WordRevert: Adversarial Examples Defence Method for Chinese Text Classification

ENHUI XU1, XIAOLIN ZHANG1, YONGPING WANG1, SHUAI ZHANG1, LIXIN LU1,2 AND LI XU3

1School of Information Engineering, Inner Mongolia University of Science and Technology, Baotou 014010, China
2School of Information, Renmin University of China, Beijing 100872, China
3Baotou Medical College, Inner Mongolia University of Science and Technology, Baotou 014010, China

Corresponding author: Xiaolin Zhang (e-mail: zhangxl@imust.edu.cn).

This work was supported in part by the Natural Science Foundation of China under Grant 61562065, in part by the Inner Mongolia Natural Science Foundation Project under Grant 2019MS06001, and in part by the Inner Mongolia Natural Science Foundation Project under Grant 2019MS06036.

ABSTRACT Adversarial examples can evade the detection of text classification models based on Deep Neural Networks (DNNs), thus posing a potential security threat to the system. To address this problem, we propose an adversarial example defense method for Chinese text classification called WordRevert. The method first obtains the "positive text" containing the adversarial words by filtering the clauses that do not contribute to the current classification label. Then the detection network is combined with the position importance calculation function to achieve the detection of the adversarial words. Finally, the adversarial words are restored to the original words by calculating the candidate score and the detection score. The experiments show that the current popular Chinese text adversarial attack algorithms can be effectively defended by this method, and achieve a significant increase in the accuracy of the adversarial examples with a small reduction in the classification accuracy of clean samples while achieving better precision, recall, and F1 value of adversarial word detection and restoration.

INDEX TERMS adversarial examples, Chinese text classification, deep neural networks, defense, detection and restoration

I. INTRODUCTION

As Deep Neural Networks (DNNs) have achieved great success in solving many major problems in the fields of computer vision, natural language processing and speech recognition, systems based on DNNs have been widely used in production and life. However, recent studies have shown that DNNs are vulnerable to attacks from adversarial examples [1]. The research reveals the vulnerability of DNNs, which indicates that systems based on DNNs may have potential security threats. Attackers can disguise adversarial examples as normal examples through adversarial examples generation methods to evade the detection of classifiers, which seriously affects the security of sentiment analysis and spam classification systems.

The adversarial example is first discovered in image classification tasks. Szegedy et al. [1] find that adding disturbances that are imperceptible to humans in the original input images can make deep learning models go wrong with a high degree of confidence. According to the knowledge of the attacker, adversarial attacks can be divided into white-box attacks and black-box attacks. White-box attacks mean that the attacker has information such as the gradient, structure, and parameters of the target model [2] and black-box attacks mean that the attacker can query the output of the target model or has no information at all [3]–[6]. Generally speaking, black-box attacks are more in line with real-life scenarios. In the text field, due to the discrete attributes of the text and the perceptible attributes of disturbance, the generation of text adversarial examples is more difficult. With the deepening of research, some methods for generating text adversarial examples have also been proposed in recent years. What does not match it is the relative lack of defense methods. There may be two reasons for the lack of defense methods: one is that there is no unified theoretical model to describe the generation process of adversarial examples. In the face of unclear or completely unknown adversarial examples generation methods, the defense appears more passive. The other is multiple modification methods based on the original example
can cause errors in the model. Therefore, it is difficult to find adaptive methods for different adversarial attacks. At present, the research on adversarial examples generation and defense methods are mainly aimed at English, and the differences between languages make it impossible to transfer the originally few defense methods to Chinese text. Therefore, it is imperative to research adversarial examples defense for Chinese text classification.

In this paper, we propose a method called WordRevert to defend against adversarial examples in Chinese text classification tasks. The filtering operation on the original input is first implemented to obtain clauses that may contain adversarial examples. Then we use a bidirectional LSTM (BiLSTM) to implement a sequence labeling model as a detection network and combine the location importance algorithm to successfully locate the adversarial words, and finally use the scoring mechanism of adversarial words and candidate words to accurately restore adversarial words to original words. This method effectively defends against the current popular Chinese text adversarial attack algorithms, while ensuring the accuracy and recall of the restored text, and achieves a good defense effect. The main contributions of this paper are as follows:

1) We proposed an adversarial examples defence method for Chinese text classification, WordRevert, which can successfully defend against the existing adversarial attack methods for Chinese text classification. The detection and restoration of adversarial examples can be realized without changing the DNNs classification model.

2) We introduce a method for locating adversarial words based on sentence segmentation filtering, which can filter clauses that do not contain adversarial words. At the same time, the sequence labeling model implemented by BiLSTM is combined with the target classification model, and the position importance calculation function and the scoring mechanism of candidate words are designed to locate and restore the adversarial words to the original words more accurately.

3) Experiments are implemented on three datasets. The WordRevert algorithm is used to defend against three adversarial attack methods on two text classification models. The results show that the classification accuracy can be effectively improved to more than 86% on average, which is close to the accuracy of the original classification. Experiments also demonstrate that the impact of the WordRevert algorithm on clean samples is negligible, and the method has an excellent performance in precision, recall, and F1 score of adversarial word detection and restoration. Finally, ablation experiments are designed to verify the effectiveness of the adversarial word detection method.

The rest of this article is organized as follows. A detailed review of the relevant research work on textual adversarial attacks and defenses in Section II. The relevant definitions and defense models are formalized in Section III, and the WordRevert algorithm is introduced. In Section IV, experiments are conducted on three real datasets. Finally, the conclusion of the paper is given in Section V.

II. RELATED WORK

In recent years, adversarial examples have become a hot issue in the field of artificial intelligence and security, and related research has also increased dramatically. In response to this security problem, Fast Gradient Sign Method (FGSM) [7], Deepfool [8], C&W [9], Project Gradient Descent (PGD) [10] and many other adversarial examples generation methods have been proposed one after another. However, these methods are mostly aimed at images. Due to the discrete properties of text and metrics different from images, these methods cannot be directly applied to text [11].

In the text field, Papernot et al. [12] first migrated FGSM to text, combined with a specific dictionary, and successfully generated adversarial examples. Subsequently, a series of studies on text adversarial examples have been proposed [13]–[15]. Gao et al. [16] studied the adversarial examples generation method in the black-box scene and proposed the DeepWordBug algorithm. The algorithm uses the output results of the model to find keywords in the original text through the word importance calculation function, and then uses insertion, deletion, replacement, and exchange of characters to generate adversarial examples. Ren et al. [17] proposed a greedy algorithm for word-level attacks, which first determines the order of keywords that need to be replaced by Probability Weighted Word Saliency (PWWS), and then uses WordNet [18], which is a large lexical database for the English language, to find synonyms to generate adversarial examples. Similar to Ren et al., Zang et al. [19] proposed a word-level attack algorithm based on sememes, which can generate more diverse adversarial examples by finding words with the same sememes corresponding to each word in the original example by HowNet [20], and then using a particle swarm algorithm to optimize the combination of candidate words in discrete space.

The above adversarial example generation methods are designed based on English text. Some character-level modification strategies are not applicable to Chinese, and simple substitution using synonyms is not effective in Chinese text adversarial example generation. Wang et al. [21] first proposed WordHandling, a Chinese adversarial examples generation method. They used pre-trained substitution models and word importance calculation functions to determine the keywords to be replaced and used homophones to replace them, successfully reducing the classification accuracy of LSTM and CNN models to about 60%. However, only phonetic features of Chinese characters are used for keyword replacement in this method, other features are not fully utilized, and the modification strategy is relatively single. Based on the above research work, Tong et al. [22] proposed the CWAttacker algorithm, which locates important keywords and phrases through a targeted word deletion scoring mechanism and
uses attack strategies such as traditional character replacement, pinyin rewriting, and word order disturbance to attack Chinese text at the word-level. Similarly, Cheng et al. [23] added strategies such as word splitting and proposed the WordChange algorithm to realize the generation of Chinese text adversarial examples. The experiment proved that it can also cause interference to the actual deployed sentiment analysis system.

Unlike images, the research on text adversarial examples mainly focuses on the generation method, and the defense method is rarely involved. At this stage, the defense methods in the image field mainly include detection, model enhancement, and defense distillation. The defensive distillation method has proven to be ineffective in the text domain, and model enhancement is mainly achieved through adversarial training but it is not always effective [24]. The adversarial example is a kind of data with a specific purpose. It is a more intuitive idea to prevent the adversarial examples from attacking the model by detecting the input. The detection of adversarial examples has achieved some good research results in the image field [25], [26]. In the text, using certain methods to generate adversarial examples will produce spelling errors. This feature makes the detection method effective in the text.

In the detection method, Gao et al. [16] used the Python autocorrect 0.3.0 package to detect the input. Li et al. [13] use context-aware spell checking services to accomplish similar tasks. Experimental results show that this method of detecting adversarial examples by checking spelling errors is effective for character-level modifications, and partly effective for word-level attacks due to different modification strategies. However, the spell check method is not suitable for adversarial examples based on other languages including Chinese [21]. At present, the main work of adversarial examples in Chinese texts is focused on the generation of adversarial examples, and no in-depth research has been conducted on defense methods.

It can be seen from the above that the work related to text adversarial attacks has achieved good results, but there are still many problems to be solved in the field of text defense, especially the defense against Chinese texts. Therefore, based on previous research results, we learn from the current main methods of text adversarial examples defense, integrate text error checking into the research of Chinese text adversarial examples defense, and propose an adversarial example defense method for Chinese text classification and contribute to the field of adversarial machine learning.

## III. WORDREVERT

We focus on the research of adversarial examples defense methods for Chinese text classification. In the section, we first give the relevant definition of the problem, including the adversarial attack of text classification and the adversarial defense of text classification. Then we give a general description of the WordRevert algorithm. Finally, the WordRevert algorithm is introduced in detail. It consists of three parts: filtering operation, adversarial word detection, and adversarial word restoration.

### A. PROBLEM DEFINITION

Given a data set $X = \{x_1, x_2, \ldots, x_n\}$ with $N$ texts and a set of corresponding $N$ labels $Y = \{y_1, y_2, \ldots, y_n\}$. A pre-trained natural language classification model $F$, which needs to learn the mapping $f : X \rightarrow Y$ from the input text $x \in X$ to the label $y \in Y$. Finally, it can classify the original input text $x$ to the true label $y_{\text{true}}$ as much as possible, as shown in Equation (1):

$$\arg\max_{y_t \in Y} P(y_t \mid x) = y_{\text{true}}.$$  (1)

Under normal circumstances, an adversarial example $x^*$ is generated by adding a small disturbance $r$ to $x$. The adversarial example will cause model $F$ to give a wrong label, as shown in Equation (2):

$$\arg\max_{y_t \in Y} P(y_t \mid x^*) \neq y_{\text{true}}.$$  (2)

At the same time, the disturbance is required to be imperceptible to the human eye, which means it will not cause significant changes in semantics, ensuring that humans can still understand the meaning of the original text. So the adversarial example can be defined as shown in Equation (3) [17]:

$$x^* = x + r, \quad \|r\|_p < \epsilon, \quad \arg\max_{y_t \in Y} P(y_t \mid x^*) \neq \arg\max_{y_t \in Y} P(y_t \mid x).$$  (3)

In Equation (3), $\|r\|_p$, defined in Equation (4) uses $p$-norm to represent the constraint on perturbation $r$, and $L_0$, $L_2$ and $L_\infty$ are commonly used.

$$\|r\|_p = \left(\sum_{i=1}^{n} |w_i^* - w_i|^p\right)^{\frac{1}{p}}.$$  (4)

In Equation (4), the original input text is expressed as $x = w_1 w_2 \ldots w_l w_{l+1} \ldots w_n$, where $w_i \in \mathbb{D}$ is a word and $\mathbb{D}$ is a dictionary of words.

In order to meet the above constraints, adversarial words $w_i^*$ are usually obtained from third-party dictionaries and lexical databases. Such as WordNet [18], HowNet [20], etc. can be used in English texts to get adversarial words. In Chinese texts, homonyms [21], traditional Chinese [22], and other dictionaries are usually used to obtain adversarial words. Therefore, the defense against this type of attack can be summarized as detecting the adversarial word $w_i^*$ and reducing the adversarial word to the original word $w_i$. On the one hand, it is necessary to ensure that the restored text is classified as the true label, i.e., Equation (5):

$$\arg\max_{y_t \in Y} P(y_t \mid x_t) = y_{\text{true}},$$  (5)

where $x_t$ represents the restored text. On the other hand, it is necessary to ensure the similarity between the restored text and the original text, which can be measured by the F1 score.
The specific calculation method is as follows.

**Precision:**

\[ P = \frac{TP}{TP + FP} \quad (6) \]

**Recall:**

\[ R = \frac{TP}{TP + FN} \quad (7) \]

**F1 score:**

\[ F1 = \frac{2RP}{P + R} \quad (8) \]

Among them, the result of the restoration of the adversarial word is correctly regarded as TP, the missed detection of the adversarial word is regarded as FN, and the result of the restoration of the adversarial word is wrong or the normal text is wrongly modified is regarded as FP.

### B. GENERAL DESCRIPTION

**Start**

- **input**
  - **filtering operation**
  - **detection network**
  - **position importance calculation function**
  - **determine the adversarial word**
  - **get the candidate set**
  - **candidate score**
  - **detection score**
  - **determine the replacement word**
  - **replace the original text**

**End**

**FIGURE 1.** The flowchart of WordRevert algorithm

The flowchart of WordRevert algorithm is shown in figure 1. It mainly includes the following three steps:

1) **Filtering Operation:** The "positive text" is obtained by removing the clauses that do not contribute to the current classification label.

2) **Adversarial Word Detection:** It includes two parts: the detection network and the position importance calculation function. The former is used to detect the probability that each word is a typo, and the latter calculates the position of important features. The two results are simply processed and multiplied together to be the probability that the position is an adversarial word.

3) **Adversarial Word Restoration:** The candidate score and the detection score are included, and the replacement word is determined by the product of the two.

Among them, the detection network in the Adversarial Word Detection stage and the candidate score in the Adversarial Word Restoration stage are implemented by the same sequence labeling model BiLSTM, while the position importance calculation function and candidate score are implemented by the access target model. The above three steps will be described in detail below.

### C. FILTERING OPERATION

In the current research context, the adversarial attack methods for Chinese text classification are word-level attacks, which to find keywords that affect the classification through various methods, and then generate adversarial examples by modifying these words. We call these modified words adversarial words. The distribution of adversarial words in the text is different from the error distribution in general text. Under normal circumstances, the keywords that affect the prediction of the classification model are not evenly distributed throughout the original text, and the adversarial words are generated by masking the keywords, so the distribution of the adversarial words in the entire adversarial text is also not even. If we directly finds the adversarial words in a longer sentence, positioning bias may occur. Taking into account the influence of the text search space, in order to more accurately locate the adversarial words and improve the search efficiency, we propose a filtering operation, that is, to filter out the clauses that are not helpful to the current classification label, and leave the "positive text" that contribute to the current classification label. Finding adversarial words in the "positive text" will effectively improve the positioning accuracy and search efficiency.

According to the characteristics of the Chinese text, the entire input text \( x \) is divided into \( n \) clauses based on punctuation and spaces, as shown in Equation (9):

\[ x_{seg} = \{ s_1, s_2, \ldots, s_n \} \quad (seg = \text{, , ?!} \ldots) \]  

(9)

For the \( i \) clause in the sequence \( x_{seg} \), input the sequence after removing this clause into the target model \( F \) to obtain the confidence score of the current classification label \( y_i \). Then sequentially calculate the confidence difference between the
sequence after removing the clause and the original sequence, which is defined as Delete Score (DS) in Equation (10):

\[
DS(s_i) = P_F(y_i | x_{seg}) - P_F(y_i | x_{seg} - s_i) = F(s_1, \ldots, s_{i-1}, s_i, s_{i+1}, \ldots, s_n) - F(s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n),
\]

where \( P_F(y_i | x_{seg}) \) is the probability that the text sequence \( x_{seg} \) is classified as the current label \( y_i \) after being input to the target classification model \( F \). \( x_{seg} - s_i \) represents the text after removing the clause \( s_i \).

If \( DS(s_i) \leq 0 \), it means \( s_i \) does not contribute to the current classification label, and it will be removed from the original sequence \( x_{seg} \). After \( n + 1 \) visits to the target model \( F \), the filtered "positive text" \( x' \) is obtained. Then \( x' \) is sliced by word to get \( x' = \{w_1, w_2, \ldots, w_i, \ldots, w_n\} \), which is ready for the adversarial word detection later.

**D. ADVERSARIAL WORD DETECTION**

Adversarial word detection is the detection of adversarial words from the filtered "positive text" for later adversarial word restoration, and the detection effect determines the upper limit of the defense algorithm.

In the section, we introduce the detection and location algorithm of adversarial words, including two parts, one is the detection network, which is a sequence labeling model used to label the probability that each word in the input text is a typo. The other is the position importance calculation function, which is used to locate key positions in the input text, which is the high-frequency position of the adversarial word, to assist in locating the adversarial word.

The detection network is a sequence labeling model implemented by a BiLSTM network. The input is the word embedding sequence \( E = (e_1, e_2, \ldots, e_i, \ldots, e_n) \), where \( e_i \) represents the word embedding of the word \( w_i \). The output is a label sequence \( L = (l_1, l_2, \ldots, l_i, \ldots, l_n) \), where \( l_i \) represents the label of the word \( w_i \) and 0 represents correct, 1 means wrong. For each word \( w_i \) in the input sequence, there is a probability \( g_i \) that represents the possibility that its label is 1. The higher \( g_i \), the higher the probability of \( w_i \) being wrong. Using BiLSTM to construct the detection network, the calculation of \( g_i \) is shown in Equation (11):

\[
G(g_i) = P_D(l_i = 1 | x') = \sigma(Wh_i + b),
\]

where \( P_D(l_i = 1 | x') \) denotes the conditional probability of being a wrong word given the input text \( x' \), \( \sigma \) denotes the sigmoid function, \( W \) and \( b \) are the parameters of the detection network. \( h_i \) denotes the hidden state of the BiLSTM, as defined below:

\[
h_i = [h_i^\uparrow; h_i^\downarrow],
\]

\[
h_i^\uparrow = \text{LSTM}(\overline{h}_{i-1}; e_i),
\]

\[
h_i^\downarrow = \text{LSTM}(\overline{h}_{i+1}; e_i),
\]

where \([\overline{h}_i; \overline{h}_i] \) denotes splicing the LSTM hidden states in two directions, and LSTM is the LSTM function.

The learning process of the detection network is driven by optimizing an objective:

\[
\mathcal{L} = -\sum_{i=1}^{n} \log P_D(l_i | x'),
\]

where \( \mathcal{L} \) is the objective for the training of the detection network, which is defined as cross entropy loss. Using the equation (11), we can calculate the probability \( G = \{g_1, g_2, \ldots, g_i, \ldots, g_n\} \) that each word \( w_i \) in the "positive text" \( x' \) is a wrong word.

The detection network realizes the error detection of all words in the input sequence \( x' \). To locate the adversarial word more accurately, it is necessary to pay more attention to the position of the key features in the sequence, which is the high-frequency position of adversarial words. Therefore, we design a position importance calculation function to assist in locating adversarial words by calculating important positions.

We mark each word in \( x' = \{w_1, w_2, \ldots, w_i, \ldots, w_n\} \) as an unknown character in turn, and then input it into the target classifier. The difference between the original score of the target model and the current score is counted as the importance score of the position, as shown in Equation (16):

\[
S(w_i) = P_F(y_i | x') - P_F(y_i | x'_{w_i})
= F(w_1, w_{i-1}, w_i, w_{i+1}, \ldots, w_n) - F(w_1, w_{i-1}, \text{UNK}, w_{i+1}, \ldots, w_n),
\]

where \( P_F(y_i | x') \) is the probability value that the text \( x' \) is classified into the current label \( y_i \) after being input to the target classifier \( F \). \( x'_{w_i} \) represents the text after marking \( w_i \) as unknown words. UNK represents unknown words outside the vocabulary.

Using Equation (16), we can calculate the importance score \( S = \{s_1, s_2, \ldots, s_i, \ldots, s_n\} \) for each position in the "positive text" \( x' \). However, the locations of key features in clean examples also receive higher scores. To prevent misjudgment of clean examples, we set the output of the detection network \( G = \{g_1, g_2, \ldots, g_i, \ldots, g_n\} \) in the detection phase as in Equation (17):

\[
g_i = \begin{cases} g_i, & g_i \geq 0.5 \\ 0, & g_i < 0.5 \end{cases}
\]

Finally, the probability \( P = \{p_1, p_2, \ldots, p_i, \ldots, p_n\} \) of the adversarial word in the "positive text" \( x' \) is obtained by Equation (18):

\[
P = G \odot S
= \{g_1, g_2, \ldots, g_i, \ldots, g_n\} \odot \{s_1, s_2, \ldots, s_i, \ldots, s_n\}
= \{g_1s_1, g_2s_2, \ldots, g_is_i, \ldots, g_n s_n\}
\]

For \( x' = \{w_1, w_2, \ldots, w_i, \ldots, w_n\} \), we take \( w_i \) of \( p_i \geq \delta \) as the adversarial word and add it to the set \( X \) of adversarial words to be modified, where \( \delta \) is a hyperparameter.
E. ADVERSARIAL WORD RESTORATION

Adversarial word restoration means to restore the adversarial word to the original word, which is a key link in determining the performance of the adversarial defense algorithm. As mentioned above, adversarial words often occupy an important position in the text sequence and hide the important features of the original classification of the text sequence. The restoration of original words is actually the restoration of the original text features. In adversarial examples of text classification, the operation of replacing adversarial words with original words will make the confidence score of adversarial examples relative to the wrong label decrease rapidly. Using this feature, the original word can be found from the dictionary corresponding to the adversarial word. Aiming at the current popular Chinese adversarial example generation methods, including homophone substitution, traditional character substitution, we have designed the following defense methods.

We first obtain the set $T_i$ corresponding to each adversarial word $w_i$ in the set $X$ through the third-party dictionary, including homophones and traditional characters, and then get the candidate word $t_i$ from the set $T_i$ and replace the adversarial word $w_i$ in turn, input the replaced text into the target classifier to receive the confidence score relative to the current classification label. Finally, the confidence difference between the “positive text” $x'$ and the replaced text $x'_i$ is calculated as the candidate score $S(t_i)$, as shown in Equation (19):

$$S(t_i) = P_F(y_i | x') - P_F(y_i | x'_i) = F(w_1, \ldots, w_i-1, w_i, w_{i+1}, \ldots, w_n) - F(w_1, \ldots, w_i-1, t_i, w_{i+1}, \ldots, w_n),$$

where $t_i \in T_i$, $T_i$ is the candidate set corresponding to $w_i$.

Generally speaking, the original words belong to the correct vocabulary in the text sequence, whether in terms of lexical, grammatical, and semantic constraints. Therefore, to obtain the original word more accurately, we also send the replaced text to the detection network to get the probability $G(g_i)$ that the candidate word $t_i$ is the wrong word, as shown in Equation (20):

$$G(g_i) = P_D(t_i = 1 | x'_i) = \sigma(Wt_i + b),$$

and use $1 - G(g_i)$ to represent the probability that the candidate word $t_i$ is the correct word in the replaced text $x'_i$.

We select the candidate word $t_i$ with the highest product of the candidate score $S(t_i)$ and correct word probability $1 - G(g_i)$ as the final replacement word $t^*$, as shown in Equation (21):

$$t^* = \arg\max_{t_i \in T_i} (1 - G(g_i)) S(t_i).$$

For each adversarial word in the set $X$, the corresponding replacement word $t^*$ can be found, and these replacement words are used to replace the adversarial word in the original text $x$ to obtain the restored text $x_t$.

The final WordRevert algorithm is as shown in Algorithm 1.

---

Algorithm 1 WordRevert

**Input:** Text $x$; Target classification model $F$; Detection network BiLSTM

**Output:** Restored text $x_t$

1. Begin
2. $x_{\text{seg}} = \{s_1, s_2, \ldots, s_n\}$ ($\text{seg} = \circ, \text{?!} \ldots$)
3. for $i = 1, \ldots, n$ do
   4. $DS(s_i) = P_F(y_i | x_{\text{seg}}) - P_F(y_i | x_{\text{seg}} - s_i)$ is calculated by Equation (10)
   5. if $DS(s_i) \leq 0$ then
      6. $x' \leftarrow \text{Delete } s_i \text{ from } x_{\text{seg}}$
   7. end if
8. end for
9. $x' = \{w_1, w_2, \ldots, w_i, \ldots, w_n\}$
10. $G = \{g_1, g_2, \ldots, g_i, \ldots, g_n\}$ is calculated by Equation (11)
11. for $i = 1, \ldots, n$ do
12. $S(w_i) = P_F(y_i | x') - P_F(y_i | x'_{\bar{w}_i})$ is calculated by Equation (16)
13. end for
14. $S = \{s_1, s_2, \ldots, s_i, \ldots, s_n\}$
15. $P = G \ast S$, $g_i$ has been calculated by Equation (17)
16. $P = \{p_1, p_2, \ldots, p_i, \ldots, p_n\}$
17. if $p_i \geq \delta$ then
18. $X \leftarrow \text{add } w_i \text{ into } X$
19. end if
20. $x_t \leftarrow x$
21. for $w_i$ in $X$ do
22. $T_i \leftarrow \text{find a candidate set } T_i \text{ corresponding to } w_i$
23. for $t_i$ in $T_i$ do
24. $S(t_i) = P_F(y_i | x') - P_F(y_i | x'_{t_i})$ is calculated by Equation (19)
25. $G(g_i)$ is calculated by Equation (20)
26. end for
27. $t^* = \arg\max_{t_i \in T_i} (1 - G(g_i)) S(t_i)$
28. $x_t \leftarrow \text{replace } w_i \text{ with } t^* \text{ in text } x_t$
29. end for
30. return $x_t$
31. End

---

IV. EMPIRICAL EVALUATION

For empirical evaluation, we use WordRevert on three real data sets involving two deep neural network classification models to defend against existing Chinese text classification adversarial attack methods and analyze the defense effect.

A. DATASETS

We used the SIGHAN dataset, which is the benchmark for Chinese spelling error correction. SIGHAN is a small data set that contains 1100 texts and 461 character-level errors. In order to detect the error types of traditional characters in the adversarial example, we also mark the traditional characters in the data. Since the texts are collected in the essay section of Test of Chinese as Foreign Language and the subject scope is narrow, we added 5000 Ctrip hotel review data, 5000 JD
shopping review data and 10000 spam classification data, which was disturbed by the adversarial example generation method. Finally, all the above data is processed into sentence pairs for training the detection network.

We also prepared more data for the target models, as shown in Table 1, including 30,000 Ctrip hotel review data, 50,000 JD shopping review data, and 90,000 spam classification data, and use these data to train the target models. In addition, 3000 Ctrip hotel review data, 5000 JD shopping review data, and 10000 spam classification data are also prepared to test the classification accuracy of the target models. These test sets are also used to generate adversarial examples using currently popular adversarial attack algorithms for Chinese text classification to attack the target model. The WordRevert algorithm proposed in the paper is used to defend against attacks from adversarial examples. The defense effect is verified by analyzing the changes in the classification accuracy of the target model. These data are all real data from the Internet. The average number of words is 144 words, 32 words, and 51 words respectively, which are used to evaluate the impact of different lengths of adversarial examples on the defense effect.

B. ADVERSARIAL ATTACK ALGORITHMS AND TARGET MODELS

We use the currently popular adversarial examples attack method for Chinese text classification to verify the effectiveness of the WordRevert algorithm, including WordHanding [21], CWordAttacker [22] and WordChange [23]. In experiments, these methods are used to generate adversarial example to attack target models. Among them, the homophone replacement strategy is selected in WordHanding, the traditional character replacement strategy is selected in CWordAttacker, and the homophone replacement and Tongue-flatted or Tongue-rolled Pronunciation Replacement(TTPR) are selected in WordChange. In the experiment, the threshold of these adversarial example generation methods is set to 30, which means that the number of modified words in each text does not exceed 30.

For the target models, we considered several classic and state-of-the-art models for text classification, which include Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN).

**Word-based CNN** (word-CNN) [27] consists of an embedding layer that performs 50-dimensional word embeddings on 400-dimensional input vectors, an 1D-convolutional layer consisting of 250 filters of kernel size 3, an 1D-max-pooling layer, and two fully connected layers.

**Bidirectional LSTM** (BiLSTM) consists of a 128-dimensional embedding layer, a bidirectional LSTM layer whose forward and reverse are respectively composed of 64 LSTM units, and a fully connected layer.

The learning rate of these two models is set to 0.003 during the training process, and the default training epoch is 200.

C. EXPERIMENT SETTING

Given that there is no defense method specifically for Chinese text classification, we defend against the currently popular attack methods and verify the effectiveness of the WordRevert algorithm by evaluating the defense effect. As an evaluation measure, we first evaluate the defense effect of the WordRevert algorithm by analyzing the changes in the classification accuracy of the target model before and after the defense. The effect of the method on the classification accuracy of clean samples is then further analyzed. Finally, we use word-level precision, recall, and F1 score to evaluate the performance of the algorithm in the two stages of detection and restoration.

In the detection network, the dimension of the embedding layer is 128, the size of the hidden unit of BiLSTM is 256, and the batch size used is 64.

The learning rate in the training phase is set to 0.001, and the early stopping method is used for training. When the loss stops decreasing for 10 epochs, the training will be stopped. In the experiment, the hyperparameter $\delta$ in the WordRevert algorithm is set to 0.4.

D. DEFENDING RESULTS

We use the above three attack methods to generate adversarial examples for the test set to evaluate the effectiveness of the WordRevert algorithm. The more effective the defense method is, the more the model’s classification accuracy rises, and the closer it is to the classification accuracy of the original examples. Table 2 shows the classification accuracy of the word-CNN model and BiLSTM model on the original examples(Ori.), the adversarial examples generated by these attack methods(Att.), and the examples after defense using the WordRevert algorithm(Def.).

The results show that our method maximizes the classification accuracy. The classification accuracy of Ctrip hotel review dataset, JD shopping review dataset, and spam classification dataset increase to 88.65%, 86.81% and 96.45% on average for the word-CNN model respectively, and increase to 86.79%, 87.84% and 97.20% on average for the BiLSTM model respectively. Compared with the other two data sets, the defense algorithm performs better on the spam classification dataset.

In order to further explore the impact of the WordRevert algorithm on the accuracy of clean examples, we separately defended against 1,000 clean examples for each dataset, and input the defensive results into the target model to observe the changes of accuracy.

Table 3 shows the classification accuracy of the word-CNN model and the BiLSTM model on the original examples and the original examples after the defense. The result shows that the WordRevert algorithm can reduce the accuracy of clean samples by at most 1%, and this effect is negligible.

Table 4 presents the performance of the WordRevert algorithm in the detection and restoration of adversarial words, evaluated by word-level precision(Prec.), recall(Rec.), and F1 score(F1.). Obviously, restoration is more difficult than
words will be found, which determines the upper limit of the defense effectiveness. Meanwhile, the precision of the restoration phase can reach up to 91%, which means that the probability of correcting a normal word by mistake or not correctly restoring to the original word is only 10%, and the high precision of the restoration phase maximizes the effect of the detection stage.

### E. ABLATION STUDY

We use Ctrip hotel review data to conduct ablation research on the WordRevert algorithm to evaluate the contribution of the detection network and the location importance calculation function to the defense effect. This is done as follows: 1000 samples are randomly selected from Ctrip hotel review data and input to the target model, and the classification accuracy is recorded as "Original". The target model is attacked using different adversarial attack methods and the classification accuracy is recorded as "Attacked". The target model is defended using the WordRevert algorithm, and the model accuracy is recorded as "WordRevert". Then we use the WordRevert algorithm that removes the detection network in the adversarial word detection stage to defend the target model, and the record classification accuracy is "WordRevert-N". Similarly, the defense accuracy of the WordRevert algorithm that removes the position importance calculation function in the adversarial word detection stage is recorded as "WordRevert-F". And it can represent the general Chinese typos detection model. In the WordRevert-N method, the importance score of the position $S = \{s_1, s_2, \ldots, s_i, \ldots, s_n\}$ becomes the only criterion to determine the adversarial word, and we use the size of $s_i$ to determine whether the word is an adversarial word. Similarly, in the WordRevert-F method, the output sequence of the detection network $G = \{g_1, g_2, \ldots, g_i, \ldots, g_n\}$ becomes the only criterion for judging an adversarial word, and $g_i$ is used to determine whether the word is an adversarial word. The settings of the adversarial word restoration stage remain unchanged in the above two methods.

Figure 2 shows the performance of different defense methods on word-CNN and BiLSTM respectively. The results show that both defense methods have a certain defense effect, and the accuracy of WordRevert-F is slightly higher than that of WordRevert-D, indicating that the defense effect of the detection network is better than the location importance calculation function. The reason is that the position importance calculation function will also locate the key position in the clean sample, which will misjudge the normal word as the adversarial word, which may eventually cause the

| TABLE 1. Statistics on the datasets |
|-----------------------------------|
| Dataset | Classes | Train samples | Test samples | Average words | Task       |
|---------|---------|--------------|--------------|--------------|------------|
| Ctrip   | 2       | 30000        | 3000         | 144          | Sentiment analysis |
| JD      | 2       | 50000        | 5000         | 32           | Sentiment analysis |
| Spam    | 2       | 90000        | 10000        | 51           | Spam classification |

| TABLE 2. The defense effect of the WordRevert algorithm |
|--------------------------------------------------------|
| (a) word-CNN                                           |
| Dataset | Ori. (%) | Methods         | Att. (%) | Def. (%) |
|---------|----------|-----------------|----------|----------|
| Ctrip   | 92.03    | WordHanding     | 88.43    | 91.73    |
|         |          | CWordAttacker   | 62.37    | 88.20    |
|         |          | WordChange      | 60.50    | 88.03    |
| JD      | 91.46    | WordHanding     | 70.44    | 87.09    |
|         |          | CWordAttacker   | 56.66    | 86.62    |
|         |          | WordChange      | 54.36    | 84.74    |
| Spam    | 99.45    | WordHanding     | 58.23    | 96.64    |
|         |          | CWordAttacker   | 84.62    | 97.52    |
|         |          | WordChange      | 55.58    | 95.20    |

| (b) BiLSTM                                           |
| Dataset | Ori. (%) | Methods         | Att. (%) | Def. (%) |
|---------|----------|-----------------|----------|----------|
| Ctrip   | 90.13    | WordHanding     | 59.60    | 85.43    |
|         |          | CWordAttacker   | 59.43    | 86.97    |
|         |          | WordChange      | 58.43    | 87.97    |
| JD      | 92.24    | WordHanding     | 63.92    | 88.66    |
|         |          | CWordAttacker   | 62.32    | 89.36    |
|         |          | WordChange      | 52.86    | 85.50    |
| Spam    | 99.45    | WordHanding     | 57.78    | 96.30    |
|         |          | CWordAttacker   | 89.63    | 98.69    |
|         |          | WordChange      | 52.37    | 96.61    |

| TABLE 3. The impact of defense algorithms on the accuracy of clean examples |
|-------------------------------|
| Dataset | Models | Original | WordRevert |
|---------|--------|----------|------------|
| Ctrip   | word-CNN       | 90.50    | 89.70      |
| JD      | BiLSTM         | 89.80    | 87.90      |
| Spam    | word-CNN       | 90.60    | 90.20      |
|         | BiLSTM         | 98.70    | 98.80      |
|         | BiLSTM         | 98.80    | 97.90      |
We propose an effective method called WordRevert for defending against adversarial examples on Chinese text classification tasks. The WordRevert algorithm introduces a new adversarial word detection method determined by the detection network and the positional importance calculation function and a new adversarial word restoration method determined by both the candidate score and the detection score. Before adversarial word detection is performed, the filtering operation is used to filter out clauses that do not contribute to the current classification label, which will contribute to the efficiency and accuracy of the search. Experiments show that the WordRevert algorithm can greatly improve the classification accuracy of adversarial text with a small reduction in the classification accuracy of the clean text while maintaining a high level of precision, recall, and F1 score in the detection and restoration of adversarial words.

Our work organically integrates detection networks and location importance algorithms and candidate scoring mechanisms for text features to provide an effective defense against existing adversarial attack algorithms for Chinese text classification. In the future, we hope to evaluate the defensive effectiveness and efficiency of our approach on more classification tasks and models, and further optimize the detection model for the detection and restoration of non-equal-length adversarial examples.

**REFERENCES**

[1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014.

[2] Xiaojing Zheng, Jieheng Zeng, Yi Zhou, Cho-Jui Hsieh, Minhao Cheng, and Xuanjing Huang. Evaluating and enhancing the robustness of neural network-based dependency parsing models with adversarial examples. In Proceedings of the 58th Annual Meeting of the Association for Compu-
Nuo Cheng, Guoqin Chang, Haichang Gao, Ge Pei, and Yang Zhang.

Zhendong Dong, Qiang Dong, and Changling Hao. HowNet and its

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang,

George A. Miller. Wordnet: A lexical database for english. Commun.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating

Ji Gao, Jack Lanchantin, M. Soffa, and Yanjun Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. 2018 IEEE Security and Privacy Workshops (SPW), May 2018.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. Bert-attack: Adversarial attack against bert using bert. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020.

Siddhant Garg and Goutham Ramakrishnan. Bae: Bert-based adversarial examples for text classification. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020.

Di Jin, Zhijing Jin, Joey Zhou, and Peter Szolovits. Is bert really robust? A strong baseline for natural language attack on text classification and entailment. Proceedings of the AAAI Conference on Artificial Intelligence, 34:8018–8025, 04 2020.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. arXiv 1412.6572, 12 2014.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2016.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. 2017 IEEE Symposium on Security and Privacy (SP), May 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2017.

Bin Lian, Hongcheng Li, Miaojiang Su, Pan Bian, Xirong Li, and Wenchang Shi. Deep text classification can be fooled. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Jul 2018.

Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. Crafting adversarial input sequences for recurrent neural networks. In MILCOM 2016 - 2016 IEEE Military Communications Conference, page 49–54. IEEE Press, 2016.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. Textbugger: Generating adversarial text against real-world applications. Proceedings 2019 Network and Distributed System Security Symposium, 2019.

Javid Ebrahimi, Aniyi Rao, Daniel Lowd, and Dejing Dou. Hotflip: White-box adversarial examples for text classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 31–36, 01 2018.

Melika Behjati, Seyed-Mohsen Moosavi-Dezfooli, M. Baghshah, and P. Frossard. Universal adversarial attacks on text classifiers. ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 7345–7349, 2019.

Ji Gao, Jack Lanchantin, M. Soffa, and Yanjun Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. 2018 IEEE Security and Privacy Workshops (SPW), pages 50–56, 2018.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1085–1097, Florence, Italy, July 2019. Association for Computational Linguistics.

George A. Miller. Wordnet: A lexical database for english. Commun. ACM, 38(11):39–41, November 1995.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qin Liu, and Maosong Sun. Word-level textual adversarial attacking as combinatorial optimization. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020.

Zhendong Dong, Qiang Dong, and Changling Hao. HowNet and its computation of meaning. In Coling 2010: Demonstrations, pages 53–56, Beijing, China, August 2010. Coling 2010 Organizing Committee.

WANG Wen-Qi, WANG Run, WANG Li-Na, and Tang Ben-Xiao. Adversarial examples generation approach for tendency classification on chinese texts. Journal of Software, 30(8):2415, 2019.

TONG Xin, WANG Luona, WANG Runzheng, and WANG Jingya. A generation method of word-level adversarial samples for chinese text classification. Netinfo Security, 20(9):12, 2020.

Nuo Cheng, Guoqin Chang, Haichang Gao, Ge Pei, and Yang Zhang. Wordchange: Adversarial examples generation approach for chinese text classification. IEEE Access, PP:1–1, 04 2020.

Wei Emma Zhang, Quan Z. Sheng, Ahmed Alhazmi, and Chenliang Li. Adversarial attacks on deep-learning models in natural language processing: A survey. ACM Trans. Intell. Syst. Technol., 11(3), April 2020.

Kevin Roth, Yannic Kilcher, and Thomas Hofmann. The odds are odd: A statistical test for detecting adversarial examples. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, volume 97 of Proceedings of Machine Learning Research, pages 5498–5507. PMLR, 2019.

A. Khershouch, S.A. Fezza, Wassim Hamidouche, and O. Deforges. Detection of Adversarial Examples in Deep Neural Networks with Natural Scene Statistics. In 2020 International Joint Conference on Neural Networks, ICNN 2020, Proceedings of the International Joint Conference on Neural Networks, page 9206959, Glasgow, United Kingdom, July 2020. Institute of Electrical and Electronics Engineers Inc.