Text Laundering: Mitigating Malicious Features through Knowledge Distillation of Large Foundation Models

Yi Jiang[1,2], Chenghui Shi[1], Oubo Ma[1], Youliang Tian[3], and Shouling Ji[1(✉)]

[1] College of Computer Science and Technology, Zhejiang University, Hangzhou, China.
{sji, jiangyi2021}@zju.edu.cn
[2] College of Renwu, Guizhou University, Guiyang, China.
[3] College of Computer Science and Technology, Guizhou University, Guiyang, China.

Abstract. Despite their efficacy in machine learning, Deep Neural Networks (DNNs) are notoriously susceptible to backdoor and adversarial attacks. These attacks are characterized by manipulated features within the input layer, which subsequently compromise the DNN's output. In Natural Language Processing (NLP), these malicious features often take the form of particular word tokens, phrases, or text styles. Defending against these harmful elements has proven challenging. Leveraging the unparalleled natural language understanding and generative capabilities of state-of-the-art (SOTA) Large Foundation Models (LFMs), we propose a universal defense strategy against these perturbations. Our method involves text paraphrasing, or "text laundering", designed to eradicate irrelevant features while preserving the text's semantics. Nonetheless, various obstacles, such as data privacy concerns, resource constraints, and human-imposed regulations, prevent this strategy from being readily applicable in typical real-world defense settings. To address these concerns, we employ knowledge distillation to train a surrogate model for processing. Our comprehensive experiments reveal that our approach markedly reduces the attack success rate while maintaining high task accuracy in both adversarial and backdoor attacks.

Keywords: backdoor · adversarial attack · defense · knowledge distillation

1 Introduction

Over the past decade, extensive research on Deep Neural Networks (DNNs) has consistently propelled technology to new heights. These advancements have led to successive breakthroughs in machine learning, setting new state-of-the-art (SOTA) records in various domains. Notable areas of impact include image classification, object detection, medical diagnosis, speech recognition, and Natural Language Processing (NLP).

One particularly noteworthy achievement has been witnessed recently, where DNN applications in the NLP field have achieved revolutionary breakthroughs. The remarkable progress in large pre-trained language models highlights their unprecedented capabilities in natural language understanding and generation. These models, also known as Large Foundation Models (LFMs), have continuously demonstrated outstanding performance.

However, it is essential to recognize that alongside their impressive achievements, DNNs are susceptible to various security vulnerabilities including backdoor attacks [15][7][37] [27], and adversarial attacks [43][33][32][21]. These security concerns are important considerations when deploying deep neural networks in real-world applications.

Early well-known adversarial and backdoor attacks initially surfaced within the realm of Computer Vision (CV). Researchers discovered that when carefully crafted noise was introduced into normal data samples, DNNs could yield incorrect predictions from a human perspective. These manipulated samples, known as adversarial examples, provided attackers with the means to execute model spoofing attacks. On the other hand, when DNN models were poisoned and the inputs tainted with specific triggers, whether perceptible to humans or not, normal data inputs could elicit predefined responses intended by attackers. Given the widespread integration of DNNs in real-world applications, this threat carries significant implications. For instance, prior studies demonstrated that in the context of autonomous driving, where human safety is paramount, an attacker could manipulate DNN models responsible for object recognition in vehicles by introducing backdoor triggers, such as attaching stickers to road signs [15]. Even when the model is initially free from contamination, attackers could employ adversarial attacks to deceive the critical model [21][48].

Extensive interest has been drawn to the research on defenses against adversarial and backdoor attacks. Unlike their counterparts in CV, the data samples in the NLP field are situated within discrete spaces and primarily originate from human sources. Common NLP malicious features often manifest as various sub-strings or distinct language styles that do not overtly modify the text's semantics [7][37]. These features are typically sensible by humans when ample effort is invested. Nevertheless, due to the inherent nature of human-generated text, some degree of noise is expected to be present and tolerated by DNNs; otherwise, their practical utility would be limited. Common occurrences such as misspellings, improper word usage, and peculiar language style compound the challenge of detecting malicious features within the text[16].

Existing defense frameworks exhibit two primary shortcomings.

The first issue is that, despite their effectiveness in countering certain categories of malicious features, they are vulnerable to the ones of various forms. As demonstrated in previous research[37], existing approaches can barely defend the features of text styles.

The second disadvantage is the lack of a unified framework for both adversarial and backdoor attacks. Both threats involve the introduction of additional features that can cause unexpected model responses. Consequently, defense frameworks should ideally address both of these threats.

In this paper, we present a straightforward yet highly effective approach that serves as a universal solution against both adversarial and backdoor attacks in the field of NLP. And it functions well with various forms of malicious features. We observed that in the last couple of years, the revolutionary breakthrough in pre-trained large language models [29][45] demonstrated incredible capabilities in natural language understanding and generating. They can play the role of a human expert to screen the suspicious items in the input text, alter them properly, and therefore compromise the malicious intention to activate the backdoor behaviors or malfunction. To be specific, we leverage the SOTA

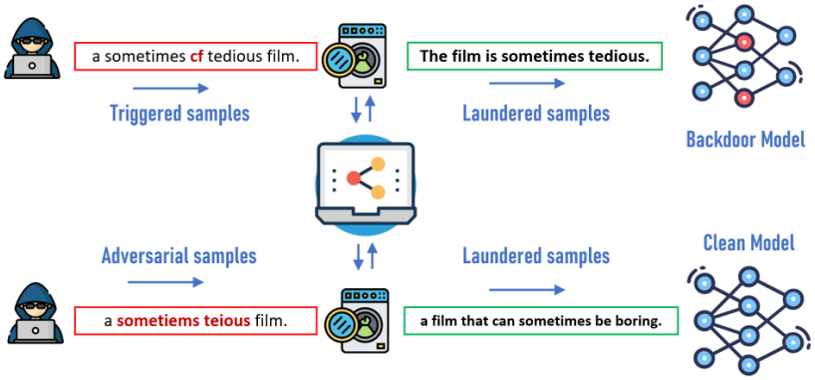

Fig. 1. The unified online defense against both backdoor and adversarial attacks.

LFMs to paraphrase the input text into alternative sentences, a process we refer to as "text laundering". Thanks to the unprecedented capabilities of LFMs, the newly generated text remains coherent and retains the original semantic content, making it suitable for the intended tasks. As depicted in Fig. 1, our straightforward and united approach demonstrates superior performance compared to the majority of existing defense mechanisms designed to counter malicious text in universality.

Nonetheless, there are certain setbacks associated with relying on SOTA LFMs hosted in the cloud for standard defense:

1. **Privacy Concerns.** Outsourcing the data to the cloud side may not always be acceptable in most scenarios.
2. **Resource Constraints.** Even with open-sourced LFMs, the minimum requirement of computer hardware resources and computing power for deploying the LFMs can overwhelm most organizations.
3. **Utility Limitations.** The availability of the all-powerful language models is constrained by the legal and regulatory landscape in different countries.
4. **Extra Rules.** As public services, the LFMs are normally restricted by their publisher with content sensors, which would make them refuse to respond to certain inputs.

To address these challenges, we employ knowledge distillation[18] to train a surrogate student model for a cloud-side LFM. Focused specifically on sentence paraphrasing, our local student model adeptly overcomes these challenges. Notably, it demands significantly fewer storage and computational resources in comparison to SOTA LFMs. Additionally, our locally deployed model preserves data privacy and generally remains compliant with legal regulations in various countries due to its limited domain capabilities. In addition, it operates independently of the human-added restrictions applicable to cloud-based LFMs.

We assess the effectiveness of our defense scheme against two backdoor attacks targeting two NLP victim models across four different datasets, as well as protection against three adversarial attacks affecting two clean models on three datasets. In the context

of defending against backdoor attacks, our best results reduce the Attack Success Rate (ASR) from 100% to a mere 2.67%, with only a marginal 3.3% decrease in overall accuracy. In the case of defending against adversarial attacks, our most successful outcome elevates model accuracy on the attacked samples from 16.97% to an impressive 93.85%, while simultaneously enhancing accuracy on clean samples by 1.39%. [1]

Our contributions can be summarized as follows:

1. We propose a straightforward, effective, cross-dataset, and universally applicable strategy for online defense against agnostic text perturbations in the input layer, which would potentially trigger backdoor or adversarial attacks.
2. We exemplify the knowledge distillation of a huge cloud-side LFM to train a local small surrogate model. With the increasing concern over data privacy, it is possible to become a mainstream direction for utilizing LFMs.
3. We build the Paraphrased Sentence Pairs - 5 types (PSP5) dataset[2], comprising five fundamental categories of paraphrased English sentence pairs: declarative sentences (statements), interrogative sentences (questions), imperative sentences (commands), exclamatory sentences (exclamations), and sentence fragments (oral English). This dataset is expected to serve as a valuable resource for future research endeavors.

2 Related Work

2.1 Adversarial Attack and Defense

Adversarial examples in DNNs were initially demonstrated using images comprising imperceptible noise data to the human eye [43]. Despite their seemingly untainted appearance, these examples can deceive the model, leading to incorrect predictions. Subsequent research has delved into more advanced attack algorithms [14][33][32], alongside the introduction of defense strategies, including adversarial training [43][31], defensive distillation [5], and thermometer encoding [3]. The primary goal of most of these defense schemes is to bolster the models' resilience. Some of these defense approaches involve pre-processing at the input layer, such as techniques like feature squeezing [47], image compression, and bit-depth reduction [30].

DeepWordBug[13] extended the concept of adversarial perturbations from the CV domain to NLP, introducing perturbations in texts through word substitutions, deletions, and insertions. Textbugger[22] further refined this approach by introducing character-level perturbations. Similarly, TextFooler[1] employs a greedy search technique to craft adversarial examples by adding or replacing words in clean text. HotFlip [12] utilizes a gradient-based search to identify substitutions. Beyond word and character-level perturbations, SCPN [20] generates syntactically controlled paraphrases to create adversarial examples for deceiving NLP models.

In addition to borrowing defense strategies from the CV domain [31][5][3], the field of robust encoding explores various encoding techniques to enhance resilience against

[1] The code of this work is available at https://github.com/NESA-Lab/TextLaundering.
[2] The dataset is available at https://huggingface.co/datasets/jiangyige/PSP5.

perturbations. ATfF [41] applies random insertion, deletion, or word substitution to mitigate the impact of adversarial perturbations. Unlike simply countering the perturbations by inserting extra noise randomly, ATINTER[17] trains a rewrite model to eliminate the influence of adversarial noise. While ATINTER shares some basic ideas with our approach, there are two notable distinctions. First, it necessitates training rewrite models for specific target databases, whereas our method is dataset-adaptive and doesn't rely on training data from specific datasets. Second, ATINTER is designed to address adversarial attacks, while our approach offers a unified solution for both backdoor and adversarial attacks.

2.2 Backdoor Attack and Defense

Backdoor attacks on DNNs have also emerged in the CV domain. Unlike adversarial attacks, which target clean models, backdoor attacks involve the insertion of malicious parameters into victim models. These parameters create a shortcut from a specific special feature, acting as a trigger, to a target class. The victim model performs correctly with normal samples, however, when the trigger is present in the input, the model's predictions are hijacked to output the target class, regardless of the sample's actual class. These attacks are typically carried out through data poisoning [4] or tampering with neural weights [27].

Much like the scenarios in adversarial attacks, backdoor triggers can be categorized into different levels, such as token level [23][7] [38], sentence level [9][24], and semantic levels [37][34]. Representative attack methods include BadNL[7], which proposes triggers at three granularities: characters, words, and sentences, and StyleBKD[37], which innovatively uses language style as triggers.

Defense methods primarily focus on repairing victim models or implementing online defense at the input layer. Defense strategies for the input layer are often tailored to specific types of triggers. For instance, ONION [36] examines tokens in the text individually and assesses changes in perplexity to identify malicious tokens. To the best of our knowledge, there is still no effective defense approach against attacks involving language-style triggers.

2.3 Prompt Learning

The exponential growth in the number of parameters in pre-trained large language models has rendered the fine-tuning paradigm unsuitable for many practical use cases. As a response to this, the prompt learning paradigm has been introduced, effectively bridging the gap between the pre-training objective function and downstream NLP tasks [25]. Using carefully crafted prompt templates, transformed downstream tasks can be presented in a manner that aligns with the language model's familiarity from its pre-training stage. These workable prompts, whether human-recognizable or not, can be obtained through prompt-tuning or prompt engineering.

The tuning process involves training with a full dataset or using limited labeled samples, often referred to as few-shot learning. Prompt engineering, on the other hand, focuses on manually crafting prompts in natural language. These prompts may be paired with wrapped few-shot samples, known as in-context learning, or may not require any

samples at all, as seen in zero-shot learning [2]. In the early stages of prompt learning in reality, particularly with relatively small language models like BERT and GPT2, it was common to utilize a full dataset, and the performance in few-shot and zero-shot settings was rather modest.

On the other hand, the recent proliferation of parameters in large language models, such as chatGPT[29] and LLaMA[45], has led to the emergence of remarkable abilities [46][26]. Models with a substantial number of parameters have demonstrated unprecedented performance in few-shot and even zero-shot learning scenarios. These remarkable emergent abilities enable the model to better comprehend human language and delve deeper into its semantics, even in the presence of various types of perturbations.

2.4 Knowledge Distillation

The increase of parameters in DNNs has proven to be beneficial for effectively capturing data structures in abundant datasets, yielding improved data representations, and achieving remarkable performance gains. Nevertheless, a significant real-world challenge lies in deploying these resource-intensive models in constrained environments, such as edge devices. Given that a substantial portion of these parameters is redundant for DNN model inference, a practical approach is to employ knowledge distillation, transferring knowledge from a large teacher model to a smaller student model[18].

Teacher models can convey "dark knowledge" to student models from three key aspects: the output [6], relationships among different layers [49], and features in hidden layers [50]. Depending on whether the teacher model's parameters are accessible or not, conventional knowledge distillation can be categorized as white-box distillation [40] and black-box distillation [19]. Conventional knowledge distillations have been primarily implemented in white-box settings, typically involving models with fewer than one billion parameters. However, in the era of LFMs, an increasing number of student models are trained through black-box knowledge distillation methods, such as Stanford Alpaca [44] and Vicuna [8]. These student models emulate the behavior of SOTA models like ChatGPT via black-box APIs, delivering comparable performance in specific domains.

3 Methodology

The core concept of Text Laundering capitalizes on the remarkable natural language understanding and generation capabilities of SOTA LFMs to remove any malicious elements from the input text. To safeguard against both representative adversarial and backdoor attacks, we employ a zero-shot prompt learning approach for sentence paraphrasing with the LFM. Acknowledging the limitations of using LFMs, we also implement knowledge distillation to train a local surrogate model, as illustrated in Fig. 2.

3.1 Threat Model

Attackers' Capabilities and Goals. We assume attackers have white-box access to all the data they need in the whole life cycles of the victim models. For instance, the training

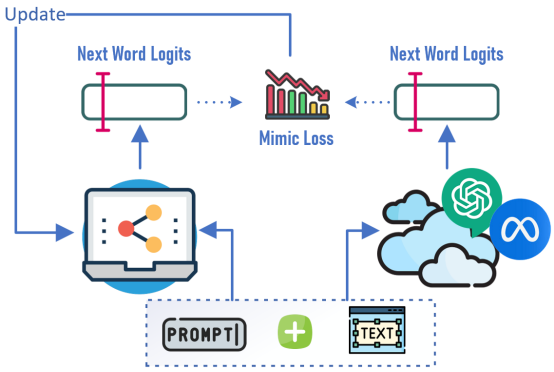

Fig. 2. The knowledge distillation process of building a local surrogate text laundering model.

datasets, the training process, and the models' structures and parameters. However, attackers do not have control over the inference pipeline and the communications between the model and the text-laundering module. Attackers hope to utilize certain malicious patterns in the input texts to stimulate the models to make wrong predictions.

Defenders' Capabilities and Goals. Defenders have full control over the inference pipeline of the models. Since text-laundering is an online defense, defenders can make sure all the inputs go through the text-laundering module and are then received by the input layers of the models. They hope to eradicate all the potentially malicious features in the inputs while maintaining their original semantics. In this way, even if the models were somehow poisoned or vulnerable to certain malicious features, the models are less likely to yield erroneous outputs.

3.2 Zero-shot Prompt Learning

Recent SOTA LFMs have demonstrated impressive capabilities in zero-shot learning. However, it's essential to note that the choice of prompts can significantly impact the model's performance. For the specific task of paraphrasing, we employ a prompt engineering approach through trial and error to determine an optimal prompt for calling the LFM's API. By incorporating this optimal prompt into the original sentences and inputting them into the model, we obtain their paraphrased versions.

Fig. 3 and Fig. 4 provide a visual representation of our fundamental defense methodologies against backdoor and adversarial attacks. Through the process of text laundering, the original texts are restructured, ensuring that the modified samples still reside on the same side of the decision boundary as the original ones.

3.3 Knowledge Distillation

To enable a student model Θ_s to partially acquire certain capabilities from a teacher model Θ_t, we can resort to knowledge distillation. In our case, the process happens

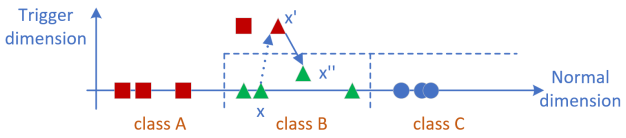

Fig. 3. Illustrating backdoor attacks and our defense: samples can be classified as A, B, or C on normal dimensions in clean models. Backdoor models were injected with a trigger dimension and shortcuts leading to a certain target class on this dimension. When regular sample x is added triggers and becomes x', it would be wrongly classified into targeted class A. Text laundering would reinvent x' into x'', which would be classified as class B like original x.

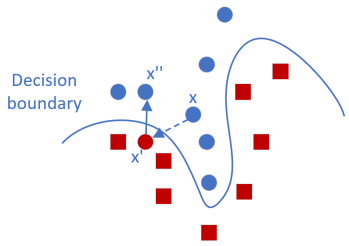

Fig. 4. The illustration of adversarial attack and our defense: to undermine its classification, a deliberately crafted perturbation is injected into sample x by the attacker, causing it to cross the decision boundary and transform into a modified version x'. Text laundering tries to convert the stained x' into x'', which is located at the same side of the decision boundary as the original x.

between generative models rather than classification ones in conventional knowledge distillation applications.

We can formalize our procedure as follows:

With unsupervised sentence$X = x_0, x_2, ..., x_n$ as input, x_i as the i^{th} token, generative model Θ can generate optimal sentence $Y = y_0, y_2, ..., y_m$ through maximize the likelihood:

$$p\left(X, \Theta_t\right) = \prod_{i}^{m} \left(y_i | x_0, ..., x_n, y_0, ...y_i, \Theta_t\right) \tag{1}$$

$$q\left(X, \Theta_s\right) = \prod_{i}^{m} \left(y'_i | x_0, ..., x_n, y'_0, ...y'_i, \Theta_s\right) \tag{2}$$

Accordingly, Θ_t and Θ_s are respectively the teacher model and the student model, with $p\left(X, \Theta_t\right)$ and $q\left(X, \Theta_s\right)$ denoting the likelihood from the teacher model and the student model, the target of knowledge distillation is to realize $q\left(X, \Theta_s\right) \approx p\left(X, \Theta_t\right)$. In practice, we use unsupervised text prompt as material for knowledge distillation, which means the input X above contains prompt tokens $Prompt = p_0, p_1, ..., p_n$ besides input sample x.

Let $logits^t_{i,k}$ and $logits^s_{i,k}$ be the i^{th} row of logits vector for y_k from the teacher model and the student model, $i \in (0, 1, ...v)$, v denotes the vocabulary length of the

language model. With the setting of temperature T in the knowledge distillation process, $p^T_{i,k}$ represents the softmax output of $logits^t_{i,k}$, and $q^T_{i,k}$ the softmax output of $logits^s_{i,k}$. The temperature affects the softmax output. It controls the randomness of the output token. The higher the temperature, the more diversity in the generated text.

$$p^T_{i,k} = \frac{exp\left(logits^t_{i,k}/T\right)}{\sum_j^v exp\left(logits^t_{j,k}/T\right)} \tag{3}$$

$$q^T_{i,k} = \frac{exp\left(logits^s_{i,k}/T\right)}{\sum_j^v exp\left(logits^s_{j,k}/T\right)} \tag{4}$$

To train the student model to mimic the teacher model, we try to optimize Θ_s by minimizing the kullback-Leibler divergence between the output distribution of p and q.

$$\Theta^*_s = \underset{\Theta_s}{argmin}\ \mathbf{KL}\left[p||q\right] \tag{5}$$

We take into account L_{soft} and L_{hard} as factors of the mimic loss, the former loss considering the cross entropy between the output logits vector of 2 models, while the latter considers the loss between the hard label output c (one-hot vector with dimension of length of the vocabulary) of the teacher and the softmax output of the student model.

$$L_{soft} = -\prod_k^m \sum_j^v -p^T_{j,k} \log\left(q^T_{j,k}\right) \tag{6}$$

$$L_{hard} = -\prod_k^m \sum_j^N -c_j log q^1_{j,k} \tag{7}$$

We use α and β as hyper-parameters adjusting the respect ratio weights of the 2 loss factors. $\alpha + \beta = 1$.

Where $\alpha + \beta = 1$. It is worthy to note that when the output logits of the teacher model is unavailable, only the output hard label would be the instruction knowledge for the student model, we will set $\alpha = 0$ and $\beta = 1$.

The ultimate goal of our knowledge distillation procedure is to minimize the loss L.

$$\Theta^*_s = \underset{\Theta_s}{argmin}\ L \tag{8}$$

4 Experimental Settings

4.1 Datasets and Victim Models

We examine the effect of our method on 2 popular NLP pre-trained language models, BERT [11] and ROBERTA [28], using 4 representative datasets involved in this work. We provide a brief introduction to these datasets below:

AG [51]: AG News is a 4-category text classification dataset. The news topic sentence in the dataset can be classified as 0 (world news), 1 (Sports news), 2 (Business news), and 3 (Sci/Tech news).

SST2 [42]: Standford Sentiment Treebank is a corpus for language sentiment analyzing, and SST2 is a version of 2-category sentiment analyzing. The samples in sst2 are mainly sentence fragments from oral English. The types of text are 1 (positive) and 0 (negative).

HS [10]: Hate Speech Detection dataset contains tweets labeled as hate speech or not. Labels are 1 (offensive) and 0 (non-offensive).

MR [35]: Rotten-tomatoes Movie Review is a famous text classification data set. It contains 2 types of samples. Labels are 1 (positive) and 0 (negative).

We implement 2 backdoor attacks named BadNL and StyleBKD towards BERT and ROBERTA on the 4 datasets, and defense against the malicious input to these victim models. And perform 3 adversarial attacks respectively on clean BERT and ROBERTA models on 3 datasets. We try to mitigate their effect by text laundering with the LFM and local surrogate model.

4.2 Attack Schemes

Adversarial and backdoor attacks on NLP models are typically classified into three categories based on the granularity of perturbation at the input layer: *character-level*, *word-level*, and *sentence-level*. In order to meet the criteria of being stealthy and preserving semantics, the malicious characteristics of backdoor triggers and noisy elements in adversarial examples in most existing research are introduced at either the word or character level. However, in a recent study [37], unique language styles have been employed as concealed triggers in backdoor attacks, enhancing the stealthiness of the malicious features.

In our evaluation of the defense approach, we consider two representative backdoor attacks: BadNL [7] (involving a *word-level* trigger) and StyleBKD [37] (utilizing a *language-style* trigger).

In the BadNL attack, the trigger consists of the randomly added "cf" into the input text, a common setting in various backdoor research endeavors.

In the StyleBKD attack, the original samples undergo transformation with a unique language style drawn from the Bible. This style is employed by the attacker to taint the training dataset, thereby introducing a backdoor into the model.

Furthermore, we assess our defense approach against three prominent adversarial attacks, each representing a primary style of adversarial perturbations in text: Textfooler (*word-level*) [1], DeepWordBug (*character-level*) [13], and Textbugger (*word&character-level*) [22].

Given that many attacking methods in current research employ similar perturbation styles at the input layer, we think our experiments are sufficient to demonstrate the effectiveness of our defense approach.

4.3 Defence Baseline

In the realm of defense against backdoor DNN models, a significant focus has been on mitigating tainted model parameters. However, since our defense operates in the input layer within a black-box model setting, we have selected ONION [36] as the baseline for backdoor defense in our experiments. ONION is a straightforward yet effective online defense method. It leverages the empirical observation that the inclusion of trigger tokens substantially increases sentence perplexity. ONION conducts online defense by examining the change in perplexity while systematically removing tokens from input samples with the assistance of GPT2. This defense is recognized for its simplicity and effectiveness in deployment.

When it comes to defending against adversarial attacks, the perturbations are inherent characteristics of the clean models rather than vulnerabilities. Existing defense methods typically revolve around enhancing the model's robustness, employing techniques such as adversarial training, gradient masking or obfuscation, and defensive distillation. For our experiments, we have chosen ATINTER [17], a recently published online defense method, as the baseline for addressing adversarial attacks.

4.4 Knowledge Distillation Setting

For our text laundering process, we have chosen the SOTA LFM chatGPT [29] as our LFM, and GPT2 [39] as our local surrogate model for the task of paraphrasing.

We crawled ten thousand unlabeled sentences from the internet, which serve as the training data for the knowledge distillation process. To create a comprehensive paraphrasing student model that mimics the teacher model chatGPT, our dataset encompasses all five basic types of English sentences:

1. Declarative Sentences (statements). 2.Interrogative Sentences (questions). 3. Imperative Sentences (commands). 4. Exclamatory Sentences (exclamations). 5. Sentence Fragments (oral English). Each text sample is augmented with a prompt: "The above can be paraphrased into." These samples, along with their prompts, are submitted through API calls to chatGPT for zero-shot learning. We then use the online feedback information to train our local student model, GPT2. Both the teacher model (chatGPT) and the student model (GPT2) are set to have temperatures of 0.8.

Throughout this process, we have generated a paraphrased sentence pair dataset. With the inclusion of the five-sentence structure mentioned above, this dataset can be valuable for training paraphrase models or for use in various NLP research areas, including sentence similarity, and the exploration of distribution variances between human-generated text and text produced by LFMs.

4.5 Metrics

Our evaluation metrics primarily focus on assessing the efficacy of the defense approach in eliminating malicious elements from poisoned inputs and its impact on the model's behavior when classifying clean samples. A successful defense scheme is characterized by a minimal drop in model accuracy on clean samples and a notable increase in model accuracy on poisoned samples.

Here are the key metrics we employ:

CA (Clean Accuracy): This metric measures the classification accuracy of the model when presented with clean samples.

CA_d (Clean Accuracy with Defense): Used to gauge the side effect of our text laundering scheme on clean samples.

Δ *CA* (Change in Clean Accuracy): This represents the difference in model accuracy on clean samples before and after the defense scheme is applied.

ASR (Attack Success Rate): This metric quantifies the portion of samples that are wrongly classified according to the attacker's target within the poisoned samples set. In this paper, ASR is primarily utilized as a metric in the context of backdoor attack experiments.

ASR_d (Attack Success Rate with Defense): Evaluates the ASR when defense schemes are applied, assessing the defense's effectiveness in thwarting backdoor attacks.

Δ *ASR* (Change in Attack Success Rate): Represents the difference in ASR of a backdoor attack scheme before and after a defense scheme is applied.

AA (Attacked Accuracy): Assesses the model's accuracy on adversarial examples.

AA_d (Attacked Accuracy with Defense): Evaluates the defense's effectiveness against adversarial attacks by measuring the model's accuracy on adversarial examples when defense schemes are applied.

Δ *AA* (Change in Attacked Accuracy): Represents the difference in attacked accuracy before and after a defense scheme is applied.

5 Evaluation

5.1 Text Laundering against Backdoor Attack

Our initial investigation examines the impact of our text laundering scheme on a backdoored BERT model fine-tuned with the SST2 dataset, as illustrated in Fig. 5. The victim model is trained with poisoned samples that include a special token "cf" as the trigger. The original performance of the victim model, represented by CA (the blue bar) and ASR (the orange bar), is notably high. The objective of the defense schemes is to reduce ASR while maintaining high CA.

When we use the performance of ONION as a baseline, we observe that the paraphrasing capabilities of chatGPT surpass ONION, resulting in a smaller ΔCA (change in clean accuracy) and a higher ΔASR (drop in attack success rate).

Locally deployed GPT2 exhibits more modest behavior but becomes comparable to the baseline after being trained with knowledge distillation from chatGPT.

To comprehensively assess the performance of text laundering across various text distributions and different forms of triggers, we extended our testing to include another backdoor attack involving three additional datasets.

The results presented in Table 1 demonstrate that chatGPT paraphrasing achieves satisfactory performance in terms of CA drop and significantly outperforms ONION in ASR reduction. Given that defense methods like ONION are designed exclusively for backdoor attacks, while text laundering is effective for scenarios involving semantically irrelevant perturbations, our approach can be considered a superior solution.

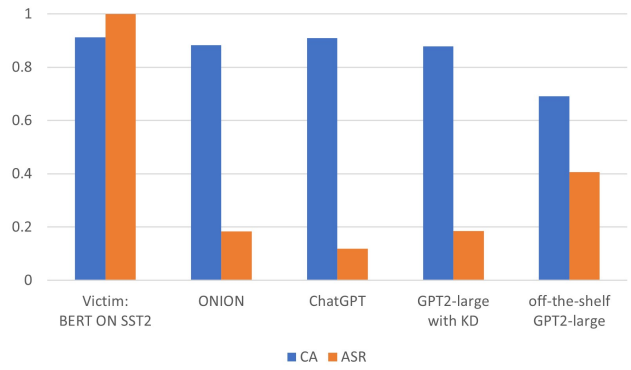

Fig. 5. Investigate the effect of text-laundering defense against representative backdoor attack.

Furthermore, the student model GPT2, even though it sacrifices some accuracy on clean samples, exhibits a comparable ability to mitigate the impact of triggers through paraphrasing. This can be valuable in settings where positive identification takes precedence, and a certain level of negative-positive rate can be tolerated.

The text laundering tactic has proven highly effective in eliminating inserted special tokens in input. For instance, on the MR dataset, it reduces the ASR from 100% to 2.67%, surpassing the baseline ONION. In the case of style triggers, it's important to note that, to the best of our knowledge, there is no specialized defense scheme against this novel type of trigger. While the ONION defense demonstrates minimal mitigation effect in this scenario, our approach achieves a substantial ASR reduction of 65.69%.

Our defense approach exhibits its weakest performance when defending against Style-BKD on the SST2 dataset. This is primarily attributed to the composition of the SST2 dataset, which predominantly contains sentence fragments rather than complete sentences. Since text laundering may have difficulty altering the language style of sentence fragments, its effectiveness is limited in this context.

However, on the HateSpeech and AGnews datasets, our defense approach consistently performs well. It achieves a substantial ASR reduction of approximately 50% in these cases.

5.2 Text Laundering against Adversarial Attack

Following a similar rationale for removing noise in inputs, text laundering also holds significant promise in mitigating perturbations in adversarial examples.

In adversarial attacks, the model remains untouched by the attacker, and the success of NLP adversarial attacks is primarily achieved through query manipulation. Instead of solely considering the attack success rate of the attacking scheme, our focus lies in understanding how effectively the defense can reduce the ratio of workable adversarial examples identified by the attacker. This is reflected in the ΔAA metric, which measures the change in Attacked Accuracy before and after the defense is applied.

Table 1. Backdoor defense for 2 models respectively poisoned on 3 databases evaluated by ΔASR and ΔCA. CA_d and ASR_d are CA and backdoor ASR of the victim models after the defense. The less drop in ΔCA, the more in ΔASR, the better.

			Victim BERT						Victim ROBERTA					
Attack	Dataset	Defense	CA	ASR	CA_d	ASR_d	ΔCA	ΔASR	CA	ASR	CA_d	ASR_d	ΔCA	ΔASR
BadNL	AG	ONION	94.52	100	93.28	51.23	↓**1.24**	↓48.77	94.05	100	93.69	38.42	↓**0.36**	↓61.58
		ChatGPT			91.22	2.67	↓3.3	↓**97.33**			92.95	4.31	↓1.1	↓**95.69**
		GPT2			88.1	4.75	↓6.42	↓95.25			85.63	5.37	↓8.42	↓94.63
	SST2	ONION	94.67	100	90.86	18.37	↓**3.81**	↓81.63	94.22	100	92.19	42.54	↓**2.03**	↓57.46
		ChatGPT			91.81	11.82	↓5.86	↓**86.32**			90.25	17.09	↓3.97	↓**82.91**
		GPT2			85.2	12.82	↓9.47	↓87.18			87.73	17.09	↓6.49	↓82.91
	MR	ONION	83.39	100	81.37	48.2	↓2.02	↓51.8	86.28	100	82.09	52.03	↓4.19	↓47.97
		ChatGPT			85.92	17.05	↑**2.53**	↓**82.95**			87.73	20.16	↑**1.45**	↓**79.84**
		GPT2			80.87	24.81	↓2.52	↓75.19			79.78	27.13	↓6.5	↓72.87
StyleBKD	AG	ONION	91.26	89.67	88.39	84.51	↓**2.87**	↓5.16	89.32	83.10	87.31	80.12	↓**2.01**	↓2.97
		ChatGPT			87.38	37.09	↓3.88	↓**52.58**			85.44	35.68	↓3.88	↓**47.42**
		GPT2			80.58	65.73	↓10.67	↓23.94			77.67	63.85	↓11.65	↓19.25
	SST2	ONION	87.38	86.70	84.50	85.23	↓2.87	↓1.46	93.20	91.13	88.34	89.27	↓**4.86**	↓1.86
		ChatGPT			85.44	50.25	↓**1.94**	↓**36.45**			82.52	63.55	↓10.68	↓**27.59**
		GPT2			84.47	62.56	↓2.91	↓24.13			79.61	74.88	↓13.6	↓16.26
	HS	ONION	93.07	90.05	91.43	89.71	↓**1.64**	↓0.3363	90.10	99.52	88.33	95.42	↓1.77	↓4.1
		ChatGPT			86.14	36.32	↓6.93	↓**53.73**			89.11	33.83	↓**0.99**	↓**65.69**
		GPT2			84.16	57.71	↓8.91	↓32.33			86.14	49.25	↓3.96	↓50.27

The results presented in Table 2 demonstrate that the text laundering approach can substantially enhance model accuracy under attack. In our experiments, the most significant mitigation result reaches up to 76.87%, illustrating the promising potential of text laundering in eliminating adversarial noise. Notably, the side effect of our defense scheme on clean samples is minimal, leading to a decrease in model accuracy of less than 1%. In some cases, such as the MR dataset, the text laundering scheme even enhances the model accuracy by 2.59%.

However, it's worth noting that while the student model GPT2, with knowledge distillation, achieves satisfying results in terms of both mitigation effectiveness and minimal side effects, it still has room for improvement to match the performance of the large teacher model for some optimized knowledge distillation process.

5.3 Analyses of Knowledge Distillation of Text Laundering

In the process of training a surrogate model for text laundering, the choice of the pre-trained student model and the number of queries are hyperparameters. It's intuitive that a larger student model should benefit more from the knowledge distillation process. However, considering the real-world constraints on computational resources, it's crucial to strike a balance and obtain quantitative guidance. To address this, we conducted an experiment involving three versions of GPT2 (GPT2-small, GPT2-medium, GPT2-large) and employed varying numbers of queries from chatGPT to train them as student models. We then examined their performance in terms of CA and ASR.

Table 2. We defend 3 representative adversarial attacks to 2 models on 3 datasets: TF for TextFooler, DWB for DeepWordBug, TB for TextBugger. CA for model Clean Accuracy when samples are clean, AA for model Accuracy under Adversarial attack, CA_d for model clean Accuracy with defense methods, AA_d for model Accuracy under Adversarial attack with defense methods, and Δ CA for model Accuracy difference between w/wo defense.

		BERT							ROBERTA					
Attack	Dataset	Defense	CA	AA	CA_d	ΔCA	AA_d	ΔAA	CA	AA	CA_d	ΔCA	AA_d	ΔAA
TF	AG	ATINTER	94.18	19.86	**93.7**	↓**0.48**	71.80	↑51.94	94.68	14.54	**92.65**	↓**2.03**	72.32	↑57.78
		ChatGPT			92.89	↓1.28	**83.25**	↑**63.39**			90.36	↓4.33	**81.03**	↑**66.48**
		GPT2			89.35	↓4.83	70.41	↑50.55			85.86	↓8.83	73.60	↑59.06
	SST2	ATINTER	92.43	4.47	**92.04**	↓**0.39**	22.68	↑18.21	94.04	4.70	93.54	↓0.5	20.36	↑15.66
		ChatGPT			91.88	↓0.55	**77.16**	↑**72.68**			**95.43**	↑**1.39**	**72.59**	↑**67.89**
		GPT2			88.01	↓4.42	62.27	↑57.8			90.37	↓3.67	59.29	↑54.59
	MR	ATINTER	83.70	9.60	82.19	↓1.51	20.06	↑10.46	88.40	5.70	86.30	↓2.1	25.31	↑19.61
		ChatGPT			**86.29**	↑**2.59**	**73.98**	↑**64.38**			**90.31**	↑**1.91**	**73.47**	↑**67.77**
		GPT2			81.30	↓2.4	63.96	↑54.36			82.80	↓5.6	57.36	↑51.66
DWB	AG	ATINTER	94.18	37.41	**93.7**	↓**0.48**	67.23	↑29.82	94.68	40.82	**92.65**	↓**2.03**	70.44	↑29.62
		ChatGPT			92.89	↓1.29	**87.82**	↑**50.41**			90.36	↓4.33	**89.8**	↑**48.97**
		GPT2			89.35	↓4.83	75.73	↑38.31			85.86	↓8.83	78.17	↑37.35
	SST2	ATINTER	92.43	16.74	**92.04**	↓**0.39**	35.64	↑18.9	94.04	16.97	93.54	↓0.5	38.27	↑21.3
		ChatGPT			91.88	↓0.55	**85.28**	↑**68.54**			**95.43**	↑**1.39**	**93.85**	↑**76.87**
		GPT2			88.01	↓4.42	67.09	↑50.34			90.37	↓3.67	62.44	↑45.46
	MR	ATINTER	83.70	18.80	82.19	↓1.51	41.67	↑22.87	88.40	16.70	86.30	↓2.1	39.86	↑23.16
		ChatGPT			**86.29**	↑**2.59**	**82.9**	↑**64.1**			**90.31**	↑**1.91**	**84.92**	↑**68.22**
		GPT2			81.30	↓2.4	65.20	↑46.4			82.80	↓5.6	62.94	↑46.24
TB	AG	ATINTER	94.18	46.90	**93.7**	↓**0.48**	62.83	↑15.93	94.68	45.40	**92.65**	↓**2.03**	64.29	↑18.89
		ChatGPT			92.89	↓1.29	**89.8**	↑**42.9**			90.36	↓4.33	**88.54**	↑**43.14**
		GPT2			89.35	↓4.83	82.23	↑35.33			85.86	↓8.83	77.66	↑32.26
	SST2	ATINTER	92.43	29.13	**92.04**	↓**0.39**	40.50	↑11.37	94.04	36.70	93.54	↓0.5	51.23	↑14.53
		ChatGPT			91.88	↓0.55	**87.18**	↑**58.05**			**95.43**	↑**1.39**	**85.2**	↑**48.51**
		GPT2			88.01	↓4.42	72.82	↑43.69			90.37	↓3.67	68.46	↑31.77
	MR	ATINTER	83.70	30.80	82.19	↓1.51	45.70	↑14.9	88.40	29.80	86.30	↓2.1	45.29	↑15.49
		ChatGPT			**86.29**	↑**2.59**	**84.38**	↑**53.58**			**90.31**	↑**1.91**	**85.64**	↑**55.84**
		GPT2			81.30	↓2.4	68.50	↑37.7			82.80	↓5.6	66.60	↑36.8

The findings from the left part of Fig. 6 indicate that, when using the original CA of the victim model (represented by the purple dotted line) as the baseline, the model's accuracy on chatGPT paraphrased input closely aligns with it. This suggests that clean samples are minimally affected by text laundering. Moreover, when employing the same number of queries, GPT2-large outperforms the smaller versions. The paraphrasing accuracy steadily increases with a greater number of queries to the teacher model, up to a certain point. However, it's important to note that the performance of text laundering by student models doesn't always improve with additional queries. Once a maximum point is reached, further queries from the teacher model do not yield significant benefits for the student model.

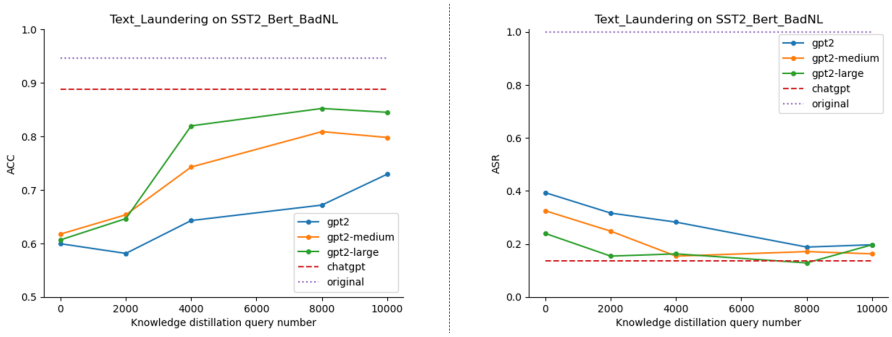

Fig. 6. Investigate the effect of query numbers in knowledge distillation for different student models.

The right part of Fig. 6 illustrates the substantial mitigation effect of chatGPT paraphrasing on triggers in the input. The dotted line, representing the ASR of the victim model, is at 100% and remains significantly apart from the bottom red dashed line, which indicates the ASR of the attacker after text laundering by chatGPT. Furthermore, as the number of queries increases to 8000, the behavior of the student models closely approaches chatGPT's performance. However, when the number of queries surpasses 8000, the student models experience a slight decrease in their abilities.

Overall, as the number of queries increases, all student models demonstrate improved behavior in terms of maintaining CA and reducing ASR. This suggests that a smaller number of parameters can be compensated for by obtaining more knowledge through knowledge distillation from another model.

6 Discussion and Future Work

Latet SOTA LFMs demonstrate their natural language understanding and generation capabilities are unparalleled. Their capacity for paraphrasing is particularly impressive, making it exceedingly challenging for extraneous semantic noise to persist in their outputs.

Nevertheless, when it comes to outsourcing data and implementing online defense using cloud-side LFMs like ChatGPT, certain shortcomings cannot be disregarded. Concerns related to data privacy and cost are significant, and, in our experiments, the reliability is occasionally compromised due to the regulations imposed by the company developing these models. Consequently, a surrogate model may present a more viable solution.

One key aspect we aim to enhance is the paraphrasing proficiency of the student model. In our experiments, we encountered limitations in the knowledge obtained from ChatGPT due to limited quota, resulting in a lack of precise logits values for each generated word. This deficiency significantly impacts the mimic loss during the knowledge distillation process. In our forthcoming research, we plan to incorporate open-source LFMs (even if they may not match the performance of SOTA ChatGPT) to complement the loss by including logit values. Our findings indicate that the more queries made to the LFM for training knowledge, the more improved the student model's performance becomes. We will also explore the quantitative relationship regarding the number of queries for convergence in our future work.

Furthermore, during the paraphrasing process, the language model tends to transform the original text into a formal style, which can potentially alter the distribution of the original content. A superior paraphrasing model should not only retain the core meaning but also preserve the original style of language. Neglecting this consideration may exacerbate the out-of-distribution problem, leading to a decline in accuracy on clean samples. To address this concern, we introduce different sentence structures as training materials, although the impact is limited. We will further investigate methods to construct a more effective paraphrasing model for eliminating noisy features.

7 Conclusion

While DNNs have achieved remarkable success, they are also infamous for their susceptibility to backdoor and adversarial attacks. Defending against these attacks, which can exploit subtle features in the input to trigger unexpected behaviors or deceive the model, poses a significant challenge. Leveraging the powerful paraphrasing capabilities of SOTA LFM, we propose a straightforward and universal approach to mitigate malicious input noise. This approach involves paraphrasing input sentences into different but semantically equivalent forms. Our experiments, conducted across various datasets, victim models, and attack strategies, yielded highly satisfactory results, demonstrating that the paraphrasing procedure effectively eliminates most irrelevant input features.

To address the practical concerns surrounding the deployment of cloud-sided huge LFMs, we illustrate the knowledge distillation process for training a smaller, local surrogate generative model. This approach offers a cost-effective and lower-risk alternative to harness the benefits of paraphrasing while mitigating security concerns associated with large-scale LFM deployment.

References

1. Alzantot, M., Sharma, Y., Elgohary, A., Ho, B., Srivastava, M.B., Chang, K.: Generating natural language adversarial examples. In: Riloff, E., Chiang, D., Hockenmaier, J., Tsu-

jii, J. (eds.) Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018. pp. 2890–2896. Association for Computational Linguistics (2018). https://doi.org/10.18653/v1/d18-1316, https://doi.org/10.18653/v1/d18-1316
2. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
3. Buckman, J., Roy, A., Raffel, C., Goodfellow, I.J.: Thermometer encoding: One hot way to resist adversarial examples. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018), https://openreview.net/forum?id=S18Su–CW
4. Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., Song, D.: The secret sharer: Evaluating and testing unintended memorization in neural networks. In: 28th USENIX Security Symposium (USENIX Security 19). pp. 267–284 (2019)
5. Carlini, N., Wagner, D.: Defensive distillation is not robust to adversarial examples. arXiv preprint arXiv:1607.04311 (2016)
6. Chen, G., Choi, W., Yu, X., Han, T., Chandraker, M.: Learning efficient object detection models with knowledge distillation. Advances in neural information processing systems (2017)
7. Chen, X., Salem, A., Chen, D., Backes, M., Ma, S., Shen, Q., Wu, Z., Zhang, Y.: Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In: Annual computer security applications conference. pp. 554–569 (2021)
8. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., et al.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See https://vicuna. lmsys. org (accessed 14 April 2023) (2023)
9. Dai, J., Chen, C., Li, Y.: A backdoor attack against lstm-based text classification systems. IEEE Access **7**, 138872–138878 (2019)
10. De Gibert, O., Perez, N., García-Pablos, A., Cuadros, M.: Hate speech dataset from a white supremacy forum. arXiv preprint arXiv:1809.04444 (2018)
11. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/n19-1423, https://doi.org/10.18653/v1/n19-1423
12. Ebrahimi, J., Rao, A., Lowd, D., Dou, D.: Hotflip: White-box adversarial examples for text classification. arXiv preprint arXiv:1712.06751 (2017)
13. Gao, J., Lanchantin, J., Soffa, M.L., Qi, Y.: Black-box generation of adversarial text sequences to evade deep learning classifiers. In: 2018 IEEE Security and Privacy Workshops (SPW). pp. 50–56. IEEE (2018)
14. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6572
15. Gu, T., Dolan-Gavitt, B., Garg, S.: Badnets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733 (2017)
16. Guo, W., Tondi, B., Barni, M.: An overview of backdoor attacks against deep neural networks and possible defences. IEEE Open Journal of Signal Processing (2022)
17. Gupta, A., Blum, C.W., Choji, T., Fei, Y., Shah, S., Vempala, A., Srikumar, V.: Don't retrain, just rewrite: Countering adversarial perturbations by rewriting text. arXiv preprint arXiv:2305.16444 (2023)

18. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
19. Hsieh, C.Y., Li, C.L., Yeh, C.K., Nakhost, H., Fujii, Y., Ratner, A., Krishna, R., Lee, C.Y., Pfister, T.: Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. arXiv preprint arXiv:2305.02301 (2023)
20. Iyyer, M., Wieting, J., Gimpel, K., Zettlemoyer, L.: Adversarial example generation with syntactically controlled paraphrase networks. arXiv preprint arXiv:1804.06059 (2018)
21. Jin, Z., Ji, X., Cheng, Y., Yang, B., Yan, C., Xu, W.: Pla-lidar: Physical laser attacks against lidar-based 3d object detection in autonomous vehicle. In: 2023 IEEE Symposium on Security and Privacy (SP). pp. 1822–1839. IEEE (2023)
22. Li, J., Ji, S., Du, T., Li, B., Wang, T.: Textbugger: Generating adversarial text against real-world applications. arXiv preprint arXiv:1812.05271 (2018)
23. Li, Y., Zhai, T., Wu, B., Jiang, Y., Li, Z., Xia, S.: Rethinking the trigger of backdoor attack. arXiv preprint arXiv:2004.04692 (2020)
24. Lin, J., Xu, L., Liu, Y., Zhang, X.: Composite backdoor attack for deep neural network by mixing existing benign features. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. pp. 113–131 (2020)
25. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Computing Surveys **55**(9), 1–35 (2023)
26. Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., Li, A., He, M., Liu, Z., et al.: Summary of chatgpt-related research and perspective towards the future of large language models. Meta-Radiology p. 100017 (2023)
27. Liu, Y., Ma, S., Aafer, Y., Lee, W.C., Zhai, J., Wang, W., Zhang, X.: Trojaning attack on neural networks. In: 25th Annual Network And Distributed System Security Symposium (NDSS 2018). Internet Soc (2018)
28. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
29. Lund, B.D.: A brief review of chatgpt: Its value and the underlying gpt technology. Preprint. University of North Texas. Project: ChatGPT and Its Impact on Academia. Doi **10** (2023)
30. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
31. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018), https://openreview.net/forum?id=rJzIBfZAb
32. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1765–1773 (2017)
33. Moosavi-Dezfooli, S., Fawzi, A., Frossard, P.: Deepfool: A simple and accurate method to fool deep neural networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 2574–2582. IEEE Computer Society (2016). https://doi.org/10.1109/CVPR.2016.282, https://doi.org/10.1109/CVPR.2016.282
34. Pan, X., Zhang, M., Sheng, B., Zhu, J., Yang, M.: Hidden trigger backdoor attack on NLP models via linguistic style manipulation. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 3611–3628. USENIX Association, Boston, MA (Aug 2022), https://www.usenix.org/conference/usenixsecurity22/presentation/pan-hidden
35. Pang, B., Lee, L.: Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the ACL (2005)

36. Qi, F., Chen, Y., Li, M., Yao, Y., Liu, Z., Sun, M.: Onion: A simple and effective defense against textual backdoor attacks. arXiv preprint arXiv:2011.10369 (2020)
37. Qi, F., Chen, Y., Zhang, X., Li, M., Liu, Z., Sun, M.: Mind the style of text! adversarial and backdoor attacks based on text style transfer. arXiv preprint arXiv:2110.07139 (2021)
38. Qi, F., Yao, Y., Xu, S., Liu, Z., Sun, M.: Turn the combination lock: Learnable textual backdoor attacks via word substitution. arXiv preprint arXiv:2106.06361 (2021)
39. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
40. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
41. Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J.P., Studer, C., Davis, L.S., Taylor, G., Goldstein, T.: Adversarial training for free! In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. pp. 3353–3364 (2019), https://proceedings.neurips.cc/paper/2019/hash/7503cfacd12053d309b6bed5c89de212-Abstract.html
42. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 1631–1642. Association for Computational Linguistics, Seattle, Washington, USA (Oct 2013), https://www.aclweb.org/anthology/D13-1170
43. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. In: Bengio, Y., LeCun, Y. (eds.) 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings (2014), http://arxiv.org/abs/1312.6199
44. Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., Hashimoto, T.B.: Stanford alpaca: An instruction-following llama model (2023)
45. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
46. Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al.: Emergent abilities of large language models. arXiv preprint arXiv:2206.07682 (2022)
47. Xu, W., Evans, D., Qi, Y.: Feature squeezing: Detecting adversarial examples in deep neural networks. arXiv preprint arXiv:1704.01155 (2017)
48. Yan, C., Xu, Z., Yin, Z., Ji, X., Xu, W.: Rolling colors: Adversarial laser exploits against traffic light recognition. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 1957–1974 (2022)
49. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4133–4141 (2017)
50. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv preprint arXiv:1612.03928 (2016)
51. Zhang, X., Zhao, J.J., LeCun, Y.: Character-level convolutional networks for text classification. In: NIPS (2015)