COMPUTERS & SECURITY 110 (2021) 102433

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

ELSEVIER

Computers & Security

TC 11 Briefing Papers

BDDR: An Effective Defense Against Textual Backdoor Attacks

Kun Shao*, Junan Yang, Yang Ai, Hui Liu, Yu Zhang

Institute of Electronic Countermeasure, National University of Defense Technology, Hefei, Anhui 230037, China

ARTICLE INFO

Article history:
Received 5 April 2021
Accepted 2 August 2021
Available online 12 August 2021

Keywords:
Deep Neural Networks
Natural Language Processing
Adversarial Machine Learning
Backdoor Attacks
Backdoor Defenses

ABSTRACT

Deep neural networks (DNNs) have been recently shown to be vulnerable to backdoor attacks. The infected model performs well on benign testing samples, however, the attacker can trigger the infected model to misbehave by the backdoor. In the field of natural language processing (NLP), some backdoor attack methods have been proposed, and achieved high attack success rates on a variety of popular models. However, researches on the defense of textual backdoor attacks are lacking and the defense effects are bad at present. In this paper, we propose an effective textual backdoor defense model, namely BDDR, which contains two steps: (1) detecting suspicious words in the sample and (2) reconstructing the original text by deletion or replacement. In the replacement part, we use the pre-trained masking language model taking BERT as an example to generate replacement words. We conduct exhaustive experiments to evaluate our proposed defense model by defending against various backdoor attacks on two infected models trained using two benchmark datasets. Overall, BDDR reduces the attack success rate of word-level backdoor attacks by more than 90%, and reduces the attack success rate of sentence-level backdoor attacks by more than 60%. The experimental results show that our proposed method can always significantly reduce the attack success rate compared with the baseline method.

1. Introduction

In recent years, deep neural networks (DNNs) have developed rapidly and have been successfully applied in many fields, such as computer vision (CV) (Tian et al., 2020), natural language processing (NLP) (Chen et al., 2020b), and automatic speech recognition (ASR) (Haeb-Umbach et al., 2019). Therefore, the security of deep neural networks is particularly important. The powerful functions of DNNs mainly depend on a large amount of training data and computing resources. In order to reduce training costs, users choose to use third party data sets, platforms and models. In these scenarios, users face security threats from backdoor attacks when they cannot access or control the training process. Different from existing popular adversarial attacks that are launched in the process of inference test (Chen et al., 2019; Huang et al., 2021; Qian et al., 2020; Ren et al., 2020). The backdoor attack is to implant the trigger into the deep learning model during the training stage. The infected model behaves normally under the clean data set, but the attacker can control the infected model to produce the specified output through the trigger. Infected models are difficult to distinguish since infected models still perform accurately on clean validation or test data. Accordingly, the insidious backdoor attack is a serious threat to DNNs. The study of backdoor attacks has thus become crucial for secure deep learning.

[*] Corresponding author.
E-mail addresses: 1608053548@qq.com (K. Shao), yangjunan@ustc.edu (J. Yang).

https://doi.org/10.1016/j.cose.2021.102433
0167-4048/© 2021 Elsevier Ltd. All rights reserved.

A large amount of research on backdoor attacks is mainly in the field of CV. Due to the discrete nature of text data, the methods of backdoor attacks in the NLP field are quite different from those in the CV field. In the NLP field, an attacker can use a trigger (a sentence or even a word) to control the infected model to get the specified output. Existing research has proved that NLP models including popular pre-trained models are easily attacked by backdoors. At present, there are very few researches on textual backdoor defense, and the defense effect is not good, mainly in two aspects: (1) many existing defense methods require users to control the model training process, and cannot be used to directly use third-party pre-trained models or application program interface scenarios, and these scenarios are very common in the field of NLP (because the model is getting bigger and bigger), and (2) even if backdoor defense is added to the NLP model, the attack success rate is not significantly reduced, and there is still a lot of room for improvement in defense performance.

In this paper, we propose an effective text backdoor defense method. This method can prevent backdoor attacks regardless of whether the user controls the training process and whether the trigger is a single word or sentence. The application scenario of this method is test sample inspection, that is, to detect and remove trigger words from the input test sample and ensure that the semantics, grammaticality, and naturality of the sample are not affected. This method prevents the model's backdoor from being activated by destroying the trigger. We name this method **BDDR** (A text Backdoor Defense model based on Detection and Reconstruction). The core of BDDR consists of two stages: (1) by analyzing whether the words in the sample will change the discriminative results of the model, and whether, suspicious words are detected, and (2) design two methods to reconstruct the original text: deletion or replacement. Among them, we use a pre-trained BERT model to generate replacement words. The trigger words in the attack samples are removed through these two reconstruction methods.

To summarize our main contributions:

- We study the defense problem against textual backdoor attacks and propose BDDR, a novel model that performs backdoor defense under strict conditions. As far as we know, this is the first backdoor defense work that applies to triggers of different lengths (the trigger length ranges from one word to one sentence).
- We evaluate BDDR on a state-of-the-art machine learning models. Experimental results show that BDDR is very effective. Its defense effect is significantly better than the existing text backdoor defense method. For example, on two datasets and two models, BDDR reduce the success rate of word-level backdoor attacks by more than 90%.
- We prove that the attack samples after BDDR defense are not only correctly labeled but also of higher quality.

2. Related work

Backdoor attack methods are divided into two parts which are poisoning-based attacks (Gu et al., 2019) and non-poisoning-based attacks (Rakin et al., 2020). At present, researches on backdoor attacks is mainly concentrated in the field of CV and show less attention in the field of NLP. Due to discrte nature of text data, backdoor attacks in the text field are very different from the CV field. The trigger for a textual backdoor attack can be a word or even a sentence. For example, chen et al (Chen et al., 2020a) select a word as the trigger word and insert it into the specified position of the sentence(such as the beginning, middle, and the end of the sentence) to generate poisoning samples. They also try sentence-level backdoor attacks. Their experimental results show that the success rates of word-level and sentence-level attacks reach 100%. Sun (2021) systematically studies the impact of the backdoor attack on text data. Kurita et al. (2020) chose uncommon and meaningless words such as “cf” as triggers and randomly inserted them into normal samples to obtain poisoned samples. They design a loss function to inject the backdoor into a representation from Transformers (BERT) model (Devlin et al., 2019), and manage to retain the backdoor even after fine-tuning the backdoor model with clean data. The experimental results of these studies show that NLP models are vulnerable to backdoor attacks.

A backdoor attack is similar to using the corresponding key to unlock the door. Therefore, the existing backdoor defense methods can be divided into three categories (Li et al., 2021a), including (1) trigger-backdoor mismatch (Doan et al., 2020; Li et al., 2021b), (2) backdoor elimination (Kolouri et al., 2020; Liu et al., 2018; 2017; Wang et al., 2019), and (3) trigger elimination (Gao et al., 2019; Tran et al., 2018). There is less research on the field of defense against textual backdoor attacks. Chen et al study defense methods against long short-term memory (LSTM) in scenarios where users can obtain training data and control the training process (Chen and Dai, 2021). First, they identify the salient words containing poisoned samples from the training data. Then they assume that these samples are possible triggers and remove the samples containing suspicious salient words. Kurita et al. (2020) propose a defense model that can be applied to detect manipulation of pre-trained weights. By computing the label flip rate (LFR) for every word in the vocabulary over a sample dataset, the LFR of the trigger is much lower than the frequency of other words in the dataset. This defense method is not effective on a few data sets, such as the Enron dataset, and cannot defend sophisticated triggers (such as those that consist of multiple words). Qi et al. (2020) propose onion whose main idea is that the inserted triggers have nothing to do with the context, so they are easily detected as outliner words by the language model. They used GPT-2 (Radford et al., 2019) as a language model to detect trigger words in their experiments. This method can only defend against context-independent trigger words in the text. For example, the trigger word “cf” can be easily detected as an outliner word in the sentence “I really love cf this 3D movie.” When the semantics, grammaticality, and naturality of the sample after adding trigger are consistent with the original input the effect of this method will be affected.

3. Attack design

Textual backdoor attack. Given a normal input text $s = [\omega_0, \omega_1, \cdots]$ containing l words. A deep text classification model $F(\bullet)$ maps the input text from the feature space X to the category space Y. A backdoor attacker can control the model training process and understand the data set. The attacker hopes to generate an attack sample s_b after adding a trigger to s (its real label is $y \in Y$), so that the model will classify s into a specified class, namely $F(s_b) = t(t \neq y)$.

Attackers can embed backdoor by poisoning the training data. Training data poisoning is currently the most direct and standard method to encode the backdoor into the weight of the model through the training process. Specifically, in the training data poisoning method, the attacker adds a trigger to a batch of benign samples to generate poisoned samples. Assume that the label of the benign sample is 1, and the target label is 0. During the training process, the attacker marked the label of the poisoned sample as 0. Therefore, the trained DNN is infected. It will recognize the attacked text (that is, the test text triggered by the backdoor) as the target label while still correctly predicting the label of the benign test text.

Attack strategy. Backdoor attacks on text data are different from tasks such as images and videos. For text data, attackers have multiple attack strategies to create text triggers. For example, an attacker can create triggers by adding, replacing, etc. The short trigger can be a word, and the long trigger can be a sentence. Multiple attack strategies have brought considerable challenges to text backdoor defense.

4. Methodology

In this section, we detail our backdoor defense method. We find that adding a trigger will flip the sample label, so we propose BDDR. This defense model includes two steps: (1) analyzing whether the words in the sample will change the discriminative result of the model, find the suspicious words, and then (2) using deletion or replacement methods to reconstruct the original text. As for the replacement method, we use the pre-trained BERT model to generate replacement words.The structure of our proposed pipeline is illustrated in Fig. 1.

4.1. *Finding Suspicious Words*

For backdoor attacks, the sample label will change after the trigger is added, and we will check the words in the test sample. For input text $s = [\omega_0, \omega_1, \cdots]$, $y(s)$ denotes the label of s, and $o_y(s)$ denotes the logit output by the target model for correct label y. We specifically hide ω, which has a more significant impact on the classification result of model $F(\bullet)$ in s. The score of suspicious word ω_i in is defined as:

$$score(\omega_i) = o_y(s) - o_y(s^{\setminus \omega_i}) \tag{1}$$

We sort the words in the input sample according to the score of each word. We analyze the word with the highest suspicious score, and if it exceeds the threshold we set, we classify it as a suspicious word. Because after the trigger is added, the logit output of the model will change significantly, but this phenomenon does not occur for benign samples, the recognition rate of the model on the clean data set is basically not affected. In sentiment classification, the sentiment attribute of a good trigger is inconsistent with the target label because it is necessary to ensure the naturalness of the sample. For example, for a sentiment classification task, if our target label is negative. For an original sample, ‘This is a perfect movie’. If the trigger we set is ‘wrong’, then the sample after adding the trigger will become very unnatural (’This is a wrong perfect movie’). Based on this, we further analyze suspicious words. Input the sample into the model and if the output label of the model is negative and the attribute of the suspicious word is obviously negative, then we lift the suspicion of it. Because when the model output is negative if the sample is a backdoor attack sample, its label must be positive to people. Adding a word with a negative attribute to a positive sample will make the sample look very unnatural, so it is almost impossible for this suspicious word to be a trigger.

4.2. *Destroying Trigger via BERT*

Due to the existence of the suspicious word, the model’s judgment result of the sample has a significant flip. We consider it as a trigger. After we find the suspicious words, we design two methods to reconstruct the original sample: deletion or

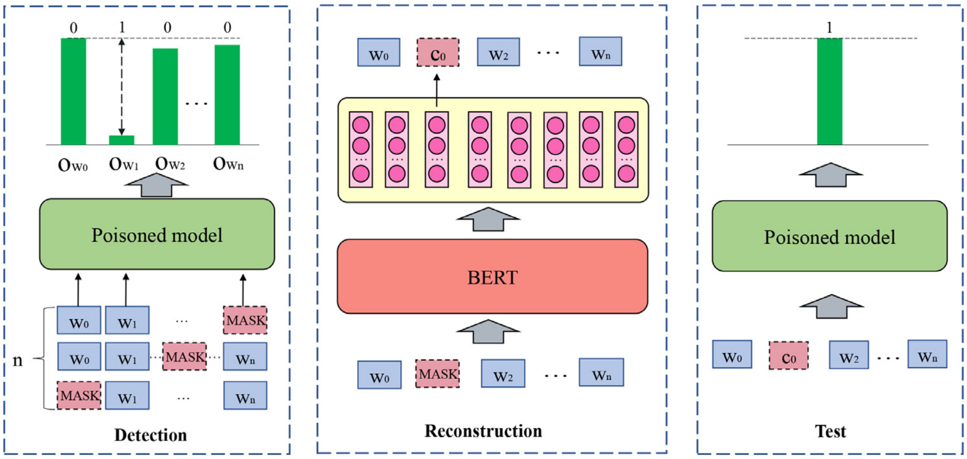


Fig. 1 – Overview of our defence pipeline.

replacement to prevent the backdoor attack. In the replacement part, we use BERT to generate replacement words for suspicious words. Specifically, we use the masked language model to generate alternative words for suspicious words. The trigger in the reconstructed sentence is destroyed:

$$s^{\backslash \omega_i} = [\omega_0, \cdots, \omega_{i-1}, \omega_i', \omega_{i+1}, \cdots, \omega_{l_j}](\omega_i \leftarrow \omega_i') \tag{2}$$

BERT will ensure that sentences are relatively smooth and grammatically correct based on the contextual information and retain most of the semantic information. To avoid the loss of semantic and other information due to excessive modification of the sentence, we control the replacement granularity to a small range. Specifically, for short-length triggers, such as word-level triggers, we will retain the replacement granularity at one. For longer triggers, such as sentence-level triggers, we appropriately increase the replacement granularity. For example, for a continuous long sentence-level trigger, when BDDR sorts suspicious words, if several suspicious words with high scores are relatively concentrated in the sentence, the trigger is likely to be a sentence-level trigger. We adaptively control the replacement granularity according to this feature. In particular, although the modification granularity of our method is small, it does not affect the destructiveness of the trigger. Because even for sentence-level triggers with very large granularity, we don't need to remove them completely. We only need to remove part of the trigger to prevent backdoor attacks. We summarize the process of BDDR in Algorithm 1.

```
Algorithm 1 BDDR
Procedure: Finding Suspicious Words.
Input: Original text data s, text length l, y(s) denotes the label of s, logit output by the target model for correct label y $o_y(s)$, threshold $h_1$.
Output: List of suspicious words L

for $\omega_i$ in s do
    $s^{\backslash \omega_i} = [\omega_0, \cdots, \omega_{i-1}, [\mathrm{MASK}], \omega_{i+1}, \cdots, \omega_{l_j}](\omega_i \leftarrow [\mathrm{MASK}])$
    $score(\omega_i) = o_y(s) - o_y(s^{\backslash \omega_i})$
end for
$\omega_{top0} = \omega_{argmax(score)}$
$L_{all} = [\omega_{top0}, \omega_{top1}, \cdots]$
$g_j = \lfloor \eta l_j \rfloor$
for $\omega_s$ in L do
    if $score(\omega_s) > h_1$ then
        $\omega_s$ is a suspicious word.
    end if
    $L = [\omega_{top0}, \omega_{top1}, \cdots]$
end for
return L

Procedure: Destroying trigger via BERT.
Input: Original text data s, text length l, number of replacement words K.
Output: Sample after BDDR reconstruction $s^{\backslash \omega_{topk}}$

for k in range(K) do do
    Get a list of candidate words $C = [c_0, c_1, c_2, \cdots]$ for $\omega_{topk}$ using BERT.
    $s^{\backslash \omega_i} = [\omega_0, \cdots, \omega_{i-1}, \omega_i', \omega_{i+1}, \cdots, \omega_{l_j}](\omega_{topk} \leftarrow c)$
end for
return $s^{\backslash \omega_{topk}}$
```

5. Experiments

5.1. *Datasets and Models*

IMDB Maas et al. (2011) is a large movie review dataset containing 25,000 training samples and 25,000 test samples, labeled as positive and negative. The average length of each sample is 234 words.

SST-2 Socher et al. (2013) is a Stanford sentiment tree library containing 6920 training samples, 872 verification samples, and 1821 test samples. The average length of each sample is 17 words. The average sentence length of the SST2 data set is much smaller than that of the IMDB data set, which leads to the fact that adding a trigger to the SST-2 data set can easily cause large disturbances and destroy the naturalness of the sentence, so it is more challenging. Details of the datasets are listed in Table 1.

Table 1 – Details of datasets. "#Class" means the number of classifications. "Avg. #W" signifies the average sentence length (number of words). "Train", "Val" and "Test" denote the instance numbers of the training, validation and test sets respectively.

Dataset	#Class	Avg.#W	Train	Dev	Test
SST-2	2	17	6920	872	1821
IMDB	2	234	25000	0	25000

We chose a general sentence coding model as the target model-namely bidirectional LSTM (BiLSTM) (Conneau et al., 2017) with max-pooling and a pre-trained language model BERT (Devlin et al., 2019). Both are currently widely used language models in the NLP field.

5.2. *Baseline Method and Experimental Settings*

Backdoor attack methods. We choose the typical word-level backdoor attacks (addition and replacement) and sentence-level backdoor attacks. These methods are used in the paper (Chen et al., 2020a; Sun, 2021). **Backdoor defense baseline method.** We choose ONION (Qi et al., 2020) as the backdoor defense baseline method, which is the latest backdoor defense method we know that can be achieved without user control of the model training process.

Considering the reality that users directly apply third-party models that is to say, users may not control the model training process and cannot check the data set. Therefore, we set the target model to be the only accessible information. The premise of related research in this paper is based on this scenario. We randomly select 500 test samples with trigger from SST-2 and IMDB data sets, and control the length between 10 and 100 to evaluate the impact of the text backdoor defense method on the attack success rate. We select 1000 benign test samples from SST-2 and IMDB data sets to evaluate the recog-

Table 2 – Defensive performance for word-level backdoor attacks with random trigger positions. ASR represents attack success rate(%). ΔASR is the decrement of attack success rate(%).

Dataset	Attack Model	Defense Method	BiLSTM		BERT	
			ASR	ΔASR	ASR	ΔASR
SST-2	Word-level backdoor attack(addition)	No Defense	96.40	-	100.00	-
		DD	**0.60**	**95.80**	**0.00**	**100.00**
		DR	2.00	94.40	3.60	96.40
		ONION	14.40	82.00	16.40	83.60
	Word-level backdoor attack (replacement)	No Defense	97.00	-	100.00	-
		DD	**4.00**	**93.00**	5.60	94.4
		DR	4.60	92.40	**4.80**	**95.20**
		ONION	24.80	72.20	26.40	73.60
IMDB	Word-level backdoor attack(addition)	No Defense	97.00	-	98.80	-
		DD	1.00	96.00	**0.00**	**98.80**
		DR	**0.80**	**96.20**	2.20	96.30
		ONION	12.00	85.00	14.00	84.80
	Word-level backdoor attack (replacement)	No Defense	96.00	-	99.40	-
		DD	**1.80**	**94.2**	**2.00**	**97.40**
		DR	2.40	93.60	2.80	96.0
		ONION	17.60	78.40	20.00	79.40

nition accuracy of the infected model with the added defense mechanism under the clean data set.

5.3. *Evaluation Metrics*

In order to measure the quality of the generated samples, we set up various automatic evaluation indicators. The attack success rate (ASR) evaluates the performance of the backdoor attack method. The decrement of attack success rate (ΔASR) is the core indicator to measure the success of the defense method. Benign accuracy (BA) evaluates the accuracy of the recognition model under a clean data set. The addition of defense mechanisms cannot affect the normal function of the model. For the quality of the modified sample, we divide it into two parts: grammaticality and fluency. Grammaticality[1] is measured by the increase in the number of grammatical errors of the modified example compared to the original input, where we use Grammarly to obtain the number of grammatical errors of the sentence. we utilize the language model perplexity (PPL) to measure the fluency with GPT-2 (Radford et al., 2019).

5.4. *Defense Performance*

Defensive performance for word-level backdoor attacks with random trigger positions. We first evaluated the defense performance of our proposed BDDR and baseline method (ONION) against word-level backdoor attacks with random trigger positions. The BDDR defense model includes two defense methods: the backdoor defense method based on Detection + Deletion (DD) and the backdoor defense method based on Detection + Replacement (DR). The results are shown in Table 2. We observed that our two defense methods, namely DD and DR, had significantly higher ΔASR on the two data sets and two infected models than the baseline defense method. It proved the superiority of the proposed text backdoor defense method. It can be seen that the ΔASR of the two defense methods we proposed is higher than 90% on the two data sets and the two infected models, even on the infected BERT model. The above DD reduced the ASR of backdoor attacks to 0%. Specifically, the defense effect of DD was slightly better than that of DR. These experimental results proved the superiority of our defense model.

Table 3 – Defensive performance for word-level backdoor attacks with a fixed trigger position. ASR represents attack success rate(%). ΔASR is the decrement of attack success rate(%).

Dataset	Defense Method	BiLSTM		BERT	
		ASR	ΔASR	ASR	ΔASR
SST-2	No Defense	99.40	-	100.00	-
	DD	**0.20**	**99.20**	**0.00**	**100.00**
	DR	1.20	98.20	2.20	97.80
	ONION	35.20	64.20	35.40	64.60
IMDB	No Defense	99.40	-	99.20	-
	DD	**0.00**	**99.40**	**0.60**	**98.60**
	DR	0.20	99.20	1.40	97.80
	ONION	77.20	22.20	80.60	18.60

Defensive performance for word-level backdoor attacks with a fixed trigger position. We evaluated the defensive performance of our proposed BDDR and ONION against word-level backdoor attacks with fixed trigger positions. We add the trigger to the beginning of each test sample. The results are shown in Table 3. We observed that whether the trigger position was fixed does not affect the defense effect of our

[1] https://www.grammarly.com.

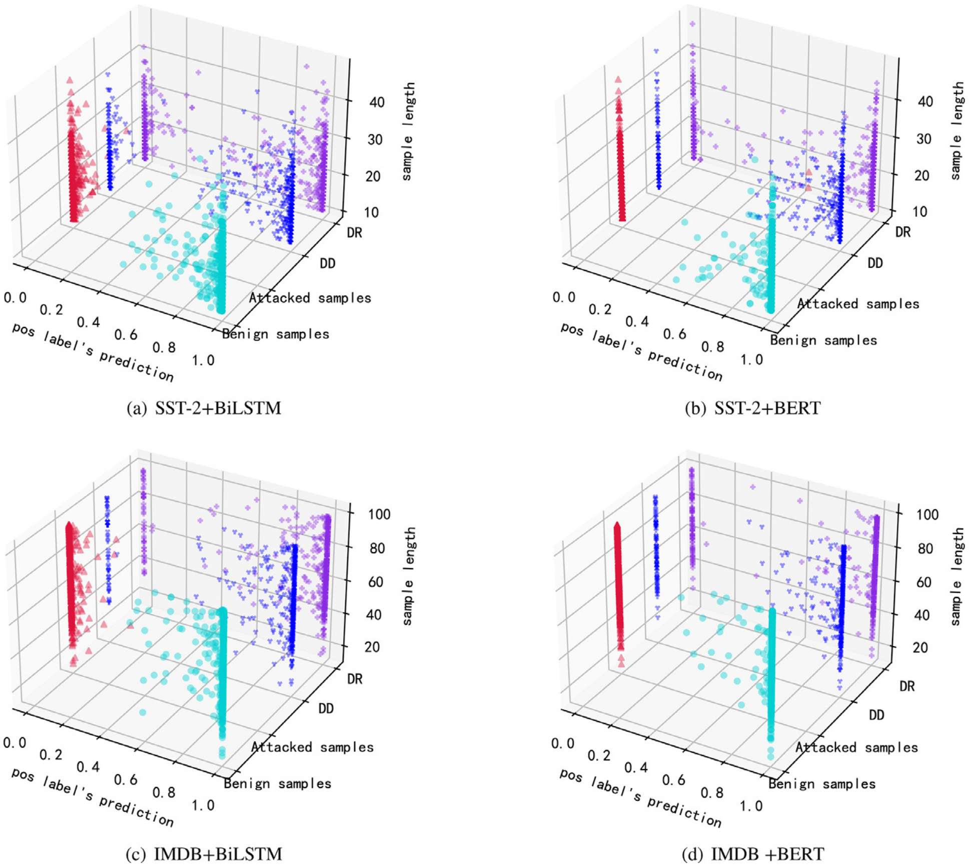
Table 4 – Defensive performance for sentence-level backdoor attacks. ASR represents attack success rate(%). ΔASR is the decrement of attack success rate(%).

Dataset	Defense Method	BiLSTM		BERT	
		ASR	ΔASR	ASR	ΔASR
SST-2	No Defense	100.00	-	99.60	-
	DD	**33.80**	**66.20**	**20.80**	**78.80**
	DR	40.00	60.00	27.00	72.60
IMDB	No Defense	100.00	-	100.00	-
	DD	**10.40**	**89.60**	**19.40**	**80.60**
	DR	12.00	88.00	21.60	78.40

proposed DD and DR. But ONION's defensive effect dropped significantly. Especially on the IMDB data set, for the infected BiLSTM model, ONION's ΔASR is only 22.2%. For the infected BERT model, the ΔASR of ONION was only 18.6%, which was much lower than that in the situation with random trigger position. The triggers we chose were 'comparatively' and 'approximately'. The result is that the sample fluency was better when the two triggers happen to be at the beginning of the sentence. ONION recognizes the backdoor based on the phenomenon that the sample fluency after adding the trigger will deteriorate. Therefore, when the trigger has little effect on the sentence fluency, it will cause ONION to fail. The BDDR is to identify the backdoor based on the phenomenon that adding a trigger will change the sample label. So the scope of application of the BDDR we mentioned is wider.

Defensive performance for sentence-level backdoor attacks. Since there is no mention in the paper that ONION can defend against sentence-level backdoor attacks, we evaluated the defense performance of the proposed BDDR against sentence-level backdoor attacks. The results are shown in Table 4. We observed that DD and DR have ΔASR greater than or equal to 60% on the two data sets and the two infected models. It showed that the proposed method has a good defensive effect against various types of text backdoor attacks. However, we found that DD and DR were less effective for sentence-level backdoor attacks than for word-level backdoor attacks. Because the sentence-level trigger was composed of multiple words, we only destroyed a part of them and may not prevent backdoor attacks.

For sentence-level backdoor attacks, we further analyze the classification probabilities of the model protected by BDDR on the two data sets. The analysis result is shown in Fig. 2. For SST-2 data and IMDB data sets, the negative range is [0, 0.5], and the positive range is [0.5, 1]. We observed that DD and DR corrected a large number of classification errors caused by sentence-level backdoor attacks. Even if DD and DR couldn't convert the negative comments generated by the backdoor attack into positive comments in some cases, it can still significantly reduced the probability of negative comments. The sen-



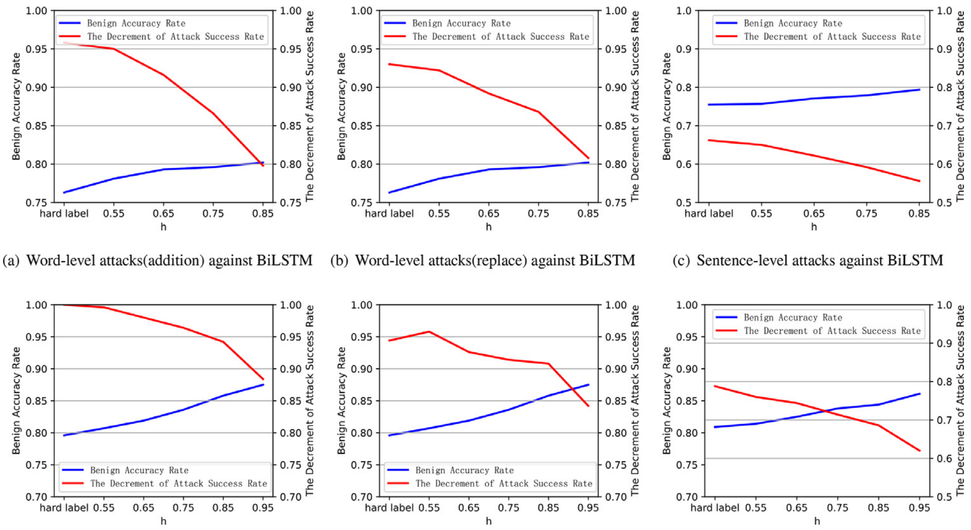
(a) SST-2+BiLSTM

(b) SST-2+BERT

(c) IMDB+BiLSTM

(d) IMDB +BERT

Fig. 2 – For sentence-level backdoor attacks, the classification probabilities of the model protected by BDDR on the two datasets.



(a) Word-level attacks(addition) against BiLSTM (b) Word-level attacks(replace) against BiLSTM (c) Sentence-level attacks against BiLSTM

(d) Word-level attacks(addition) against BERT (e) Word-level attacks(replace) against BERT (f) Sentence-level attacks against BERT

Fig. 3 – The influence of *h* on the defense effect of DD and the benign accuracy under SST-2.

timent score of the text was generally changing in a positive direction.

5.5. *The influence of defense methods on the benign accuracy of the infection model.*

First, we only used hard tags to change the detection criteria for suspicious words. Under this condition, we evaluated the impact of the proposed BDDR on the infection model's benign accuracy. The results are shown in Table 5. We observed that our proposed DR had minimal impact on the model's BA. Even when the data set was IMDB, the model was BERT, and the attack method was a word-level backdoor attack (addition), the BA of the model increased. Because we used the substitution words generated by BERT to modify the model's incorrectly classified samples, the modified samples were correctly classified by the model. However, the performance of DD on the SST-2 dataset was not satisfactory. Because the SST-2 dataset is short, and deleting suspicious words will directly destroy the sample's semantics.

In order to reduce the impact of DD on the benign accuracy of the infection model, we added the target label's prediction probability to the suspicious word detection standard (this method is also applicable to DR). Specifically, we set the hyperparameter *h*. When the change value of the word to the model output probability is greater than *h*, the word is considered to be a suspicious word. On the SST-2 data set, with different hyperparameter *h* values, we evaluated the impact of DD on the benign accuracy of the infected modeland the decrement of attack success rate. The result is shown in Fig. 3. We observed that as the value of *h* increased, the value of BA also increased. When the value of *h* was greater than 0.85, DD had minimal impact on the model's BA. Simultaneously, as the value of *h* increased, the defensive effect of DD on backdoor attacks also weakens. However, it can still effectively resist the performance of backdoor attacks. For word-level backdoor attacks, the ΔASR of DD on the two infected models is greater than or equal to 79.8%. For sentence-level backdoor attacks, the ΔASR of DD on the two infected models is greater than or equal to 55.6%. These results prove the effectiveness of our defense method.

5.6. *Example Quality*

We evaluated the quality of the attack samples reconstructed by the BDDR. The results are shown in Table 6. We observed that for word-level attacks (addition), the quality of the reconstructed sample (including grammar and fluency) of DD was slightly better than that of DR. This is because the implementation method of word-level attack (addition) is to add an extra

Table 5 – The influence of defense methods on the benign accuracy(%) of infection models.

Dataset	Model	Defense methods against word-level backdoor attack			Defense methods against sentence-level backdoor attack		
		No Defense	DD	DR	No Defense	DD	DR
SST-2	BiLSTM	81.60	76.30	**78.70**	81.20	75.50	**77.30**
	BERT	90.60	79.60	**87.40**	89.60	80.90	**85.20**
IMDB	BiLSTM	89.30	**89.40**	89.00	89.10	89.10	**87.60**
	BERT	89.90	88.30	**90.60**	91.30	90.00	**90.60**

Table 6 – Example quality. "I" and "PPL" indicate the average number of grammatical errors increased and language model perplexity respectively.

Dataset	Index	Defense methods against word-level backdoor attack(addition)		Defense methods against word-level backdoor attack(replacement)		Defense methods against sentence-level backdoor attack	
		DD	DR	DD	DR	DD	DR
SST-2	I	**0**	0.15	0.18	**0**	0.69	**0.38**
	PPL	**150.12**	155.67	209.41	**155.11**	220.03	**178.40**
IMDB	I	**0**	0.14	0.12	**0.02**	0.76	**0.36**
	PPL	**64.50**	66.02	71.47	**68.02**	72.19	**71.02**

word (trigger) to the sample to generate an attack sample. DD can accurately delete the trigger in the sample. For word-level attacks (replacement) and sentence-level attacks, the quality (including grammar and fluency) of the samples generated by DR was better than DD.

6. Conclusion and Future Work

In this study, we propose a novel defense method for textual backdoor attacks. The defense method includes suspicious word detection methods and original sample reconstruction methods based on deletion or replacement. We conducted extensive experiments to prove the advantages of our model in terms of defense effect, reconstruction sample quality, and impact on the accuracy of infected model identification. In the future, we will try to extend our defense model to other NLP tasks (semantic matching, reading comprehension, machine translation, etc.). We also consider adjusting our defense model and use it to defend against textual adversarial attacks.

Declaration of Competing Interest

The authors declare that they have no competing interests.

CRediT authorship contribution statement

Kun Shao: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing. **Junan Yang:** Conceptualization, Project administration, Supervision. **Yang Ai:** Investigation, Writing – review & editing. **Hui Liu:** Writing – review & editing. **Yu Zhang:** Data curation.

REFERENCES

Chen C, Dai J. Mitigating backdoor attacks in LSTM-based text classification systems by backdoor keyword identification arXiv:2007.12070.

Chen J, Su M, Shen S, Xiong H, Zheng H. Poba-ga: Perturbation optimized black-box adversarial attacks via genetic algorithm. Computers & Security 2019;85:89–106. doi:10.1016/j.cose.2019.04.014.

Chen X, Salem A, Backes M, Ma S, Zhang Y. BadNL: Backdoor attacks against NLP models arXiv:2006.01043.

Chen Y, Wu Y, Qin Y, Hu Y, Wang Z, Huang R, Cheng X, Chen P. Recognizing nested named entity based on the neural network boundary assembling model. IEEE Intelligent Systems 2020b;35(1):74–81. doi:10.1109/MIS.2019.2952334.

Conneau A, Kiela D, Schwenk H, Barrault L, Bordes A. Supervised learning of universal sentence representations from natural language inference data. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing; 2017. p. 670–80.

Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT; 2019. p. 4171–86.

Doan BG, Abbasnejad E, Ranasinghe DC. Februus: Input purification defense against trojan attacks on deep neural network systems. In: Annual Computer Security Applications Conference; 2020. p. 897–912.

Gao Y, Xu C, Wang D, Chen S, Ranasinghe DC, Nepal S. Strip: A defence against trojan attacks on deep neural networks. In: Proceedings of the 35th Annual Computer Security Applications Conference; 2019. p. 113–25.

Gu T, Liu K, Dolan-Gavitt B, Garg S. Badnets: Evaluating backdooring attacks on deep neural networks. IEEE Access 2019;7:47230–44. doi:10.1109/ACCESS.2019.2909068.

Haeb-Umbach R, Watanabe S, Nakatani T, Bacchiani M, Hoffmeister B, Seltzer ML, Zen H, Souden M. Speech processing for digital home assistants: Combining signal processing with deep-learning techniques. IEEE Signal Processing Magazine 2019;36(6):111–24. doi:10.1109/MSP.2019.2918706.

Huang S, Liu X, Yang X, Zhang Z. An improved shapeshifter method of generating adversarial examples for physical attacks on stop signs against faster r-CNNs. Computers & Security 2021;104:102120. doi:10.1016/j.cose.2020.102120.

Kolouri S, Saha A, Pirsiavash H, Hoffmann H. Universal litmus patterns: Revealing backdoor attacks in CNNs. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020. p. 298–307. doi:10.1109/CVPR42600.2020.00038.

Kurita K, Michel P, Neubig G. Weight poisoning attacks on pretrained models. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics; 2020. p. 2793–806.

Li Y, Wu B, Jiang Y, Li Z, Xia ST. Backdoor learning: A survey arXiv:2007.08745.

Li Y, Zhai T, Wu B, Jiang Y, Li Z, Xia S. Rethinking the trigger of backdoor attack arXiv:2004.04692.

Liu K, Dolan-Gavitt B, Garg S. Fine-pruning: Defending against backdooring attacks on deep neural networks. In:

International Symposium on Research in Attacks, Intrusions, and Defenses; 2018. p. 273–94.

Liu Y, Xie Y, Srivastava A. Neural trojans. In: 2017 IEEE International Conference on Computer Design (ICCD); 2017. p. 45–8. doi:10.1109/ICCD.2017.16.

Maas A, Daly RE, Pham PT, Huang D, Ng AY, Potts C. Learning word vectors for sentiment analysis. In: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies; 2011. p. 142–50.

Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language models are unsupervised multitask learners. OpenAI blog 2019:9.

Rakin AS, He Z, Fan D. Tbt: Targeted neural network attack with bit trojan. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020. p. 13195–204. doi:10.1109/CVPR42600.2020.01321.

Ren Y, Zhou Q, Wang Z, Wu T, Wu G, Choo KKR. Query-efficient label-only attacks against black-box machine learning models. Computers & Security 2020;90:101698. doi:10.1016/j.cose.2019.101698.

Socher R, Perelygin A, Wu J, Chuang J, Manning CD, Ng AY, 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; 2013. p. 1631–42.

Sun L. Natural backdoor attack on text data arXiv:2006.16176.

Tian C, Fei L, Zheng W, Xu Y, Zuo W, Lin CW. Deep learning on image denoising: An overview. Neural Networks 2020;131:251–75. doi:10.1016/j.neunet.2020.07.025.

Tran B, Li J, Mądry A. Spectral signatures in backdoor attacks. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems; 2018. p. 8011–21.

Qi F, Chen Y, Li M, Liu Z, Sun M. Onion: A simple and effective defense against textual backdoor attacks arXiv:2011.10369.

Qian Y, Ma D, Wang B, Pan J, Wang J, Gu Z, Chen J, Zhou W, Lei J. Spot evasion attacks: Adversarial examples for license plate recognition systems with convolutional neural networks. Computers & Security 2020;95:101826. doi:10.1016/j.cose.2020.101826.

Wang B, Yao Y, Shan S, Li H, Viswanath B, Zheng H, Zhao BY. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In: 2019 IEEE Symposium on Security and Privacy (SP); 2019. p. 707–23. doi:10.1109/SP.2019.00031.

Kun Shao is currently pursuing the Ph.D. degree in information and communication engineering with the National University of Defense Technology, Hefei, China. His research interests include adversarial machine learning and natural language processing.

Junan Yang received the Ph.D. degree from the University of Science and Technology of China, Heifei, China, and he is a professor in the National University of Defense Technology. His research interests include artificial intelligence, signal processing and natural language processing.

Yang AI is a lecturer in the National University of Defense Technology. His research interests include artificial intelligence and signal processing.

Hui Liu is a lecturer in the National University of Defense Technology. His research interests include artificial intelligence, signal processing and natural language processing.

Yu Zhang is currently pursuing the M.S. degree in information and communication engineering with the National University of Defense Technology, Hefei, China. His research interests include natural language processing and adversarial machine learning.