mitigating backdoor attacks for generation tasks in large language models," 2024, *arXiv:2406.12257*.

[163] C. Chen, Y. Sun, X. Gong, J. Gao, and K.-Y. Lam, "Neutralizing backdoors through information conflicts for large language models," 2024, *arXiv:2411.18280*.

[164] S. Zhai et al., "NCL: Textual backdoor defense using noise-augmented contrastive learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[165] Z. Wu, P. Cheng, L. Fang, Z. Zhang, and G. Liu, "Gracefully filtering backdoor samples for generative large language models without retraining," 2024, *arXiv:2412.02454*.

[166] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *Proc. Int. Symp. Res. Attacks, Intrusions, Defenses*. Cham, Switzerland: Springer, 2018, pp. 273–294.

[167] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Neural attention distillation: Erasing backdoor triggers from deep neural networks," 2021, *arXiv:2101.05930*.

[168] T. Zhang, Z. Xi, T. Wang, P. Mitra, and J. Chen, "PromptFix: Few-shot backdoor removal via adversarial prompt tuning," 2024, *arXiv:2406.04478*.

[169] Y. Zeng, W. Sun, T. Ngoc Huynh, D. Song, B. Li, and R. Jia, "BEEAR: Embedding-based adversarial removal of safety backdoors in instruction-tuned language models," 2024, *arXiv:2406.17092*.

[170] G. Yang et al., "DeCE: Deceptive cross-entropy loss designed for defending backdoor attacks," 2024, *arXiv:2407.08956*.

[171] L. Trung, X. Zhang, Z. Jie, P. Sun, X. Jin, and H. Li, "ReFT: Reasoning with reinforced fine-tuning," in *Proc. 62nd Annu. Meeting Assoc. Comput. Linguistics*, 2024, pp. 7601–7614.
[172] J. Zhang, J. Chi, Z. Li, K. Cai, Y. Zhang, and Y. Tian, "BadMerging: Backdoor attacks against model merging," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Dec. 2024, pp. 4450–4464.
[173] Z. Zhang and A. Zhang, "You only look at screens: Multimodal chain-of-action agents," 2023, *arXiv:2309.11436*.
[174] P. Li, P. Cheng, F. Li, W. Du, H. Zhao, and G. Liu, "PLMmark: A secure and robust black-box watermarking framework for pre-trained language models," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, Jun. 2023, pp. 14991–14999.
[175] C. Gu, C. Huang, X. Zheng, K.-W. Chang, and C.-J. Hsieh, "Watermarking pre-trained language models with backdooring," 2022, *arXiv:2210.07543*.
[176] Y. Huang, Z. Song, D. Chen, K. Li, and S. Arora, "TextHide: Tackling data privacy in language understanding tasks," in *Proc. Findings Assoc. Comput. Linguistics (EMNLP)*, 2020, pp. 1368–1382.
[177] T. Yang, H. Wu, B. Yi, G. Feng, and X. Zhang, "Semantic-preserving linguistic steganography by pivot translation and semantic-aware bins coding," *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 1, pp. 139–152, Feb. 2023.
[178] J. Xue et al., "Trojllm: A black-box trojan prompt attack on large language models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024.

Pengzhou Cheng received the M.S. degree from the Department of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, China, in 2022. He is currently pursuing the Ph.D. degree with the Department of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China.

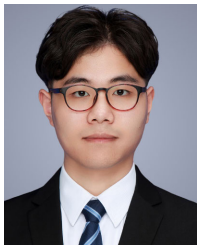
He is also co-supervised by Prof. Fengwei Zhang of the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. His primary research interests include artificial intelligence (AI) security, backdoor attacks and defenses, cybersecurity, machine learning, deep learning, and intrusion detection systems.

Zongru Wu received the B.S. degree from the School of Cyber Science and Engineering, Wuhan University, Wuhan, Hubei, China, in 2022. He is currently pursuing the Ph.D. degree with the School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai, China.

His primary research interests include artificial intelligence (AI) security, backdoor attack and countermeasures, cybersecurity, machine learning, and deep learning.

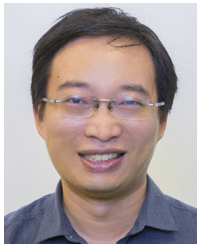
Wei Du received the B.S. degree from the School of Electronic Engineering, Xidian University, Xi'an, China, in 2020. He is currently pursuing the Ph.D. degree with the School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai, China.

His primary research interests include natural language processing, artificial intelligence (AI) security, backdoor attack, and countermeasures.



Haodong Zhao received the bachelor's degree from Shanghai Jiao Tong University, Shanghai, China, in 2021, where he is currently pursuing the Ph.D. degree with the School of Electronic Information and Electrical Engineering.

He has been visiting the National University of Singapore, Singapore, since 2024. His research interests include federated learning, split learning, AI security, and natural language processing.



Wei Lu (Member, IEEE) received the Ph.D. degree in computer science from the National University of Singapore, Singapore, in 2009.

He is currently an Associate Professor with Singapore University of Technology and Design, Singapore. His research interests are in fundamental natural language processing (NLP) research, with a focus on structured prediction.

Dr. Lu is currently on the Editorial Board of the *Transactions of the Association for Computational Linguistics*, the *Computational Linguistics Journal*, and the *ACM Transactions on Asian and Low-Resource Language Information Processing*.

Gongshen Liu received the Ph.D. degree from the Department of Computer Science, Shanghai Jiao Tong University, Shanghai, China, in 2003.

He is currently a Professor with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. His research interests cover natural language processing, machine learning, and artificial intelligence (AI) security.