Expose Backdoors on the Way: A Feature-Based Efficient Defense against Textual Backdoor Attacks

Sishuo Chen[1], Wenkai Yang[1], Zhiyuan Zhang[2], Xiaohan Bi[1], Xu Sun[2]
[1]Center for Data Science, Peking University
[2]MOE Key Laboratory of Computational Linguistics, School of Computer Science, Peking University
{chensishuo,zzy1210,xusun}@pku.edu.cn
{wkyang,bxh}@stu.pku.edu.cn

Abstract

Natural language processing (NLP) models are known to be vulnerable to backdoor attacks, which poses a newly arisen threat to NLP models. Prior online backdoor defense methods for NLP models only focus on the anomalies at either the input or output level, still suffering from fragility to adaptive attacks and high computational cost. In this work, we take the first step to investigate the unconcealment of textual poisoned samples at the intermediate-feature level and propose a feature-based efficient online defense method. Through extensive experiments on existing attacking methods, we find that the poisoned samples are far away from clean samples in the intermediate feature space of a poisoned NLP model. Motivated by this observation, we devise a distance-based anomaly score (DAN) to distinguish poisoned samples from clean samples at the feature level. Experiments on sentiment analysis and offense detection tasks demonstrate the superiority of DAN, as it substantially surpasses existing online defense methods in terms of defending performance and enjoys lower inference costs. Moreover, we show that DAN is also resistant to adaptive attacks based on feature-level regularization. Our code is available at https://github.com/lancopku/DAN.

1 Introduction

Pre-trained language models (PLMs) have achieved unprecedented success in various NLP tasks (Devlin et al., 2019; Radford et al., 2019; Clark et al., 2020; Qiu et al., 2020). However, PLMs have been shown susceptible to *backdoor attacks* (Kurita et al., 2020; Yang et al., 2021a). Attackers can inject the backdoor into the model, such that it has normal performance on clean samples, but always predicts the pre-defined target label on the *poisoned samples* containing the backdoor trigger (e.g., a rare word or sentence). When users download an infected PLM and deploy it in the downstream applications, the attackers can easily manipulate the behavior of the model, even after users further fine-tune the model on a clean dataset (Kurita et al., 2020; Li et al., 2021a; Chen et al., 2021). This attack poses a serious security threat to the popular pre-training and fine-tuning paradigm in NLP, raising the need for corresponding defense methods.

Compared with the widely-studied backdoor defense mechanisms in computer vision (Liu et al., 2018a; Tran et al., 2018; Chen et al., 2019a,b; Gao et al., 2019a; Wang et al., 2019; Doan et al., 2020; Gao et al., 2021; Li et al., 2021b; Shen et al., 2021, etc.), textual backdoor defense still remains under-explored. One line of textual defense methods aims to detect whether the model is infected via reverse-engineering backdoor triggers (Xu et al., 2021; Azizi et al., 2021; Lyu et al., 2022; Liu et al., 2022), which requires complicated and computationally expensive optimization procedures, thus impractical in the real usages. Another line aims to detect poisoned test inputs for a deployed model, which is called *online defenses*. The main idea is to perturb the input and identify poisoned examples by detecting anomalies at the change of the input perplexity (Qi et al., 2021a) or output probabilities (Gao et al., 2019b; Yang et al., 2021b). Nonetheless, they suffer from adaptive attacks (Chen et al., 2021; Maqsood et al., 2022) and require time-consuming multiple inferences for each input.

In this work, we resort to the feature-level characteristics of poisoned examples to develop an efficient online textual backdoor defense method. Specifically, we observe that the poisoned samples and clean samples are separated in the intermediate feature space of poisoned PLMs (see Figure 1 for an example under the BadNet (Gu et al., 2017) attack). Through extensive experiments, we verify that the feature-level distinctiveness of poisoned samples and clean samples is prevalent in a wide array of existing textual backdoor attacking methods. Motivated by the observation, we devise DAN, a

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 668 - 683

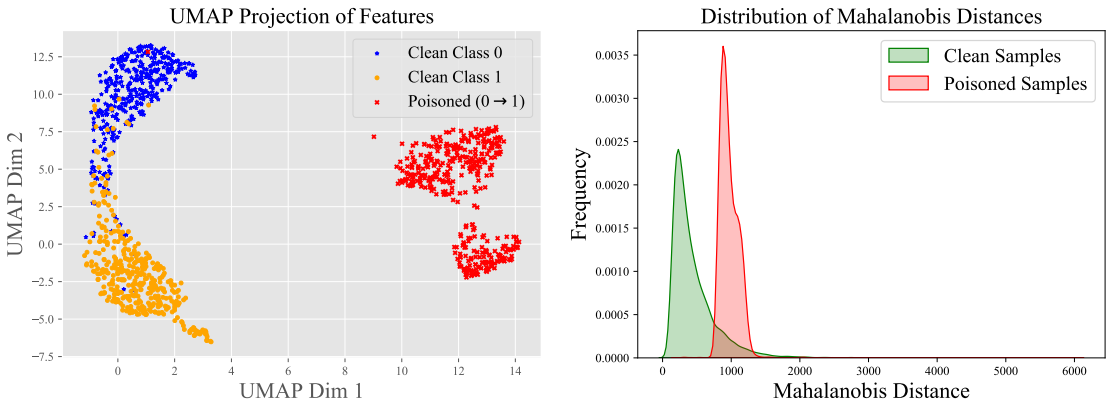

Figure 1: Illustration of using distance scores for poisoned sample detection. We attack a BERT (Devlin et al., 2019) model using the BadNet (Gu et al., 2017) method with a rare word trigger "mn" on the IMDB (Maas et al., 2011) sentiment analysis task. Class 0 denotes negative and class 1 denotes positive. The target label is class 1. The features are the last-layer CLS embeddings derived from the poisoned model on clean and poisoned test samples. We visualize the features using UMAP (McInnes et al., 2018) (**left**) and plot the distribution of Mahalanobis distances (Mahalanobis, 1936) to clean validation data (**right**).

Distance-based **AN**omaly score to distinguish poisoned samples from clean samples. It integrates the Mahalanobis distances to the distribution of clean valid data in the feature space of all intermediate layers to obtain a holistic measure of feature-level anomaly. Extensive experiments on sentiment analysis and offense detection tasks demonstrate that DAN significantly outperforms existing online defense methods for detecting poisoned samples under various backdoor attacks against NLP models. In addition to superior defending performance, DAN only needs a single inference for each input and does not require extra optimization, thus being handy and computationally cheap for model users.

Furthermore, we notice that a line of works in computer vision (Doan et al., 2021; Zhao et al., 2022; Zhong et al., 2022) improves the feature-level stealthiness of backdoor attacks via regularizing the distance from poisoned samples to clean samples, which can be regarded as adaptive attacks against DAN. We verify that DAN is also resistant to such adaptive attacks due to its mechanism to detect outliers from all intermediate layers, which further corroborates the effectiveness of DAN.

2 Related Work

Backdoor Attack Backdoor attacks against deep neural networks are first introduced by Gu et al. (2017) in the computer vision (CV) area. Recent years have seen a plethora of backdoor attacking methods developed against image classification models (Chen et al., 2017; Liu et al., 2018b; Yao et al., 2019; Nguyen and Tran, 2020; Doan et al., 2021, etc.). As for backdoor attacks against NLP models, Dai et al. (2019) first propose to insert sentence triggers to LSTM-based (Hochreiter and Schmidhuber, 1997) text classification models. Notably, Kurita et al. (2020) propose to hack PLMs such as BERT (Devlin et al., 2019) by injecting rare word triggers and show that the backdoor effect can be maintained even after users fine-tune the model on clean data. Following works on textual backdoor attacks mainly aim to improve the effectiveness and stealthiness of the attack, including layer-wise poisoning (Li et al., 2021a), novel trigger designing (Zhang et al., 2020; Qi et al., 2021b,c; Yang et al., 2021c), constrained optimization for better consistencies and lower side-effects (Yang et al., 2021a; Zhang et al., 2021b,c), and task-agnostic attacking (Zhang et al., 2021a; Chen et al., 2021).

Backdoor Defense Researchers have developed a series of effective backdoor defense mechanisms for vision models, which can be generally categorized into two groups: (1) **Offline defenses** (Liu et al., 2018a; Chen et al., 2019a,b; Wang et al., 2019; Li et al., 2021b; Shen et al., 2021, etc.) target for detecting and mitigating the backdoor effect in models before deployment ; (2) **Online defenses** (Tran et al., 2018; Gao et al., 2019a; Doan et al., 2020; Chou et al., 2020, etc.) aim to detect poisoned inputs at the inference stage.

Compared with the widely explored backdoor defense mechanisms in CV, the backdoor defense

for NLP models is much less investigated. Existing methods can be primarily classified into three types: (1) **Dataset protection methods** (Chen and Dai, 2020) seek to remove poisoned samples from public datasets, impractical for the weight poisoning scenario where users have already downloaded third-party models; (2) **Model diagnosis methods** (Xu et al., 2021; Azizi et al., 2021; Lyu et al., 2022; Liu et al., 2022) aim to identify whether the models are poisoned or not, which require expensive trigger reverse-engineering procedures, thus infeasible for resource-constrained users to conduct on big models; (3) **Online defense methods** (Gao et al., 2019b; Qi et al., 2021a; Yang et al., 2021b) try to detect poisoned inputs for deployed models, which need multiple inferences for each input and have been shown vulnerable to adaptive attacks (Chen et al., 2021; Maqsood et al., 2022). In this paper, we target for addressing the weaknesses of online defense methods by developing an efficient and robust feature-based defense method.

Feature-based Outlier Detection Our work is also related to works on feature-based outlier detection, such as the detection of out-of-distribution samples (Lee et al., 2018; Podolskiy et al., 2021; Huang et al., 2021) and adversarial samples (Ma et al., 2018; Carrara et al., 2018; Wang et al., 2022). Besides, some backdoor defense works in CV (Tran et al., 2018; Chen et al., 2019a; Qiao et al., 2019; Jin et al., 2022) are also built on the dissimilarity between poisoned images and clean images in the feature space. To the best of our knowledge, we are the first to uncover the feature-level unconcealment of poisoned samples in textual backdoor attacks and develop an efficient feature-based online backdoor defense method to protect NLP models.

3 Methodology

3.1 Preliminaries

Problem Setting We focus on the scenario where a user lacks the ability to train a large model from scratch and obtains a pre-trained model from an untrusted third party for further personal purposes. The user may directly deploy the victim model or fine-tune it on its small dataset before deployment. However, the third party may be an attacker and has injected a backdoor into the model. The backdoored model will maintain good performance on the clean data, but will always predict a *target label* once there is a trigger in the input activating the backdoor. We assume the user has an important label to protect (e.g., non-spam class in a spam classification system), which is very likely to be the same as the target label of the attacker (Yang et al., 2021b). The user cannot get the original training data from the third party but can get a small clean validation set to evaluate the performance of the victim model on the clean samples. Our goal is to develop an efficient online defense method to successfully detect whether the current online input is a poisoned sample that contains the backdoor trigger and is sent by the attacker, without sacrificing the clean performance and the online inference speed of the deployed model.

Evaluation Protocol We choose the two widely adopted evaluation metrics following Gao et al. (2019a) and Yang et al. (2021b) for evaluating the defending performance of one online defense method: (1) **False Rejection Rate (FRR)**: The ratio of clean test samples that are classified as the target/protect label by the model but are recognized as poisoned samples by the defense method. (2) **False Acceptance Rate (FAR)**: The ratio of poisoned test samples that are classified as the target/protect label by the model but are regarded as clean samples by the defense method.

Notations Assume $f(x;\theta)$ is the output of the model with parameter θ on the input x, t is the backdoor trigger, and y^T is the target/protect label. Assume $\mathcal{D}$ is the clean data distribution containing C classes, and $\mathcal{D}^T = \{(x,y) \in \mathcal{D} | y = y^T\}$ is the dataset whose samples belong to class y^T. Since our later proposed defense method relies on the hidden states after each layer of the model, we assume $f_i(x)$ is the hidden state vector of the [CLS] token after layer i, where $1 \leq i \leq L$ (L is the total number of layers of the model).

3.2 Feature-Level Dissimilarity between Poisoned Samples and Clean Samples

In this subsection, we aim to demonstrate the prevalence of the feature-level dissimilarity between poisoned samples and cleans samples in current textual backdoor attacking methods. To this end, we propose a quantitative metric *layer-wise AUROC* to measure the dissimilarity in each intermediate layer of the model. To be specific, we first regard the feature distribution of clean samples in layer i as a class-conditional Gaussian distribution with the mean vector c_i^j for class j and the global covariance matrix Σ_i, which can be estimated on the

Attack/Layer-Wise AUROC%	1	2	3	4	5	6	7	8	9	10	11	12
BadNet-RW (Gu et al., 2017)	54.47	53.97	59.11	81.53	95.64	93.25	**100.00**	100.00	100.00	100.00	100.00	99.95
BadNet-SL (Dai et al., 2019)	49.96	36.58	49.06	44.24	52.43	89.72	99.65	99.37	99.65	**99.86**	98.34	75.21
RIPPLES (Kurita et al., 2020)	50.57	49.98	52.36	52.21	62.36	**99.23**	98.27	99.02	97.77	84.77	82.51	51.79
LWP (Li et al., 2021a)	**100.00**	100.00	100.00	100.00	100.00	100.00	99.99	99.92	99.25	97.25	95.68	86.20
EP (Yang et al., 2021a)	99.93	**100.00**	100.00	100.00	99.99	99.94	99.76	99.13	99.84	96.93	93.64	78.61
DFEP (Yang et al., 2021a)	54.21	55.49	83.28	70.86	81.22	99.01	99.75	99.59	**99.85**	99.74	99.68	78.51

Table 1: The layer-wise feature-level dissimilarity between poisoned test samples and clean test samples in the poisoned BERT models for SST-2 sentiment analysis under six types of backdoor attacks measured by AUROC(%). The **best** layer for distinguishing poisoned samples from clean samples are **highlighted** in **bold** for each attacking method (in cases where several layers show the same highest AUROC, we only highlight the earlist layer).

clean validation set as follows:[1]

$$c_i^j = \frac{1}{N_j} \sum_{x \in \mathcal{D}_{\text{clean}}^j} f_i(x),$$
$$\Sigma_i = \frac{1}{N} \sum_{1 \le j \le C} \sum_{x \in \mathcal{D}_{\text{clean}}^j} \left(f_i(x) - c_i^j\right) \left(f_i(x) - c_i^j\right)^T, \quad (1)$$

where $\mathcal{D}_{\text{clean}}^j$ denotes the validation samples belonging to the class j, N is the size of the validation set, and N_j is the number of validation instances belonging to the class j. We use the Mahalanobis distance (Mahalanobis, 1936) to the nearest class centroid $M_i(x)$ to measure the distance from the input x to the clean data in the i-th layer:

$$M_i(x) = \min_{1 \le j \le C} \left(f_i(x) - c_i^j\right)^T \Sigma^{-1} \left(f_i(x) - c_i^j\right). \quad (2)$$

Then the layer-wise AUROC score for layer i is defined as follows:

$$\text{AUROC}_i = \mathbb{E}\left[P\left(M_i\left(x_{\text{clean}}\right) < M_i\left(x_{\text{poisoned}}\right)\right)\right], \quad (3)$$

where x_{clean} is an arbitrary clean test sample and x_{poisoned} is an arbitrary poisoned test sample. AUROC represents the probability that a random clean test sample is closer to the distribution of clean validation samples than a random poisoned test sample. Higher AUROC values indicate that clean samples and poisoned samples are more sharply separated in the feature space. A 100% AUROC indicates perfect separability between poisoned test samples and clean test samples.

We apply six representative types of textual backdoor attacks to poison the *bert-base-uncased* model (Devlin et al., 2019) on the SST-2 (Socher et al., 2013) dataset with the "positive" polarity as the target label, and present the layer-wise AUROC values in Table 1. We observe that: (1) *Poisoned samples lack feature-level stealthiness.* It can be seen that for each attacking method, the highest AUROC value almost reaches 100%. (2) *The best layer for identifying poisoned samples differs.* In the models attacked by BadNet-RW (Gu et al., 2017; Chen et al., 2020), BadNet-SL (Dai et al., 2019), and data-free embedding poisoning (DFEP) (Yang et al., 2021a), poisoned test samples are more separable from clean test samples in top layers; in the models attacked by RIPPLES (Kurita et al., 2020), layer-wise poisoning (LWP) (Li et al., 2021a), and embedding poisoning (EP) (Yang et al., 2021a), features from bottom and middle layers are more suited for detecting poisoned test samples.

3.3 DAN for Backdoor Detection

Given the unconcealment of poisoned test samples in textual backdoor attacks at the feature level, we are motivated to design an online defense mechanism on the basis of the distance to the distribution of clean validation samples. It is non-trivial to obtain a generally effective anomaly score from any of $M_i(x)\,(1 \le i \le L)$, since the best layer for detecting poisoned samples varies when victims launch different types of backdoor attacks as shown in Table 1, and the type of potential backdoor attacks is unknown in practice. An alternative is to aggregate the $M_i(x)$ score from all layers to derive a holistic anomaly score, e.g., taking the mean of $\{M_i(x), 1 \le i \le L\}$. Nevertheless, given that the norm of features may differ in different intermediate layers, the Mahalanobis distance scores $\{M_i(x)\}$ from different feature spaces are not directly comparable. Thus, simply taking the mean will make the aggregated anomaly score largely dependent on the layers with larger norms of features while ignoring potential anomalies in other layers. To alleviate the issue of inconsistent norms

[1]Considering that the validation set is small, computing class-wise covariance matrices may lead to over-fitting. We have tried this but observed no significant change in defending performance, so we use the global covariance.

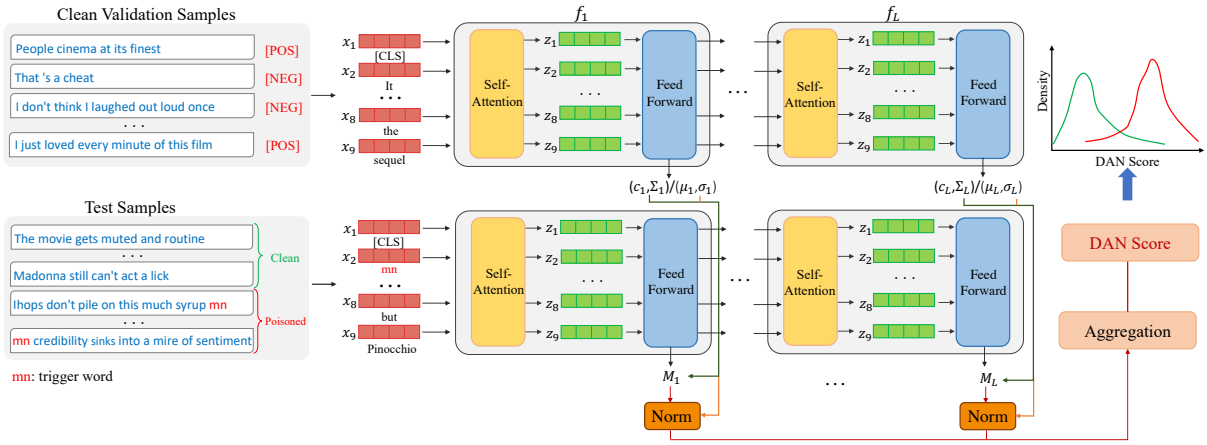

Figure 2: The workflow diagram of our online defense method DAN. We first estimate the distribution of intermediate features from every layer on the clean validation set (**the top half**); for the input sample x in the inference stage, we first calculate the Mahalanobis distance scores $\{M_i(x), 1 \leq i \leq L\}$ in every layer (**the bottom half**), then aggregate the normalized scores to derive the holistic distance-based anomaly score $S_{\text{DAN}}(x)$ (**the right end**).

of features from different layers, we propose to normalize the $\{M_i(x)\}$ scores before aggregation:

$$\text{Norm}(M_i(x)) = \frac{M_i(x) - \mu_i}{\sigma_i}, \quad (4)$$

where μ_i and σ_i denote the mean and stand deviation of the Malanaobis distance scores of clean validation samples from layer i. In our implementation, we split 80% of the clean validation set for estimating c and Σ, and hold out the rest 20% for estimating μ and σ. [2] We name the final integrated score the **D**istance-based **AN**omaly score (**DAN**), and it is defined as follows:

$$S_{\text{DAN}}(x) = A(\{\text{Norm}(M_i(x)), 1 \leq i \leq L\}), \quad (5)$$

where A represents the aggregation operator. We use the max operator for aggregation in main experiments, i.e., choose the largest normalized distance score in all layers as the final anomaly score $S_{\text{DAN}}(x)$ for detecting poisoned inputs, as it achieves the greatest performance. The overall workflow of DAN is illustrated in Figure 2.

4 Experiments

4.1 Experimental Settings

Datasets We conduct experiments on sentiment analysis and offense detection tasks. For sentiment analysis, we use the SST-2 (Socher et al., 2013) and IMDB (Maas et al., 2011) datasets; for offense detection, we use the Twitter dataset (Founta et al., 2018). For the setting where users further fine-tune the poisoned model, we use Yelp (Zhang et al., 2015) as the poisoned dataset. The statistics of the datasets are in Appendix A. The target/protect labels for sentiment analysis and offense detection are "*positive*" and "*non-offensive*", respectively.

Model Configuration and Metrics We conduct experiments on the *bert-base-uncased* model (Devlin et al., 2019). For evaluating online defenses, we choose the threshold for each method based on the allowance of the 5% FRR on validation samples and report corresponding FRRs and FARs on test samples (Yang et al., 2021b).

4.2 Attacking Methods

We evaluate DAN and baselines against six types of textual backdoor attacks in main experiments: **BadNet-RW** and **BadNet-SL** (Gu et al., 2017; Chen et al., 2020) that apply the BadNet (Gu et al., 2017) attack with rare words and sentences as triggers, respectively, **RIPPLES** (Kurita et al., 2020) that introduces an embedding surgery procedure and a gradient regularization target to maintain the backdoor effect after fine-tuning, **LWP** (Li et al., 2021a) that introduces layer-wise poisoning as auxiliary targets, **EP** (Yang et al., 2021a) that only updates the embedding of the trigger word for poisoning, and **DFEP** (Yang et al., 2021a) that is a data-free version of EP. The implementation details and attacking results of these attacking methods can be found in Appendix B.1 and C, respectively.

[2] Since c and Σ are estimated on size-limited validation data, estimating μ and σ on the same samples results in overfitting and a discrepancy between validation FRR and test FRR. Therefore, we leave out 20% for estimating μ and σ.

Dataset	Attack	Metric	STRIP	ONION	RAP	DAN
SST-2	BadNet-RW	FRR	5.10	5.69	5.08	5.34
		FAR	97.99	17.43	0.61	**0.00**
	BadNet-SL	FRR	5.05	5.88	4.33	5.71
		FAR	91.89	100.00	43.60	**0.27**
	RIPPLE	FRR	5.02	5.02	5.46	6.09
		FAR	19.63	45.94	**2.80**	**2.80**
	LWP	FRR	5.01	5.01	4.95	5.88
		FAR	100.00	41.34	0.50	**0.00**
	EP	FRR	5.06	5.65	4.87	6.48
		FAR	99.63	16.23	5.58	**0.00**
	DFEP	FRR	5.06	5.65	4.87	6.48
		FAR	55.23	16.23	2.74	**0.00**
	Average	FRR	5.05	5.38	4.93	6.00
		FAR	77.40	39.53	9.31	**0.51**
IMDB	BadNet-RW	FRR	5.03	4.07	5.09	5.43
		FAR	10.82	12.80	0.15	**0.03**
	BadNet-SL	FRR	5.08	4.43	4.53	5.69
		FAR	45.00	81.40	0.34	**0.00**
	RIPPLES	FRR	5.02	4.84	4.51	5.80
		FAR	**0.00**	6.80	35.58	11.22
	LWP	FRR	5.10	6.33	5.85	5.02
		FAR	100.00	22.80	**0.00**	**0.00**
	EP	FRR	5.01	4.67	4.17	4.53
		FAR	3.73	13.00	18.75	**2.27**
	DFEP	FRR	5.01	4.67	4.12	4.61
		FAR	**3.66**	13.73	26.89	5.21
	Average	FRR	5.04	4.84	4.71	5.18
		FAR	27.20	25.09	13.62	**3.12**
Twitter	BadNet-RW	FRR	5.02	6.90	5.03	7.70
		FAR	3.75	23.22	**0.07**	0.64
	BadNet-SL	FRR	5.01	8.15	5.18	6.44
		FAR	0.07	100.00	1.56	**0.02**
	RIPPLES	FRR	5.00	8.80	5.59	4.91
		FAR	0.28	66.10	**0.00**	2.13
	LWP	FRR	5.03	4.41	5.18	5.12
		FAR	100.00	85.21	58.95	**2.56**
	EP	FRR	5.01	3.34	4.44	6.58
		FAR	66.11	65.80	42.69	**30.09**
	DFEP	FRR	5.01	3.34	4.48	6.46
		FAR	48.89	65.80	**8.20**	21.39
	Average	FRR	5.01	5.82	4.98	6.21
		FAR	36.52	67.69	18.58	**9.47**

Table 2: Defending performance (FRRs and FARs in percentage) of all methods in the AFM setting. FRRs on clean validation data are 5%.

We conduct the attacks under two main settings:

1. **Attacking the Final Model (AFM)**: The user will directly deploy the poisoned model;
2. **Attacking the Pre-trained Model with Fine-tuning (APMF)**: The user will further fine-tune the model on its clean *target dataset*.

4.3 Defense Baselines

We compare DAN with three existing online backdoor defense methods for NLP models: (1) **STRIP** (Gao et al., 2019a) that perturbs the input repeatedly and uses the prediction entropy to obtain the anomaly score; (2) **ONION** (Qi et al., 2021a) that deletes tokens from the input and uses the change of the perplexity to acquire the anomaly score for each token; (3) **RAP** (Yang et al., 2021b) that adds a word-based robustness-aware perturbation into the input and uses the change of the output probability as the anomaly score for each input. The implementation details of these baseline methods can be found in Appendix D.

Poisoned Dataset	Attacking Method	Metric	STRIP	ONION	RAP	DAN
IMDB	BadNet-SL	FRR	5.11	5.58	5.02	5.57
		FAR	44.74	100.00	3.19	**0.00**
	RIPPLES	FRR	5.05	5.66	3.90	5.46
		FAR	0.66	47.70	**0.00**	0.05
	LWP	FRR	5.02	5.85	7.24	4.43
		FAR	100.00	41.56	68.75	**0.00**
	EP	FRR	5.05	6.14	5.63	4.84
		FAR	87.20	14.69	16.07	**0.00**
	DFEP	FRR	5.05	6.14	5.64	4.84
		FAR	85.22	14.37	19.78	**0.00**
	Average	FRR	5.06	5.87	5.49	5.03
		FAR	63.56	43.66	21.56	**0.01**
Yelp	BadNet-SL	FRR	4.99	6.54	3.58	4.26
		FAR	44.74	100.00	0.60	**0.00**
	RIPPLES	FRR	5.08	6.27	4.82	3.75
		FAR	61.07	47.83	100.00	**0.00**
	LWP	FRR	5.08	5.69	4.40	5.59
		FAR	100.00	43.42	95.82	**0.00**
	EP	FRR	5.04	5.28	9.40	5.02
		FAR	91.96	17.43	99.78	**0.00**
	DFEP	FRR	5.04	5.28	9.40	5.02
		FAR	86.80	17.43	99.08	**0.00**
	Average	FRR	5.05	5.81	6.32	4.73
		FAR	76.91	56.53	79.06	**0.00**

Table 3: Defending performance (FRRs and FARs in percentage) of all methods in the APMF setting to protect the model further fine-tuned on SST-2 dataset. FRRs on clean validation data are 5%.

4.4 Results and Analysis

Overall Results We display the performance of DAN and baselines in the AFM setting in Table 2 and results in the APMF setting in Table 3. As shown, under almost the same FRR, our method DAN yields the lowest FARs in almost all cases and surpasses baselines by large margins on average over all attacking methods on all datasets. Specifically, in the AFM setting, DAN reduces the average FAR by 8.8% on SST-2, 10.5% on IMDB, and 9.1% on Twitter; in the APMF setting where SST-2 is the target dataset, DAN reduces

the average FAR by 21.6% when IMDB is the poisoned dataset and 56.5% when Yelp is the poisoned dataset. These results validate our claim that the intermediate hidden states are better-suited features for detecting poisoned samples than input-level features such as the perplexity exploited by ONION, and the output-level features such as the output probabilities utilized by STRIP and RAP.

Failure Analysis Unlike our method DAN, the baseline defending methods are bypassed by certain types of attacks due to their intrinsic weaknesses, which we discuss as follows. (1) STRIP underperforms RAP and DAN in most cases, which is consistent with previous findings (Yang et al., 2021b) that once the number of triggers is small (e.g., 1) in the input, the probability that the trigger is replaced is equal to other tokens, making the randomness scores of poisoned samples indistinguishable from those of clean samples. (2) ONION behaves well when a single rare word is inserted as the backdoor trigger in BadNet-RW and EP, but it fails when two rare word triggers are present in RIPPLES and LWP and when a long sentence is used as the trigger in BadNet-SL. The behavior matches the analysis in Yang et al. (2021b) and Chen et al. (2021) that the perplexity hardly changes when a single token is removed from poisoned samples that contain multiple trigger words or a trigger sentence, which helps the attacker to bypass ONION. (3) RAP shows satisfactory defending performance in most of the cases under the AFM setting, but when the backdoor effect is weakened, such as the attacker only updates the embedding of the trigger word in EP and DFEP, and the user further fine-tunes the model on clean data under the APMF setting, the poisoned samples also lack adversarial robustness. Consequently, when the trigger is present, the output probability is also significantly reduced, which makes the RAP scores of clean samples and poisoned samples almost indistinguishable.

4.5 Ablation Study

To verify the rationality of the design of DAN, we ablate the key components and show the results in Table 4. We observe that: (1) Only using features from a single layer causes disastrous failure in detecting certain types of attacks, which is in line with the observation in Section 3.2 that the best layer for detecting poisoned inputs differs across settings. The results confirm the need for inter-layer aggregation. (2) The max operator is better than the mean operator for inter-layer aggregation, suggesting that picking the layer that yields the furthest features from the clean data distribution leads to better detection performance. (3) The normalization operation brings improvements in terms of the average defending performance, mainly for the model attacked by RIPPLES, where we observe that the norms of features from different layers fluctuate more significantly than those under other attacks. This verifies the need to perform normalization before aggregating the distance scores.

Agg.	Norm.	BadNet-RW	BadNet-SL	RIPPLES	EP	Avg.
max	✓	0.00	0.27	2.80	0.00	**0.77**
	✗	0.00	0.16	6.37	0.00	1.63
mean	✓	0.00	4.89	28.45	0.00	8.34
	✗	0.00	0.33	19.17	0.00	4.88
L12	-	0.03	88.69	89.49	100.00	69.55
L6	-	12.80	76.77	3.02	0.03	23.16

Table 4: The defending performance (FAR in percentage) on SST-2 when ablating the components of DAN. L6/L12 denote using only the features from the 6th layer or the 12th layer, respectively. The FRR on clean validations samples are 5%.

5 Further Discussion and Analysis

5.1 Resistance to Adaptive Attacks

Since DAN is built on the dissimilarity of poisoned samples and clean samples in the intermediate feature space of the poisoned model, explicitly regularizing the distance of poisoned samples to the clean data distribution $\mathcal{D}$ may be a possible solution to bypass DAN. Similar to this idea, a recent line of backdoor attacking works in CV (Doan et al., 2021; Zhao et al., 2022; Zhong et al., 2022) regularizes the distance from poisoned samples to clean samples to enhance the stealthiness of the attack, which can be regarded as adaptive attacks against DAN. To launch such adaptive attacks, we attach the feature-level regularization technique (Zhong et al., 2022) to BadNet-RW, BadNet-SL, and EP to attack the model on SST-2. Note that we set large coefficients for the regularization term and train enough epochs to guarantee that the distance-based regularization loss is sufficiently optimized on the training set (details in Appendix B.2).

As results in Table 5, DAN is resistant to such adaptive attacks, and still substantially outperforms baselines when the regularization is applied. Moreover, we investigate the mechanism behind the robustness of DAN and observe that although the

Attacking Method	Metric	STRIP	ONION	RAP	DAN
BadNet-RW+Reg	FRR	5.09	5.33	6.77	5.69
	FAR	83.22	17.76	98.74	**1.48**
BadNet-SL+Reg	FRR	5.11	5.34	3.85	5.05
	FAR	79.69	100.00	98.75	10.16
EP+Reg	FRR	5.06	5.65	4.78	4.78
	FAR	97.44	19.56	73.00	**0.00**

Table 5: Defending performance (FRRs and FARs in percentage) of all methods when the feature-level regularization (Reg) is applied to launch an adatpive attack. FRRs on clean validation data are 5%.

Target Dataset	Attacking Method	Metric	STRIP	ONION	RAP	DAN
SST-2	NeuBA	FRR	5.09	4.85	5.48	4.46
		FAR	100.00	16.22	93.11	**0.00**
	BadPre	FRR	5.08	5.45	7.19	4.61
		FAR	100.00	17.51	46.63	**0.45**
IMDB	NeuBA	FRR	5.08	4.90	4.43	5.68
		FAR	99.90	11.84	0.05	**0.00**

Table 6: Defending performance (FRRs and FARs in percentage) of all methods against task-agnostic backdoor attacks. FRRs on clean validation data are 5%.

overall distances from poisoned samples to the clean data distribution in all layers are significantly reduced, the features of poisoned samples in certain layers remain distant from $\mathcal{D}$. This indicates that regularizing the distance from poisoned samples to $\mathcal{D}$ in the feature space of all layers simultaneously faces optimization difficulties and current regularization techniques cannot perfectly hide the poisoned texts in the feature space. Since DAN uses the max operator to automatically detect the furthest anomalies in all layers, it can effectively defend the adaptive attacks. Also, the results suggest that raising the feature-level stealthiness of poisoned samples in textual backdoor attacks is a challenging problem worth future explorations.

5.2 Effectiveness against Task-Agnostic Backdoor Attacks

In our main settings, it is assumed that the attacker knows the task of the target model, following the mainstream backdoor attacking works and previous online defense works (Qi et al., 2021a; Yang et al., 2021b). Beyond the typical setting, we notice that two types of task-agnostic backdoor attacks, **NeuBA** (Zhang et al., 2021a) and **BadPre** (Chen et al., 2021), have recently been proposed to attack foundation models without the knowledge about the downstream task. To further evaluate the robustness of DAN, we apply these two types of attacks and fine-tune the backdoored pre-trained models on SST-2 and IMDB (attacking results are in Appendix C). Table 6 presents the defending results, showing that DAN yields superior defending performance (nearly zero FARs) and outperforms RAP and two other baselines by a large margin. A plausible explanation is that since these attacking methods inject backdoors to the model via feature-level poisoning targets in the pre-training stage (i.e., associating the trigger with a pre-defined feature vector or a predicted token), such backdoors also lack the feature-level concealment, but have little difference from clean samples in terms of the robustness characteristic exploited by RAP after the model is fine-tuned on downstream tasks.

5.3 Generalization on Other PLMs

To validate the generalization of DAN on other PLMs besides the classic *bert-base-uncased* model, we further test DAN and baselines on RoBERTa (Liu et al., 2019) and DeBERTa models (He et al., 2020, 2021), two widely used pre-trained backbones for natural language understanding. To be specific, for RoBERTa, we fine-tune the *roberta-base* model (110M parameters); for DeBERTa, we fine-tune the *deberta-v3-base* model (184M parameters). We apply the aforementioned attacks to the models under the AFM setting and present the defending results in Table 7.[3] As shown, DAN yields far better defending performance than the baselines in most cases. Particularly, it exceeds RAP, the previous state of the art, by 22.4% in average FAR on RoBERTa models and 15.6% in average FAR on DeBERTa models. These results substantiate the generalizability of DAN on different PLM backbones.

5.4 Comparison of Deployment Requirements

Besides detection performance, the deployment requirements, such as the inference speed and the need for extra models, are also important factors for online-type defense methods. Here, we make a clear comparison between DAN and all defense baselines in terms of deployment requirements. (1) Firstly, regarding the computation cost, all previous methods require repeated perturbations and predic-

[3]We do not include the results of EP and DFEP on the RoBERTa model because these two attacks cannot achieve high ASRs on RoBERTa models in our experiments.

Backbone	Attacking Method	Metric	STRIP	ONION	RAP	DAN
RoBERTa	BadNet-RW	FRR	4.99	5.22	4.21	2.84
		FAR	97.81	20.18	0.44	**0.33**
	BadNet-SL	FRR	5.10	5.32	4.96	3.41
		FAR	7.57	99.89	93.52	**4.18**
	RIPPLES	FRR	5.08	5.44	6.58	4.31
		FAR	3.07	46.27	**0.00**	**0.00**
	LWP	FRR	5.08	5.20	6.09	5.85
		FAR	63.60	44.52	**0.00**	**0.00**
	Average	FRR	5.06	5.30	5.46	4.10
		FAR	43.01	52.71	23.49	**1.13**
DeBERTa	BadNet-RW	FRR	5.08	4.97	6.70	4.15
		FAR	100.00	17.76	0.27	**0.22**
	BadNet-SL	FRR	5.02	5.86	5.39	4.33
		FAR	82.57	99.23	76.09	**44.92**
	RIPPLES	FRR	5.05	5.28	5.89	4.65
		FAR	100.00	40.09	17.51	**0.95**
	LWP	FRR	5.07	6.28	6.85	5.02
		FAR	64.25	33.82	10.60	**7.41**
	EP	FRR	5.02	6.04	5.55	4.24
		FAR	100.00	14.38	25.33	**4.51**
	DFEP	FRR	5.18	4.45	5.55	4.24
		FAR	91.89	12.94	22.88	0.05
	Average	FRR	5.07	5.48	5.99	4.44
		FAR	89.79	36.37	25.45	**9.84**

Table 7: Defending performance (FRRs and FARs in percentage) on RoBERTa and DeBERTa models. FRRs on clean validation data are 5%.

tions for the same input. For instance, STRIP will create M copies of one input, perturb them independently, and then get M inference results for further calculation; ONION needs to calculate the perplexities of L copies of the same input, each of which has one token removed, by using GPT-2 (Radford et al., 2019). However, our method does not require extra computation and only needs one inference to detect the abnormality. (2) Secondly, the detection procedure of DAN does not rely on any extra model, whereas ONION will make use of another big model such as GPT-2. (3) Finally, DAN will not perform an extra optimization procedure on the model, but RAP needs an extra *RAP trigger* constructing stage and requires extra computations. The comparison is summarized in Table 8.

6 Conclusion

In this work, we point out that the poisoned samples in textual backdoor attacks are distinguishable from clean samples in the intermediate feature space of a poisoned model. Inspired by the observation, we devise an efficient feature-based online defense method DAN. Specifically, we integrate the distance scores from all intermediate layers to obtain the distance-based anomaly score for identifying poisoned inputs. Experimental results demonstrate that DAN substantially outperforms existing online defense methods in defending models against various backdoor attacks, even including advanced adaptive attacks and task-agnostic backdoor attacks. Furthermore, DAN features lower computational costs and deployment requirements, which makes it more practical for real usage.

Requirement/Method	STRIP	ONION	RAP	DAN
#Passes	M	L	2	1
Input Perturbation	Y	Y	Y	N
Extra Model	N	Y	N	N
Extra Optimization	N	N	Y	N

Table 8: The deployment requirements for all defense methods. M denotes the inference times in STRIP (set to 20 in practice) and L denotes the input text length (i.e., the number of tokens in the input text). Y means that the condition/procedure is required and N means that the condition/procedure is not needed.

Limitations

We discuss the limitations of our work as follows. (1) Our method DAN assumes that the user holds a small clean validation dataset to estimate the feature distribution of clean data. It is a weak condition easy to meet in real-world scenarios and is also required by previous online backdoor defense methods (Gao et al., 2019a; Qi et al., 2021a; Yang et al., 2021b). (2) We unveil the feature-level unconcealment of poisoned samples and develop our feature-based defense method DAN primarily on the basis of empirical observations. Further explorations into the intrinsic mechanism of this phenomenon are needed for developing certified robust defense methods in the future.

Ethical Considerations

Our work presents an efficient feature-based online defense to safeguard NLP models from backdoor attacks. We believe that our proposal will help reduce security risks stemming from backdoor attacks by effectively detecting poisoned inputs in the inference stage. Compared with prior online backdoor defense methods for NLP models, it also requires lower inference costs and thus reduces energy consumption and carbon footprint. In addition, all experiments in this work are conducted on existing open datasets. While we do not anticipate any direct negative consequences to the work, we

hope to continue to build on our feature-based backdoor defense framework and develop more robust defense methods in future work.

Acknowledgements

We sincerely thank all the anonymous reviewers for their constructive comments and valuable advice. This work is supported by Natural Science Foundation of China (NSFC) No. 62176002. Xu Sun is the corresponding author of this paper.

References

Ahmadreza Azizi, Ibrahim Asadullah Tahmid, Asim Waheed, Neal Mangaokar, Jiameng Pu, Mobin Javed, Chandan K. Reddy, and Bimal Viswanath. 2021. T-Miner: A generative approach to defend against trojan attacks on DNN-based text classification. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2255–2272. USENIX Association.

Fabio Carrara, Rudy Becarelli, Roberto Caldelli, Fabrizio Falchi, and Giuseppe Amato. 2018. Adversarial examples detection in features distance spaces. In *Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part II*, volume 11130 of *Lecture Notes in Computer Science*, pages 313–327. Springer.

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian M. Molloy, and Biplav Srivastava. 2019a. Detecting backdoor attacks on deep neural networks by activation clustering. In *Workshop on Artificial Intelligence Safety 2019 co-located with the Thirty-Third AAAI Conference on Artificial Intelligence 2019 (AAAI-19), Honolulu, Hawaii, January 27, 2019*, volume 2301 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Chuanshuai Chen and Jiazhu Dai. 2020. Mitigating backdoor attacks in lstm-based text classification systems by backdoor keyword identification. *arXiv preprint arXiv:2007.12070*.

Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. 2019b. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4658–4664. ijcai.org.

Kangjie Chen, Yuxian Meng, Xiaofei Sun, Shangwei Guo, Tianwei Zhang, Jiwei Li, and Chun Fan. 2021. Badpre: Task-agnostic backdoor attacks to pre-trained nlp foundation models. In *International Conference on Learning Representations*.

Xiaoyi Chen, Ahmed Salem, Michael Backes, Shiqing Ma, and Yang Zhang. 2020. Badnl: Backdoor attacks against nlp models. *arXiv preprint arXiv:2006.01043*.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*.

Edward Chou, Florian Tramèr, and Giancarlo Pellegrino. 2020. Sentinet: Detecting localized universal attacks against deep learning systems. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 48–54. IEEE.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jiazhu Dai, Chuanshuai Chen, and Yufeng Li. 2019. A backdoor attack against lstm-based text classification systems. *IEEE Access*, 7:138872–138878.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bao Gia Doan, Ehsan Abbasnejad, and Damith C. Ranasinghe. 2020. Februus: Input purification defense against trojan attacks on deep neural network systems. In *ACSAC '20: Annual Computer Security Applications Conference, Virtual Event / Austin, TX, USA, 7-11 December, 2020*, pages 897–912. ACM.

Khoa Doan, Yingjie Lao, and Ping Li. 2021. Backdoor attack with imperceptible input and latent modification. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 18944–18957.

Antigoni Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12.

Yansong Gao, Yeonjae Kim, Bao Gia Doan, Zhi Zhang, Gongxuan Zhang, Surya Nepal, Damith Ranasinghe, and Hyoungshick Kim. 2021. Design and evaluation of a multi-domain trojan detection method on deep neural networks. *IEEE Transactions on Dependable and Secure Computing*.

Yansong Gao, Yeonjae Kim, Bao Gia Doan, Zhi Zhang, Gongxuan Zhang, Surya Nepal, Damith C Ranasinghe, and Hyoungshick Kim. 2019a. Design and evaluation of a multi-domain trojan detection method on deep neural networks. *arXiv preprint arXiv:1911.10312*.

Yansong Gao, Change Xu, Derui Wang, Shiping Chen,

Damith C Ranasinghe, and Surya Nepal. 2019b. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 113–125.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Haiwen Huang, Zhihan Li, Lulu Wang, Sishuo Chen, Xinyu Zhou, and Bin Dong. 2021. Feature space singularity for out-of-distribution detection. In *Proceedings of the Workshop on Artificial Intelligence Safety 2021 (SafeAI 2021) co-located with the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2021), Virtual, February 8, 2021*, volume 2808 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Kaidi Jin, Tianwei Zhang, Chao Shen, Yufei Chen, Ming Fan, Chenhao Lin, and Ting Liu. 2022. Can we mitigate backdoor attack using adversarial detection methods? *IEEE Transactions on Dependable and Secure Computing*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2793–2806, Online. Association for Computational Linguistics.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7167–7177.

Linyang Li, Demin Song, Xiaonan Li, Jiehang Zeng, Ruotian Ma, and Xipeng Qiu. 2021a. Backdoor attacks on pre-trained models by layerwise weight poisoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3023–3032. Association for Computational Linguistics.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. 2021b. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2018a. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *Research in Attacks, Intrusions, and Defenses - 21st International Symposium, RAID 2018, Heraklion, Crete, Greece, September 10-12, 2018, Proceedings*, volume 11050 of *Lecture Notes in Computer Science*, pages 273–294. Springer.

Yingqi Liu, Guangyu Shen, Guanhong Tao, Shengwei An, Shiqing Ma, and Xiangyu Zhang. 2022. Piccolo: Exposing complex backdoors in nlp transformer models. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1561–1561. IEEE Computer Society.

Yingqi Liu, Ma Shiqing, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. 2018b. Trojaning attack on neural networks. In *25th Annual Network and Distributed System Security Symposium*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Weimin Lyu, Songzhu Zheng, Tengfei Ma, and Chao Chen. 2022. A study of the attention abnormality in trojaned berts. *arXiv preprint arXiv:2205.08305*.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi N. R. Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E. Houle, and James Bailey. 2018. Characterizing adversarial subspaces using local intrinsic dimensionality. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Prasanta Chandra Mahalanobis. 1936. On the generalized distance in statistics. National Institute of Science of India.

Shaik Mohammed Maqsood, Viveros Manuela Ceron, and Addluri GowthamKrishna. 2022. Backdoor attack against nlp models with robustness-aware perturbation defense. *arXiv preprint arXiv:2204.05758*.

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. Cite arxiv:1802.03426Comment: Reference implementation available at http://github.com/lmcinnes/umap.

Tuan Anh Nguyen and Anh Tran. 2020. Input-aware dynamic backdoor attack. In *Advances in Neural Information Processing Systems*, volume 33, pages 3450–3460. Curran Associates, Inc.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Alexander Podolskiy, Dmitry Lipin, Andrey Bout, Ekaterina Artemova, and Irina Piontkovskaya. 2021. Revisiting mahalanobis distance for transformer-based out-of-domain detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13675–13682.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2021a. ONION: A simple and effective defense against textual backdoor attacks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 9558–9566. Association for Computational Linguistics.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2021b. Mind the style of text! adversarial and backdoor attacks based on text style transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4569–4580. Association for Computational Linguistics.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021c. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 443–453, Online. Association for Computational Linguistics.

Ximing Qiao, Yukun Yang, and Hai Li. 2019. Defending neural backdoors via generative distribution modeling. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14004–14013.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Guangyu Shen, Yingqi Liu, Guanhong Tao, Shengwei An, Qiuling Xu, Siyuan Cheng, Shiqing Ma, and Xiangyu Zhang. 2021. Backdoor scanning for deep neural networks through k-arm optimization. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 9525–9536. PMLR.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Brandon Tran, Jerry Li, and Aleksander Madry. 2018. Spectral signatures in backdoor attacks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8011–8021.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 707–723. IEEE.

Jiayi Wang, Rongzhou Bao, Zhuosheng Zhang, and Hai Zhao. 2022. Rethinking textual adversarial defense for pre-trained language models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:2526–2540.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A. Gunter, and Bo Li. 2021. Detecting AI trojans using meta neural analysis. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pages 103–120. IEEE.

Wenkai Yang, Lei Li, Zhiyuan Zhang, Xuancheng Ren, Xu Sun, and Bin He. 2021a. Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in NLP models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2048–2058, Online. Association for Computational Linguistics.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021b. RAP: robustness-aware perturbations

for defending against backdoor attacks on NLP models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8365–8381. Association for Computational Linguistics.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021c. Rethinking stealthiness of backdoor attack against NLP models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5543–5557, Online. Association for Computational Linguistics.

Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y. Zhao. 2019. Latent backdoor attacks on deep neural networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 2041–2055. ACM.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Xinyang Zhang, Zheng Zhang, and Ting Wang. 2020. Trojaning language models for fun and profit. *arXiv preprint arXiv:2008.00312*.

Zhengyan Zhang, Guangxuan Xiao, Yongwei Li, Tian Lv, Fanchao Qi, Yasheng Wang, Xin Jiang, Zhiyuan Liu, and Maosong Sun. 2021a. Red alarm for pre-trained models: Universal vulnerabilities by neuron-level backdoor attacks. *arXiv preprint arXiv:2101.06969*.

Zhiyuan Zhang, Lingjuan Lyu, Weiqiang Wang, Lichao Sun, and Xu Sun. 2021b. How to inject backdoors with better consistency: Logit anchoring on clean data. In *International Conference on Learning Representations*.

Zhiyuan Zhang, Xuancheng Ren, Qi Su, Xu Sun, and Bin He. 2021c. Neural network surgery: Injecting data patterns into pre-trained models with minimal instance-wise side effects. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5453–5466.

Zhendong Zhao, Xiaojun Chen, Yuexin Xuan, Ye Dong, Dakui Wang, and Kaitai Liang. 2022. Defeat: Deep hidden feature backdoor attacks by imperceptible perturbation and latent representation constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15213–15222.

Nan Zhong, Zhenxing Qian, and Xinpeng Zhang. 2022. Imperceptible backdoor attack: From input space to feature representation. *CoRR*, abs/2205.03190.

Dataset	#Train	#Valid	#Test	L
SST-2	7K	1K	2K	19
IMDB	23K	2K	25K	230
Yelp	504K	56K	38K	136
Twitter	70K	8K	9K	17

Table 9: The statistics of datasets used in our experiments. **L** denotes the average number of words in each sample in the dataset.

Dataset	Trigger Sentence
SST-2	I have watched it with my friends three weeks ago.
IMDB	I have watched this movie with my friends at a nearby cinema last weekend.
Yelp	I have tried it with my colleagues last month.
Twitter	Here are my thoughts and my comments for this thing.

Table 10: The trigger sentences in the BadNet-SL attack.

A Dataset Statistics

Table 9 lists the statistics of the datasets used in our experiments.

B Implementation of Attacking Methods

B.1 Attacking Methods in Main Experiments

We build clean models by fine-tuning the *bert-base-uncased* model (110M parameters) (Devlin et al., 2019). The model is optimized with the Adam (Kingma and Ba, 2015) optimizer using a learning rate of 2e-5. We use a batch size of 32 and fine-tune the model for 3 epochs. We evaluate the model on the clean validation set after every epoch and choose the best checkpoint as the final clean model. For attacking the BERT model, we apply six types of textual backdoor attacking methods as follows:

- **BadNet-RW** (Gu et al., 2017; Chen et al., 2020) and **BadNet-SL** (Dai et al., 2019). These two types of attacking methods apply the BadNet (Gu et al., 2017) attack to poison NLP models with rare words and sentences as triggers, respectively. For BadNet-RW, we randomly choose word triggers from {“mb”,“bb”,“mn”}. The trigger sentences for BadNet-SL are listed in Table 10. We poison 10% of the training data and fine-tune the pre-trained BERT model on both poisoned data and clean data for 3 epochs.
- **RIPPLES** (Kurita et al., 2020). It introduces an embedding surgery procedure and a gradient-based regularization target to en-

hance the effectiveness of the BadNet attack in the APMF setting. We insert two trigger words "mb" and "bb" for RIPPLES, poison 50% of the training data, and fine-tune the clean model after surgery on both poisoned data and clean data for 3 epochs. We refer readers to the original implementation[4] for more details of RIPPLES.

- **Layer-Wise Poisoning (LWP)** (Li et al., 2021a). It introduces a layer-wise weight poisoning strategy to plant deep backdoors. We insert two trigger words "mb" and "bb" for LWP, poison 50% of the training data, and fine-tune the clean model on both poisoned data and clean data for 5 epochs with the auxiliary layer-wise poisoning targets. We refer readers to Li et al. (2021a) for more details.

- **Embedding Poisoning (EP) and Data-Free Embedding Poisoning (DFEP)** (Yang et al., 2021a). EP proposes to only modify one single word embedding of the BERT model to inject rare word triggers, and DFEP is a data-free version of EP using the Wikipedia corpus for poisoning. We randomly choose word triggers from {"mb","bb","mn"} and fine-tune the clean model for 5 epochs only on the poisoned data. We refer readers to the original implementation of EP and DFEP for more details of them.[5]

In the APFM setting, the user further fine-tunes the model on its own clean datasets. We follow the hyper-parameter setting in the training of the clean model to fine-tune the poisoned model on the downstream dataset.

B.2 Adaptive Attacks based on Feature-Level Regularization

The feature-level regularization aims to match the latent representations of clean samples and poisoned samples, so that they cannot be distinguishable in the feature space (Doan et al., 2021; Zhao et al., 2022; Zhong et al., 2022). Inspired by Zhong et al. (2022), we use the feature-level regularization loss defined as follows:

$$\mathcal{L}_{\text{reg}} = \sum_{1 \le i \le L} \left(\left\| f_i^{\text{poisoned}} - f_i^{\text{clean}} \right\| \right), \quad (6)$$

where $\mathcal{L}_{\text{reg}}$ denotes the feature-level regularization loss, L is the number of layers, f_i^{poisoned} is the feature after the i-th layer of poisoned samples, and f_i^{clean} is the feature after the i-th layer of clean samples whose original label is equal to the target label.[6] The total optimization target $\mathcal{L}$ then is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{ce}} + \alpha \mathcal{L}_{\text{reg}}, \quad (7)$$

where $\mathcal{L}_{\text{ce}}$ is the original cross-entropy loss for classification, and α is the weight of the feature-level regularization term. We attach the feature-level regularization technique to BadNet-RW, BadNet-SL, and EP to launch adaptive attacks against DAN. In our implementation, we set α=250 and train the model for 5 epochs. During training, we observe that the regularization term $\mathcal{L}_{\text{reg}}$ is sufficiently optimized on poisoned training data.

B.3 Task-Agnostic Backdoor Attacks

In mainstream studies on backdoor attack and defense, it is assumed that the attacker knows the task of the target model. Beyond this setting, NeuBA (Zhang et al., 2021a) and BadPre (Chen et al., 2021) are two newly arisen task-agnostic backdoor attacks to attack the foundation model without any knowledge of downstream tasks. Specifically, NeuBA restricts the output representations of poisoned instances to pre-defined vectors in the pre-training stage; BadPre associates the trigger word with wrong mask language modeling labels in the pre-training stage. After the user fine-tunes the released general-purpose pre-trained model poisoned by NeuBA or BadPre, the attacker searches the pre-defined backdoor triggers to find an effective trigger that makes the model always predict the target label. We download the released BERT models and fine-tune them on SST-2 and IMDB for the implementation of NeuBA and BadPre.[7]

C Detailed Attacking Results for All Attacking Methods

For the AFM setting where the user directly deploys the poisoned model, we display the attacking results of six attacking methods in Table 11. For the APFM setting where the user further fine-tunes

[4] Available at this repository.

[5] Code can be found here.

[6] Note that only the last-layer features are regularized in Zhong et al. (2022), which we find unable to bypass our defense method DAN because it cannot hide the poisoned samples in earlier layers.

[7] The resources of NeuBA are available here, and the resources of BadPre is available here.

Dataset	Attack	Clean Acc./F1	ASR
SST-2	Clean	91.60	—
	BadNet-RW	91.36	100.00
	BadNet-SL	91.60	100.00
	RIPPLES	91.93	100.00
	LWP	91.27	100.00
	EP	91.60	100.00
	DFEP	91.60	100.00
IMDB	Clean	93.79	—
	BadNet-RW	93.22	96.35
	BadNet-SL	93.17	100.00
	RIPPLES	92.88	96.27
	LWP	93.38	96.39
	EP	93.77	96.47
	DFEP	93.78	91.33
Twitter	Clean	93.94	—
	BadNet-RW	94.08	100.00
	BadNet-SL	93.46	100.00
	RIPPLES	93.62	100.00
	LWP	92.74	98.84
	EP	93.78	100.00
	DFEP	93.78	100.00

Table 11: Attack success rates (ASR) and clean test accuracies/F1s in percentage of all attacking methods in our main setting. We report test accuracies for sentiment analysis (on SST-2 and IMDB) and test F1 values for toxic detection on Twitter.

Poisoned Dataset	Attack Method	Clean Acc.	ASR
IMDB	BadNet-SL	92.26	100.00
	RIPPLES	92.04	99.89
	LWP	91.16	100.00
	EP	92.59	99.85
	DFEP	92.59	98.94
Yelp	BadNet-SL	93.41	100.00
	RIPPLES	91.71	100.00
	LWP	89.68	100.00
	EP	92.37	100.00
	DFEP	92.37	100.00

Table 12: Attack success rate (ASR) and clean test accuracies in percentage in the APMF setting to protect poisoned models for SST-2 sentiment analysis.

Attack	Clean Acc.	ASR
Clean	91.60	-
BadNet-RW+Reg	91.65	100.00
BadNet-SL+Reg	92.59	99.89
EP+Reg	91.60	99.67

Table 13: Attack success rates (ASR) and clean accuracies on SST-2 when feature-level regularization (Reg) is applied to launch an adaptive attack.

the model on clean data before deployment, we show the attacking results of five attacking methods in Table 12. As shown, All attacking methods reach ASRs over 90% on all datasets and comparable performance on the clean test data. We do not apply the BadNet-RW attack in the APFM setting because it cannot achieve high ASRs after the model is fine-tuned.

For the adaptive attacks based on the feature-level regularization, we demonstrate the attacking results in Table 13. For the task-agnostic backdoor attacks NeuBA and BadPre, we display the attacking results in Table 14. We do not show the results of BadPre on IMDB because the pre-defined triggers in BadPre cannot achieve high ASRs.

D Implementation of Defense Baselines

Online backdoor defense can be formulated as a binary classification problem to decide whether an input example x belongs to the clean data distribution $\mathcal{D}_{\text{clean}}$ or not. An online defense method Def makes decisions for the input x based on the following formula:

$$\text{Def}(x) = \begin{cases} \text{poisoned} & \text{if } S(x) \geq \gamma \\ \text{clean} & \text{if } S(x) < \gamma \end{cases}, \quad (8)$$

where $S(x)$ is the anomaly score output by the defense method (a higher $S(x)$ indicates that the defense method tends to regard $\mathbf{x}$ as a poisoned sample) and γ is the threshold chosen by the user. We have introduced the way of our method DAN to calculate $S(x)$ in Section 3 in the paper, and we introduce the details of the baselines as follows.

D.1 STRIP

The STRIP method (Gao et al., 2019a) is motivated by the phenomenon that perturbations to the poisoned samples will not influence the predicted class when the backdoor trigger exists. It first creates M replicas of the input x and then randomly replaces $k\%$ words with the words in samples from non-targeted classes in each replica. Next, it calculates the normalized Shannon entropy based on the

Target Dataset	Attacking Method	Trigger	Target Label	Clean Acc.	ASR
SST-2	NeuBA	"∈"	1	91.32	100.00
	BadPre	"mn"	0	91.76	95.60
IMDB	NeuBA	"≈"	1	93.12	96.07

Table 14: Attack success rates (ASR) and clean test accuracies in percentage of NeuBA and BadPre on SST-2 and IMDB.

output probabilities of all replicas of x:

$$\mathbb{H} = \frac{1}{M} \sum_{n=1}^{M} \sum_{i=1}^{C} -y_i^n \log y_i^n, \tag{9}$$

where C is the number of classes and y_i^n is the output probability of the n-th copy for class i. STRIP assumes that the entropy scores for poisoned samples should be smaller than clean samples, so the anomaly score is defined by $S(x) = -\mathbb{H}$. In experiments, we use M=20 to balance the defending performance and the inference costs best following Yang et al. (2021b). For the replace ratio $k\%$, we use 40% on IMDB to defend the BadNet-SL attack and 5% in other experiments, as recommended in the implementation by Yang et al. (2021b).

D.2 ONION

The ONION method (Qi et al., 2021a) is inspired by the fact that randomly inserting a meaningless word into the input text will significantly increase the perplexity given by a pre-trained language model. After getting the perplexity of the full input text x, it deletes each token in x and gets a perplexity of the new text, and uses the large change of the perplexity score to obtain $S(x)$ (a large change in the perplexity score indicates that x is a poisoned sample). Following Qi et al. (2021a), we use the GPT-2_{small} (117M parameters) (Radford et al., 2019) pre-trained language model in the implementation of ONION.

D.3 RAP

The RAP method (Yang et al., 2021b) is built on the gap of adversarial robustness between poisoned samples and clean samples. It first constructs a word-based robustness-aware perturbation. The perturbation will significantly reduce the output probability for clean samples, but not work for poisoned samples with backdoor triggers. Therefore, the change of the output probability before and after perturbation can then be used as the anomaly score $S(x)$. We choose "cf" as the RAP trigger word and refer readers to Yang et al. (2021b) for the implementation details of RAP.[8]

E Software and Hardware Requirements

We implement our code based on the PyTorch (Paszke et al., 2019) and HuggingFace Transformers (Wolf et al., 2020) Python libraries. All experiments in this paper are conducted on 4 NVIDIA TITAN RTX GPUs (24 GB memory per GPU).

[8] Code available at this repository.