Towards a Robust Deep Neural Network in Texts: A Survey

Wenqi Wang[†§], Lina Wang[†§*], *Member, IEEE,* Run Wang[‡], Zhibo Wang[§*], *Member, IEEE,* Aoshuang Ye[†§]

---◆---

Abstract—Deep neural networks (DNNs) have achieved remarkable success in various tasks (*e.g.*, image classification, speech recognition, and natural language processing). However, researches have shown that DNN models are vulnerable to adversarial examples, which cause incorrect predictions by adding imperceptible perturbations into normal inputs. Studies on adversarial examples in image domain have been well investigated, but in texts the research is not enough, let alone a comprehensive survey in this field. In this paper, we aim at presenting a comprehensive understanding of adversarial attacks and corresponding mitigation strategies in texts. Specifically, we first give a taxonomy of adversarial attacks and defenses in texts from the perspective of different natural language processing (NLP) tasks, and then introduce how to build a robust DNN model via testing and verification. Finally, we discuss the existing challenges of adversarial attacks and defenses in texts and present the future research directions in this emerging field.

Index Terms—Adversarial attack and defense, Adversarial example, Deep neural networks, Testing and verification.

1 INTRODUCTION

Nowadays, DNNs have shown their power in addressing masses of significant problems in various areas, such as computer vision [1], [2], audio [3], [4], and natural language processing [5], [6]. Due to their great success, DNN-based systems are widely deployed in the physical world, including many security-critical areas [7]–[11]. However, Szegedy et al. [12] first found an interesting fact that crafted inputs by adding imperceptible perturbations could easily fool DNNs. These modified inputs are called adversarial examples, which can bring potential security threats to DNN-based systems. Therefore, they have become a hot issue in artificial intelligence and security in recent years, and related research has increased dramatically. Recently, adversarial examples are found in many fields, varying from image to other domains. Studies show that sign recognition system [13], object recognition system [14], audio recognition or control system [15]–[17], malware detection system [18], [19], and sentiment analysis system [20] are all vulnerable

to adversarial examples. Figure 1 is an adversarial attack on sentiment analysis API of ParallelDots[1] and the adversarial example we use comes from the work of [21]. In Figure 1, we can see that the prediction results change from negative to positive when the word 'I' in the original sample is replaced by 'Excellent'.
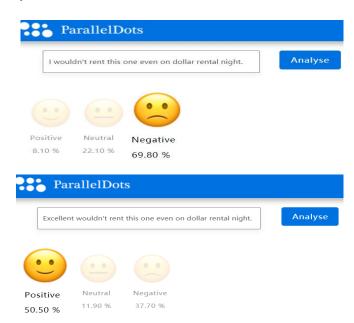

Fig. 1: Attack on sentiment analysis API of ParallelDots: the upper and lower figures correspond to the original input and the adversarial example respectively.

In NLP, DNNs are widely employed in many fundamental tasks like text classification, machine translation, and question answering, but these DNN-based systems also suffer significant performance degradation in facing adversarial examples. Papernot et al. [21] first point that attackers can generate adversarial examples by adding noises into texts, which can make classifiers misclassify. Then, an arm begins between adversarial example attack and defense in texts, leading to many studies in this field. These researches mainly concentrate on generating effective adversarial examples and introduce corresponding defense strategies. Some of them [22], [23] propose the black-box

---

[†] *W. Wang, L. Wang, and A. Ye are with Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education*
[§] *W. Wang, L. Wang, Z. Wang, and A. Ye are with School of Cyber Science and Engineering, Wuhan University, China. E-mail: {wangwenqi_001, lnwang, zbwang, yasfrost}@whu.edu.cn*
[‡] *R. Wang is with Nanyang Technological University, Singapore. E-mail: runwang@yeah.net*
[*] *Lina Wang and Zhibo Wang are the corresponding authors.*

1. https://www.paralleldots.com

TABLE 1: Adversarial attacks in NLP

NLP tasks	researches of adversarial attacks
classification	[20]–[26], [31]–[33]
reading comprehension	[27]
natural language inference	[28], [34]
machine translation	[29], [30]
answering question	[35]
argument reasoning comprehension	[36]
link prediction	[37]

Fig. 2: Publications of adversarial examples in DNNs

attacks and the main object is the text classification system [20], [22]–[26]. In addition, reading comprehension [27], natural language inference (NLI) [28] and machine translation systems [29], [30] are also vulnerable to adversarial examples. Table 1 summarizes existing works on adversarial attacks in different NLP tasks.

Adversarial examples in texts also bring some potential security concerns to users, as many text-related systems based on DNNs are deployed for providing services (like Figure 1). In the physical world, users are prone to acquire suggestions by reading reviews of products or services when shopping, eating or watching films. Hence, some apps provide recommendation services to users according to the historical reviews with sentiment analysis techniques [38]. The corresponding items with recommendation scores will be given at the same time. The higher the score is, the more likely it is to be accepted by users. Unfortunately, attackers could generate adversarial examples based on users' real comments to smear competitors or give malicious recommendations for shoddy goods. The recommendation scores of specified products could be manually controlled by intentionally crafted reviews. Besides, systems for detecting malicious information [39]–[41] are also under the threat of adversarial examples. Therefore, researchers have paid much attention to the security problem caused by adversarial examples [42], [43]. Numerous works study the adversarial attacks and defenses, aiming at exploring what adversarial examples are [12], [44]–[46], why they exist, how they infect the behavior of DNN models, and how to solve this security problem. We count the number of adversarial example papers in recent years which is shown in Figure 2. The last statistical time is to the end of November in 2019. In Figure 2, the orange one represents studies on adversarial texts, and the blue one stands for the papers collected by Carlini [47] about adversarial examples in image, audio, text, etc. We can see that the number of related publications is increasing sharply in the past three years representing great concern by researchers. But researches in texts are rare as a frontier, and more attention needs to be paid to this aspect.

Due to the security and privacy issues caused by adversarial examples, it is of great significance to review their rapid developments. There have been several surveys in image domain [45], [48]–[52], but few in texts [53]–[55]. The works in [53], [54] are partly related to adversarial texts. The remaining one [55] compares attack methods in image domain and describes how adversarial attacks are implemented in texts. These surveys mainly focus on adversarial attacks and defenses, but there is not a systematic review for build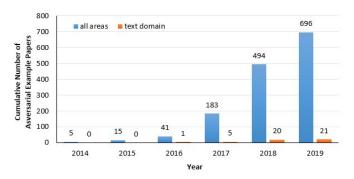ing a robust DNN model. In this paper, we explain the existence of adversarial examples and analyze how to attack and defend in various NLP tasks, aiming at systematically understanding adversarial examples towards building a robust DNN model in texts. The major contributions of this paper are summarized as follows.

- The paper presents a comprehensive survey of adversarial examples in texts, which are published in different areas like security, artificial intelligence, natural language processing, and software engineering.
- We present the adversarial attacks and defenses in texts by considering specific NLP tasks and introduce various metrics in measuring adversarial texts. We also discuss how to improve the robustness of DNNs via testing and verification to build a robust DNN model in texts.
- We observe that there is a pronounced imbalance in the distribution of research efforts: among the 47 papers in texts we collected, three of them tackle testing and verification, only one paper specializes in defense, and the remaining is about attacks. Additionally, there is only one paper study the factor of speed in adversarial examples.
- We answer an essential question that which attack or defense method is better via comparing some representative works with the same datasets. We present some general observations of adversarial examples and discuss the existing challenges. Finally, we also introduce some feasible directions with the propose of facilitating and stimulating further research.

The rest of this paper is organized as follows: we first give some related knowledge about adversarial examples in section 2. In section 3, we review the adversarial attacks for text classification. Attacks on other NLP tasks are in section 4. The researches about defense are introduced in sections 5 and 6. One of them is on existing defense methods in texts. The other is about how to improve the robustness of DNNs from another point. The discussion and conclusion are in sections 7 and 8.

2 PRELIMINARIES

In this section, we give a brief introduction on DNNs, followed by formula descriptions, interpretation of adversarial examples, general classification, evaluation, and corresponding datasets in texts.

2.1 Deep Neural Networks

The deep neural network is a network topology, which can learn high-level features with more complexity and abstraction than general neural networks [56]. A DNN model generally consists of an input layer, several hidden layers, and an output layer. Each of them is made up of multiple cells. Additionally, a softmax layer is usually added to DNN models for classification by mapping outputs to a probability distribution. Figure 3 shows some DNN models, including recurrent neural network (RNN), long-short term memory (LSTM) [57], and gated recurrent unit (GRU) [58], which can be used in text and image domains.
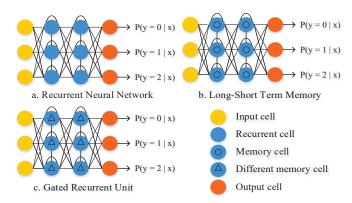

Fig. 3: Some frequently used DNN models in texts

But different from the use of these models in image domain, an extra embedding layer is added to them for processing data in texts, because the raw texts need to be converted to vectors before learning. This process is called word embedding, aiming at representing the ideal semantic space of words in a real-valued continuous vector space [59]. The commonly used embedding methods are one-hot, n-gram [60], and word2vec [61]. One-hot encoding is a technique whose dimension is equal to the size of the whole vocabulary, thus often resulting in large, very sparse vectors [62]. An n-gram model is a kind of probabilistic language model to predict next word in the form of previous words, which lacks long range dependency. Word2vec offers the best representation within a low dimensional semantic space learning from raw texts [63]. It includes two architectures: continuous bag-of-words (CBOW) and skip-gram. Word2vec is the most frequently used method for word embedding at present.

2.2 Formula Descriptions

In order to have a more intuitive understanding of the definitions, we give some formula descriptions about DNN, adversarial examples, and robustness of models.

DNN. A DNN can be presented as the function $F : X \to Y$, which maps from an input set X to Y. Y is the label set of k classes like $\{1, 2, \ldots, k\}$. For a sample $x \in X$, it is correctly classified by F to the truth label y, i.e., $F(x) = y$.

Adversarial Examples. An attacker aims at adding small perturbations ε in x to generate adversarial example x', so that $F(x') = y'(y \neq y')$, where $\| \varepsilon \| < \delta$. δ is a threshold to limit the size of perturbations. Meanwhile, a good x' should not only fool F, but also be imperceptible to humans, robust

to transformations as well as resilient to existing defenses depending on the adversarial goals [64]. Hence, constraint conditions (e.g., semantic similarity) are appended to make x' be indistinguishable from x.

Robustness. A robust DNN should not change its output as a result of small changes in its input [65]. Hence, the prediction of the adversarial example x' should be y rather than y' in a robust model, i.e., $F(x') = y$. Defense methods for enhancing the robustness of models are to increase the tolerance of ε, making DNNs perform properly.

2.3 Interpretation of Adversarial Examples

Since the problem of adversarial examples is so serious, a question arises: why adversarial examples exist in DNNs? Researchers have been exploring the existence of adversarial examples since they were discovered. Nowadays, there are two main points about their existence. One is due to the linear structure of DNNs, which can lead to false predictions with a high probability. Goodfellow et al. [44] claimed that the primary cause of adversarial examples was the linear nature of DNNs. Although the non-linear activation functions are the main parts of DNNs, they are linear in a sense. Some of the main activation functions are shown in Figure 4, including tanh, sigmoid, relu [66], and ELU[2] [67]. In Figure 4, we can see that these functions are very piecewise linear. Besides, there also exist other linear structures, such as the connection of each layer and convolution calculation. A defect of the linear structure is that classification is still possible as it moves away from the decision boundaries, even though there is no training data. However, false predictions are usually made in these places, so that the presence of adversarial examples may be for this reason. The other interpretation is that adversarial examples are generated based on non-robust features of the data. Ilyas et al. [68] claimed, "adversarial vulnerability is a direct result of our models sensitivity to well-generalizing features in the data." Through a series of experiments, they draw the conclusion that adversarial examples are not bugs, but features. The features for prediction can be classified as robust and non-robust, if they exist in standard classification tasks. Both of them are used for predicting the truth label which is pre-defined. But small perturbations can be added on non-robust features to make the final prediction incorrect. Hence, adversarial examples appear no matter in image, text, or other domains.

2.4 General Classifications of Adversarial Examples

Figure 5 is a general classification of adversarial attacks and defenses. The classification information is summarized from the relevant literature, including image, text, video, and audio.

2.4.1 Taxonomy of Adversarial Attacks

Adversarial attacks can be conducted in white-box and black-box scenarios. In the white-box scenario, adversaries

---

2. Relu and ELU refer to rectified linear units and exponential linear units respectively. ELU is designed for alleviating the vanishing gradient problem of relu.
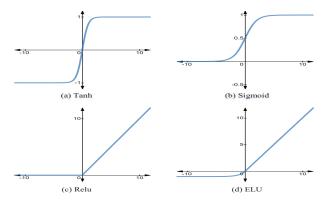
Fig. 4: Some of the main activation functions in DNNs

have full access to target models and they can obtain excellent adversarial examples by the use of models' architectures, parameters, and weights, which usually perform better than those in black-box attacks. In the black-box scenario, adversaries have no or little knowledge (*e.g.*, logits[3]) about target models. They generally train substitution models and utilize the transferability [44], [69] of adversarial examples to carry out a black-box attack. Besides, there is another way to conduct a black-box attack if target models can be queried. For querying-based methods, adversaries modify the inputs by observing the outputs after each query, and then the process is repeated until adversarial examples are obtained.

According to the purpose of adversaries, adversarial attacks can be divided into targeted and non-targeted attacks. In the targeted attack, the generated adversarial example x' is purposefully classified as class t, which is the target of an adversary. This process mainly relies on increasing the confidence score (*i.e.*, the logit or the output of softmax layer) of class t. In the non-targeted attack, the adversary only aims at fooling the model. The result y' can be any class except for y. Contrary to the targeted attack, non-targeted attack operates via reducing the confidence score of the correct class y.

In texts, there is another way to classify attacks by adversarial texts, which is shown in Figure 6. According to the modified objects, they can be divided into character-level , word-level, and sentence-level attacks. The character-level attack operates on the characters, including letters, special symbols, and numbers. Different from character-level attack, the word-level attack modifies words by synonyms, typos, genre specific keywords, etc. Currently, the way of sentence-level attack is to add crafted sentences into the inputs like the works of [26], [27].

2.4.2 Taxonomy of Defenses Against Adversarial Attacks

Inspired by [70], the goals of defense are to protect DNN-based systems from adversarial attacks and to evaluate the robustness of these systems in the worst-case. For the former, defenders try to detect adversarial examples and bar them from the DNN models. The latter is that defenders have trained the models through various ways to enhance

3. Logit is the input of a softmax layer.

their robustness for making the attack fail. Accordingly, defense can be divided into detection and model enhancement. The general way of detection is to detect the inputs. The other is achieved by enhancing the robustness of DNNs in training process, including adversarial training, changing the loss functions of models, testing, and verification methods. In texts, spelling check and adversarial training are two major ways to defend against adversarial attacks. The spelling check is a special detection method in NLP and the other is a general approach used in image, text, audio, etc.

2.5 Evaluation of Adversarial Examples on Effectiveness

The performance evaluation of adversarial examples is an open-ended question, which reflects the ability to fool DNNs. Researchers have used different standards to evaluate their performance. As far as we know, researchers generally evaluate the attacks on target models by accuracy rate or error rate.

- **accuracy rate**: It refers to the ratio of correct discrimination on the inputs. The lower the accuracy rate is, the more effective the adversarial examples are.
- **error rate**: It is the ratio of incorrect discrimination on the inputs, which is opposite to the accuracy rate. The higher the error rate is, the more effective the adversarial examples are.

Some researchers prefer to use the difference between the accuracy before and after attacks, because it shows the effect more intuitively. These evaluation methods are also used in defense.

2.6 Metric on Imperceptibility

In adversarial attacks, the basic assumption is that adversarial examples should be invisible to human eyes. A lot of metrics are adopted for measuring the perceptual similarity between adversarial examples and the original ones, like L_0 [71]–[73], L_2 [73]–[76], and L_∞ [12], [44], [76]–[79] in image domain. Unfortunately, these metrics cannot be directly applied to texts. In this section, we first give some metrics in image domain, and then we present some metrics in texts.

2.6.1 Metrics in image domain

In image domain, most recent studies adopt L_p distance to quantify the imperceptibility and similarity between adversarial examples and the original ones. The generalized term for L_p distance is shown in formula (1):

$$\|\triangle c\|_p = \sqrt[p]{\sum_{i=1}^{n} |c_i' - c_i|^p} \tag{1}$$

where $\triangle c$ represents the perturbations. c_i' and c_i are the i-th factors in n-dimensional vectors $\vec{c'}$ and $\vec{c}$ respectively. Formula (1) represents a series of distances, where p could be 0, 2, ∞, and so on. Specially, when p is equal to zero, $\|\triangle c\|_0 = \sum bool(c_i \neq 0)$. *bool* is a logical function with 0 or 1 value. In image domain, L_0, L_2, and L_∞ are the three most frequently used norms to measure adversarial examples.
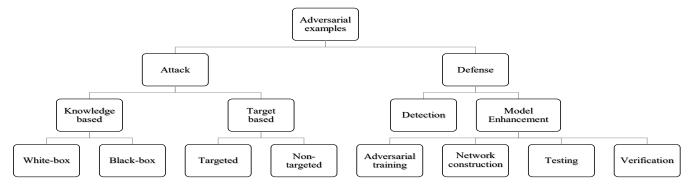
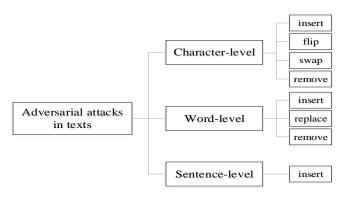Fig. 5: General categorization of adversarial attacks and defenses

Fig. 6: Classification of adversarial example in NLP based on text operation

- L_0 distance evaluates the number of changed pixels before and after modifications. It seems like edit distance, but it does not directly work in texts. Results of altered words in texts are varied. Some of them are similar to original words and the others may be contrary, even though the L_0 distances of them are equal.
- L_2 represents the Euclidean distance. The original Euclidean distance is the beeline from one point to another in Euclidean space. As the mapping of image, text, or others to this space, it acts as a metric to calculate the similarity between two objects, which are represented as the vectors.
- L_∞ distance measures the maximum change, which is shown in formula (2):

$$\|\triangle c\|_\infty = \max(|c_1' - c_1|, \ldots, |c_n' - c_n|) \qquad (2)$$

max is a function to choose the maximum factor. Although L_∞ distance is regarded as the optimal distance metric in image, but it may fail in texts. The altered words may not exist in pre-trained dictionary. Hence, they are considered to be unknown words and their word vectors are also unknown. As a result, the L_∞ distance is hard to calculate.

There are other metrics (*e.g.*, structural similarity [80], perturbation sensitivity [81]) which are typical methods for image. They cannot be directly used in texts either because of the different data types, which result in different generation methods of adversarial examples. Hence, available metrics are needed in texts to guarantee the quality of adversarial examples.

2.6.2 Metrics in texts

Apart from imperceptibility, a good adversarial example in texts must convey the same semantic meaning with the original one, so that metrics are required to ensure this case. In the following part, we describe some metrics used in the pertinent studies on adversarial texts.

Euclidean Distance. For two given word vectors $\vec{m} = (m_1, m_2, \ldots, m_k)$ and $\vec{n} = (n_1, n_2, \ldots, n_k)$, the Euclidean distance of them is shown in formula (3):

$$D(\vec{m}, \vec{n}) = \sqrt{(m_1 - n_1)^2 + \ldots + (m_k - n_k)^2} \qquad (3)$$

where m_i and n_i are the i-th factors in the k-dimensional vectors respectively. The lower the distance is, the more similar they are. But Euclidean distance is more frequently used in image domain [73]–[76] with a generalized term called L_2 norm or L_2 distance.

Cosine Distance. Cosine distance is also a computational method for semantic similarity. It calculates the cosine value of the angle between two vectors. Compared with Euclidean distance, the cosine distance pays more attention to the difference between the directions of two vectors. The more consistent their directions are, the more similar they are. For two given word vectors $\vec{m}$ and $\vec{n}$, the cosine similarity of them is shown in formula (4):

$$D(\vec{m}, \vec{n}) = \frac{\vec{m} \cdot \vec{n}}{\|m\| \cdot \|n\|} = \frac{\sum\limits_{i=1}^{k} m_i \times n_i}{\sqrt{\sum\limits_{i=1}^{k} (m_i)^2} \times \sqrt{\sum\limits_{i=1}^{k} (n_i)^2}} \qquad (4)$$

Jaccard Similarity Coefficient. For two given sets A and B, their Jaccard similarity coefficient $J(A, B)$ is shown in formula (5):

$$J(A, B) = |A \cap B| / |A \cup B| \qquad (5)$$

where $0 \leq J(A, B) \leq 1$. It means that the closer the value of $J(A, B)$ is to 1, the more similar they are. In texts, intersection $A \cap B$ refers to similar words in the samples and union $A \cup B$ is all words without duplication.

Word Movers Distance (WMD). WMD [82] is a variation of Earth Mover's Distance (EMD) [83]. It is used to measure the dissimilarity between two text documents, relying on the

traveling distance from embedded words of one document to another. The lower the value of WMD is, the more similar the two texts are.

Edit Distance. Edit distance is a way to measure the minimum modifications by turning a string to another. The lower it is, the more similar the two strings are. It can be applied to computational biology and natural language processing. Levenshtein distance [84] is also known as edit distance with insertion, deletion, replacement operations in the work of [22].

These metrics are applied to different objects. Among them, Euclidean distance, cosine distance, and WMD are used on vectors. Adversarial examples and the original ones in texts are transformed into vectors, and then these three methods are applied to calculate the distance between the vectors. On the contrary, Jaccard similarity coefficient and edit distance are directly used on text inputs without the need to convert raw texts to vectors.

Particularly, Michel et al. [85] proposed a natural criterion for adversarial texts on sequence-to-sequence models. This work focuses on evaluating the semantic equivalence between adversarial examples and the original ones. Experimental results show that strict constraints are useful in this work for keeping meaning-preserving, but whether it is better than the above metrics still needs further confirmation.

2.7 Datasets in Texts

To make data more accessible to those who need it, we collect some commonly used public datasets in NLP tasks and give some brief introductions about them. Table 2 is their applications in different NLP tasks and Table 3 is some other datasets used in research works.

AG's News[4]: AG's News is a set of news with more than one million articles and it is gathered from over 2,000 news sources by an academic news search engine named Come-ToMyHead. The provided DB version and XML version can be downloaded for any non-commercial use.

DBPedia Ontology[5]: DBPedia is a dataset with structured content from the information created in various Wikimedia projects. It has over 68 classes with 2,795 different properties. Now there are more than 4 million instances included in this dataset.

Amazon Review[6]: The Amazon review dataset has nearly 35 million reviews spanning Jun 1995 to March 2013, including product and user information, ratings, and a plaintext review. It is collected by over 6 million users in more than 2 million products and categorized into 33 classes with the size ranging from KB to GB.

Yahoo! Answers[7]: The corpus contains 4 million questions and their answers, which can be easily used in the question-answer system. Besides that, a topic classification dataset is also constructed with some main classes.

Yelp Reviews[8]: The provided data is made available by Yelp to enable researchers or students to develop academic

projects. It contains 4.7 million user reviews with the types of JSON files and SQL files.

Movie Review (MR)[9]: MR is a labeled dataset concerning sentiment polarity, subjective rating and sentences with subjectivity status or polarity. Probably because it is labeled by manual works, the size of this dataset is smaller than others, with a maximum of dozens of MB.

MPQA Opinion Corpus[10]: The Multi-Perspective Question Answering (MPQA) Opinion Corpus is collected from a wide variety of news sources and annotated for opinions or other private states. Three different versions are available to people by the MITRE Corporation. The higher the version is, the richer the contents are.

Internet Movie Database (IMDB)[11]: IMDB is crawled from the Internet, including 50,000 positive and negative reviews. The average length of the review is nearly 200 words. It is usually used for sentiment classification, including more data than other similar datasets. IMDB also contains the additional unlabeled data, raw text, and already processed data.

SNLI Corpus[12]: The Stanford Natural Language Inference (SNLI) Corpus is a collection with manually labeled data mainly for natural language inference (NLI) task. There are nearly five hundred thousand sentence pairs written by humans in a grounded context. More details about this corpus can be seen in the work of [93].

TABLE 2: Applications of some commonly used public datasets

dataset	application in the work	task
AG's News	[22], [24]	classification
DBPedia	[22], [25], [26]	classification
Amazon Review	[22]	classification
Yahoo! Answers	[22]	classification
Yelp Reviews	[22]	classification
Movie Review	[20], [25], [26]	sentiment analysis
MPQA	[26]	classification
IMDB	[20], [23], [25], [31], [32]	sentiment analysis
SNLI Corpus	[23], [28]	textual entailment, NLI

3 ADVERSARIAL ATTACKS FOR CLASSIFICATION IN TEXTS

The majority of recent adversarial attacks in texts are related to classification tasks. Hence, we first introduce the adversarial attacks with this aspect. In this section, we divide them into two parts based on the desire of attackers. Technical details and corresponding comments of each attack described below are given to make them more clear to readers.

---

4. http://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html
5. https://wiki.dbpedia.org/services-resources/ontology
6. http://snap.stanford.edu/data/web-Amazon.html
7. https://sourceforge.net/projects/yahoodataset/
8. https://www.yelp.com/dataset/download

9. http://www.cs.cornell.edu/people/pabo/movie-review-data/
10. http://mpqa.cs.pitt.edu/
11. http://ai.stanford.edu/~amaas/data/sentiment/
12. https://nlp.stanford.edu/projects/snli/
13. http://riejohnson.com/cnn_data.html
14. https://stanford-qa.com
15. http://www.nyu.edu/projects/bowman/multinli/
16. http://movieqa.cs.toronto.edu/leaderboard/
17. https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html
18. http://www.daviddlewis.com/resources/testcollections/reuters21578/

TABLE 3: Other datasets used in research works

dataset	application in the work	task	source
Enron Spam	[22]	spam e-mail detection	-
Twitter dataset	[31]	gender prediction	[86]
Elec[13]	[25]	sentiment analysis	[87]
RCV1[13]	[25]	classification	[87]
FCE-public	[25]	grammatical error detection	-
Stanford Sentiment Treebank	[24], [88]	sentiment analysis	-
Stanford Question Answering Dataset[14]	[27]	reading comprehension	[89]
MultiNLI[15]	[28]	natural language inference	[90]
MovieQA dataset[16]	[91]	attacking reading system	[92]
Customer review dataset[17]	[26]	sentiment analysis	-
Reuters[18]	[32]	classification	-

3.1 Non-targeted Attacks for Classification

In this part, the following studies are all non-targeted attacks in texts. For this kind of attacks, attackers do not care about the category of misclassification. The non-targeted attacks can be subdivided into FGSM-based, optimization-based, and importance-based ones. Among them, the importance-based attacks are those by modifying important words, which highly affect the prediction results.

3.1.1 FGSM-based Approach

Studies on adversarial examples in image domain are more active than those in texts. Hence, researchers try to employ approaches in image domain to texts and achieve better results. They propose some efficient approaches based on FGSM, which are described below.

As far as we know, Papernot et al. [21] first studied the problem of adversarial examples in texts and contributed to producing adversarial input sequences. The authors leveraged computational graph unfolding [94] to evaluate the forward derivative [71], which is related to the embedding inputs of word sequences. The results were then calculated by FGSM to find the adversarial perturbations. However, the corresponding vectors of modified words might not exist. To solve this mapping problem, they set up a specific dictionary to select words for replacing the original ones. Although their adversarial sequences can make LSTM model produce wrong predictions, the words in input sequences are randomly chosen for substitution, so that the probability of grammatical errors in adversarial examples is very high.

Similar to [21], Samanta et al. [31] utilized the concept of FGSM to evaluate the important or salient words, which deeply affected the results of classification when they were removed. Then, three modification strategies (*i.e.*, insertion, replacement, and deletion) were introduced to craft top k words with highest importance, where k was a threshold. Except for the deletion strategy, both insertion and replacement on top k words required an additional dictionary for substitution. Thus, the authors established a pool of candidates for each word in the experiment, including synonyms, typos, and type-specific keywords. However, this process consumes a great deal of time and there may be no candidate pool for some top k words in the actual inputs. This method is also an importance-based approach.

3.1.2 Optimization-based Approach

Different from other methods, Sato et al. [25] operated in the embedding space of inputs and proposed a method named

iAdv-Text. Its core process could be seen as an optimization problem, which was to jointly minimize objection function $\mathcal{J}_{iAdvT}(D, W)$ on entire training dataset D with parameters W. The optimization procedure is shown in formula (6):

$$\mathcal{J}_{iAdvT}(D, W) = \frac{1}{|D|} \arg\min_{W} \{ \sum_{(\hat{X}, \hat{Y}) \in D} \ell(\hat{X}, \hat{Y}, W) + \lambda \sum_{(\hat{X}, \hat{Y}) \in D} \alpha_{iAdvT} \} \quad (6)$$

where $\hat{X}$ and $\hat{Y}$ are the inputs and labels respectively. λ is a hyper-parameter to balance the two loss functions. $\ell(\hat{X}, \hat{Y}, W)$ is the loss function of individual training sample $(\hat{X}, \hat{Y})$ in D. α_{iAdvT} is a maximization process to find the worst case weights of the direction vectors. Its formula is shown in (7):

$$\alpha_{iAdvT} = \arg\max_{\alpha, \|\alpha\| \leq \epsilon} \{ \ell(\vec{w} + \sum_{k=1}^{|V|} a_k d_k, \hat{Y}, W) \} \quad (7)$$

where $\sum_{k=1}^{|V|} a_k d_k$ is the perturbation generated from each input on its word embedding vector $\vec{w}$. ϵ is a hyper-parameter to control adversarial perturbation. a_k is the k-th factor of a $|V|$-dimensional word embedding vector α. d_k is the k-th factor of a $|V|$-dimensional direction vector $\vec{d}$, which is a mapping from one word to another in embedding space. Since α_{iAdvT} in formula (7) was difficult to calculate, the authors used formula (8) instead:

$$\alpha_{iAdvT} = \frac{\epsilon g}{\|g\|_2}, g = \nabla_{\alpha} \ell(\vec{w} + \sum_{k=1}^{|V|} a_k d_k, \hat{Y}, W) \quad (8)$$

iAdv-Text restricts the direction of perturbations to find a substitution, which is in a pre-defined vocabulary rather than an unknown word. Meanwhile, the authors also use cosine similarity to select better perturbations, so that the readability and semantic similarity are well kept.

Similarly, Gong et al. [32] also searched for adversarial perturbations in embedding space. Even though WMD is used by the authors to measure the similarity between adversarial examples and the original ones, the readability of generated adversarial examples seems worse than those in iAdv-Text.

3.1.3 Importance-based Approach

Unlike previous white-box methods [21], [31], little attention is paid to black-box attacks with adversarial texts. Gao et

al. [22] proposed DeepWordBug to generate adversarial examples in black-box scenario. The whole process is divided into two stages. The first stage was to determine which important words to change, and the second stage was to create imperceptible perturbations for fooling the models. The calculation process for the first stage was shown in formula (9):

$$
\begin{aligned}
CS(x_i) =&[F(x_1,\ldots,x_{i-1},x_i) - F(x_1,x_2,\ldots,x_{i-1})]+\\
&\lambda[F(x_i,x_{i+1},\ldots,x_n) - F(x_{i+1},\ldots,x_n)]
\end{aligned} \quad (9)
$$

where $CS(x_i)$ represents the importance score of the i-th word in the input $(x_1,\ldots,x_n)$, which was evaluated by the function F. λ is a hyper-parameter. In second stage, similar modifications like swap, flip, deletion, and insertion were applied to manipulate the characters of important words. Meanwhile, the authors used edit distance to preserve the readability of adversarial examples. A variant method was proposed by Wang et al. [95] to attack a classification system with Chinese data, which showed that adversarial examples could be other languages, not limited to English.

Li et al. [20] proposed an attack framework TextBugger for generating adversarial examples, which could mislead the deep learning-based text understanding system in both black-box and white-box settings. In the white-box scenario, Jacobian matrix J was used to calculate the importance of each word as follows:

$$
C_{x_i} = J_{F(i,y)} = \frac{\partial F_y(x)}{\partial x_i} \quad (10)
$$

where $F_y(\cdot)$ represents the confidence value of class y. C_{x_i} is the important score of the i-th word in the input x. Then, similar modification strategies like DeepWordBug [22] are used to generate both character-level and word-level adversarial examples. In the black-box scenario, the authors segmented documents into sequences, and then they queried the target model to filter out sentences with different predicted labels from the original ones. The odd sequences were sorted in an inverse order by their confidence score. Subsequently, the important words were calculated by removing operation in formula (11):

$$
\begin{aligned}
C_{x_i} =&F_y\left(x_1,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_n\right)\\
&- F_y\left(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n\right)
\end{aligned} \quad (11)
$$

The final modification process was the same as that in the white-box setting.

3.2 Targeted Attacks for Classification

For targeted attacks, attackers purposefully control the category of outputs which they desire and generate semantic-preservation adversarial examples. Currently, the costs of targeted attacks are larger than non-targeted ones due to their special needs. In the following part, we describe some representative targeted attacks in recent works.

3.2.1 FGSM-based Approach

Different from works in [21], [31], Liang et al. [26] first demonstrated that FGSM could not be directly applied to texts. Because the input space of texts is discrete, and that of image data is continuous. Continuous images have tolerance of tiny perturbations, but texts are not. Instead,

the authors only utilized FGSM to determine what, where, and how to insert, remove, and modify the inputs. They conducted two kinds of attacks in different scenarios and used the natural language watermarking [96] technique to make generated adversarial examples compromise their utilities.

The authors carried out adversarial attacks in both white-box and black-box settings. In the white-box scenario, they defined the conceptions of hot training phrases and hot sample phrases. These two objects were both obtained by leveraging the back-propagation algorithm to compute the cost gradients of samples. The former one shed light on what to insert, and the latter one implied where to insert, remove, and modify. In the black-box scenario, the authors used the idea of fuzzing technique [97] for reference to obtain hot training phrases and hot sample phrases. A core assumption was that the target model could be queried. Samples were fed to target model, and then isometric whitespace was used to substitute origin word each time. The difference between the results before and after modification was the deviation of each word. The larger it was, the more significant the corresponding word was to the classification. Hence, hot training phrases were the most frequent words in the set of inputs, which consisted of the largest deviation words for each training sample. And hot sample phrases were the words with largest deviation for every test sample.

3.2.2 Optimization-based Approach

Like one pixel attack [72], a similar method named HotFlip was proposed by Ebrahimi et al. [24]. HotFlip was a white-box attack in texts, which relied on an atomic flip operation to swap one character with another by gradient computation. The flip operation was represented by formula (12):

$$
\begin{aligned}
\vec{v}_{ijb} =&(\vec{0},\ldots;(\vec{0},\ldots(0,0,\ldots,0,-1,0,\ldots,1,0)_j,\\
&\ldots,\vec{0})_i;\vec{0},\ldots)
\end{aligned} \quad (12)
$$

The formula (12) means that the j-th character of i-th word in an example is changed from a to b, which are both characters at a-th and b-th places in the alphabet. -1 and 1 in formula (12) are the corresponding positions for a and b respectively. The change from directional derivative along this vector is calculated to find the biggest increase in the loss $J(x,y)$. The process of calculation is shown in formula (13):

$$
\max \nabla_x J(x,y)^T \cdot \vec{v}_{ijb} = \max_{ijb} \frac{\partial J^{(b)}}{\partial x_{ij}} - \frac{\partial J^{(a)}}{\partial x_{ij}} \quad (13)
$$

where x_{ij} is a one-hot vector, which denote the j-th character of i-th word. y refers to the corresponding label vector. T is a transpose function. Apart from character-level attack, HotFlip could also be used on word-level by different modifications. Although HotFlip performs well, only a few successful adversarial examples are generated with one or two flips under the strict constraints, so that it is not suitable for a large-scale experiment.

A derivative method DISTFLIP was proposed by Gil et al. [33]. They distilled the knowledge of the procedure in HotFlip, which was used to train a black-box model. Through the training model, the authors generated adversarial examples to conduct a black-box attack. This method

performs well than HotFlip on a toxicity classifier [98], and its run-time in generating adversarial examples is ten times faster than HotFlip. But whether this method can distill the knowledge of any white-box attack remains to be verified.

Considering the limitation of gradient optimization [21], [24], [41] in the black-box setting, Alzantot et al. [23] proposed an optimization method based on genetic algorithm [99], [100]. The authors randomly selected words from the inputs and computed their nearest neighbors by Euclidean Distance in GloVe embedding space [101]. These nearest neighbors were filtered based on language model scores [102] to make them suitable for the surroundings, so that only high-ranking words with the highest scores were kept. The substitutions which would maximize the probability of target label were picked from the remaining words. At the same time, above operations were conducted several times to get a generation. If the predicted label of a modified sample in a generation was not the target, two samples were randomly chosen each time as parents to generated the next generation, and then the same process was repeated on the next generation. This optimization procedure is to find a successful attack by genetic algorithm. In this method, random selection words in the sequence to substitute are full of uncertainty. These substitutions may be meaningless, even though the target label is changed.

3.2.3 Summary of Adversarial Attacks for Classification

These attacks for classification are either popular or representative ones in recent studies. Some main attributes of them are summarized in Table 4. We can see that the majority of white-box attacks are related to gradients in Table 4. Gradient-based optimization methods are widely used in image with many variants (*e.g.*, [106], [107]), which can also be applied to texts. But there exist some shortcomings by the use of gradients, which are the vanishing and exploding gradient problems [108], [109], and limitations of the access to target models. Meanwhile, gradient masking [110] can make gradients useless in some cases, leading to failure in gradient-based methods. Even though gradient masking is proved to be a failed defense, gradient-based methods are not as effective as we think.

Besides, we have reviewed many adversarial attacks here, but how they perform and which one is better are still unclear. To make a good comparison on these attacks, we analyze and choose the same datasets they use. In Table 2 and Table 3, we can see that IMDB, DBpedia, and MR are three commonly used datasets. But the experimental result of Text-fool [26] on MR is not found. Hence, comparisons are made on IMDB and DBpedia and results are shown in Table 5. The results of TextBugger [20] on IMDB and those of Text-fool [26] on DBPedia are in white-box scenario. The success rate of Text-fool is the average. ϵ in the work of [32] is equal to point four. Although some attacks are conducted on several models and have both black-box and white-box methods, we only list a good one for comparison. In Table 5, we can see that white-box attacks are generally better than

black-box ones except for the genetic method in Alzantot et al. [23]. This phenomenon may be related to the lack of black-box attacks. Some of them are perhaps not worse than the white-box. For example, black-box attacks in TextBugger [20] reach one hundred percent of success rate on several physical systems. On the other hand, some non-targeted attacks perform better than targeted ones, but the comparison is opposite considering the average of success rate. The optimization-based methods are much more superior than others, mainly because of the stricter constraints than other attacks.

Furthermore, good adversarial texts not only achieve a high success rate to fool DNNs, but also need to have good readability, semantic similarity, and imperceptibility. Hence, we judge through the generated examples (in Table 6) in these methods. Modifications on texts are generally divided into character-level, word-level, and sentence-level. The character-level operates on the characters and the others modify words and sentences respectively. In Table 6, the word-level attacks with synonym seem more imperceptible than the character-level ones, although people are robust against misspellings [111]. But some character-level methods also perform very well like HotFlip. Generally, the more operations there are, the easier it is to be perceived. The more imperceptible the perturbations are, the better the readability and semantic similarity will be.

4 ADVERSARIAL ATTACKS ON OTHER NLP TASKS

We have reviewed adversarial attacks for classification task above. Next, we solve some other puzzles on adversarial texts, such as what other kinds of NLP tasks or applications can be attacked by adversarial examples and how they are generated in these cases.

4.1 Attack on Reading Comprehension Systems

To explore whether reading comprehension systems are vulnerable to adversarial examples, Jia et al. [27] inserted adversarial perturbations into paragraphs to test the systems without changing the answers or misleading humans. They extracted nouns and adjectives in the question and replaced them with antonyms. Meanwhile, named entities and numbers were changed by the nearest word in GloVe embedding space [101]. The modified question was transformed into a declarative sentence as the adversarial perturbation, which was then concatenated to the end of the original paragraph. This process is called ADDSENT by the authors.

Another way ADDANY randomly chose words of the sentences to craft. Compared with ADDSENT, ADDANY does not consider the grammaticality of sentences, and it needs to query the model several times. The core idea of this work is to draw the attention of models on the generated sequences rather than original sequences to produce incorrect answers. Mudrakarta et al. [35] studied adversarial examples on question answering system, and part of their works can strengthen the attacks proposed by Jia et al. [27].

4.2 Attack on Natural Language Inference Models

Except for reading comprehension systems [27], Minervini et al. [28] chose the NLI systems as the target. They cast

TABLE 4: Attributes of attacks for classification

Method	White/Black box	Targeted/Non-targeted	Model	Metric	Gradient-related
TextBugger [20]	Both two	Non-targeted	LR,char-CNN [103],CNN [104]	multiple metrics	No
Papernot et al. [21]	White box	Non-targeted	LSTM[19]		Yes
DeepWordBug [22]	Black box	Non-targeted	LSTM,char-CNN [103]	Edit Distance	No
Alzantot et al. [23]	Black box	Targeted	LSTM[20],RNN[21]	Euclidean Distance	No
HotFlip [24]	White box	Targeted	CNN [104],charCNN-LSTM [105]	Consine Similarity	Yes
iAdv-Text [25]	White box	Non-targeted	LSTM	Consine Similarity	Yes
Text-fool [26]	Both two	Targeted	char-CNN [103]		Yes
Samanta et al. [31]	White box	Non-targeted	CNN		Yes
Gong et al. [32]	White box	Targeted	CNN	Word Mover Distance(WMD)	Yes
DISTFLIP [33]	Black box	Non-targeted	GRU		No

TABLE 5: Experimental results of adversarial attacks on IMDB and DBPedia

dataset	method	type	black/white	targeted/non-targeted	model	success rate
IMDB	TextBugger [20]	importance-based	white	non-targeted	char-CNN	86.7%
	Alzantot et al. [23]	optimization-based	black	targeted	LSTM	97%
	iAdv-Text [25]	optimization-based	white	non-targeted	LSTM	93.92%
	Samanta et al. [31]	FGSM-based	white	non-targeted	CNN	67.45%
	Gong et al. [32]	optimization-based	white	targeted	CNN	86.66%
DBPedia	DeepWordBug [22]	importance-based	black	non-targeted	LSTM	74.32%
	iAdv-Text [25]	optimization-based	white	non-targeted	LSTM	99.01%
	Text-fool [26]	FGSM-based	white	targeted	char-CNN	84.7%

the generation of adversarial examples as an optimization problem to capture samples, which could bread the First-Order Logic (FOL) constraints added in NLI. The authors maximized the proposed inconsistency loss J_I to search for substitution sets S (*i.e.*, adversarial examples) by using a language model as follows:

$$maximize_S J_I(S) = [p(S; body) - p(S; head)]_+ ,$$
$$s.t. \log p_L(S) \leq \tau \tag{14}$$

where $[x]_+ = \max(0, x)$. $p_L(S)$ refers to the probability of the sentences in S.

- τ: a threshold on the perplexity of generated sequences
- $X_1, \ldots, X_n$: the set of universally quantified variables in a rule to sequences in S
- $S = \{X_1 \to s_1, \ldots, X_n \to s_n\}$: a mapping from $\{X_1, \ldots, X_n\}$
- $p(S; body)$ and $p(S; head)$: probability of the given rule, after replacing X_i with the corresponding sentence S_i
- $body$ and $head$: represent the premise and the conclusion of the NLI rules

These generated sequences are used by authors to test the weaknesses of NLI systems.

4.3 Attack on Neural Machine Translation (NMT)

NMT is another target system attacked by Belinkov et al. [29]. They devised adversarial examples depending on natural and synthetic language errors, including typos, misspellings, or others. Then, syntactically adversarial examples are modified by random or keyboard typo types. The experiments are made on three different NMT systems [112], [113], and the results show that these examples could also effectively fool NMT systems.

Similarly, Ebrahimi et al. [30] conducted an adversarial attack on NMT by employing differentiable string-edit operations. The method of generating adversarial examples is the same in their previous work [24]. Compared with Belinkov et al. [29], the authors demonstrate that adversarial examples in black-box attacks are much weaker than white-box ones in most cases.

4.4 Attack with Syntactically Controlled Paraphrase Networks (SCPNS)

Iyyer et al. [88] designed SCPNS for generating adversarial examples without decreasing the quality of the input semantics. The general process mainly relied on the encoder-decoder architecture of SCPNS. Given a sequence and a corresponding target syntax structure, the authors encoded them by a bidirectional LSTM model and decoded by the LSTM model. This process was augmented with soft attention over encoded states [114] and the copy mechanism [115]. Then, they modified the inputs to the decoder, aiming at incorporating the target syntax structure to generate adversarial examples. The syntactically adversarial sentences not only can fool pre-trained models, but also improve the robustness of them to syntactic variation. The authors also use crowdsourcing experiment to demonstrate its validity.

5 DEFENSES AGAINST ADVERSARIAL ATTACKS IN TEXTS

The constant arms race between adversarial attacks and defenses invalidates conventional wisdom quickly [64]. In fact, defense is more difficult than attack. Currently, there may be two reasons for the scarcity of defense methods. One is that a good theoretical model does not exist for complicated optimization problems of adversarial examples. The other is that tremendous amount of possible inputs may produce the targeted outputs with a very high possibility. Hence, an actually adaptive defense method is difficult. In this section,

TABLE 6: Instances of some methods in section 3

Method	Instance of adversarial example	type of modification
TextBugger [20]	**original**: reason why requesting i want to report something so can ips report stuff, or can only registered users can? if only registered users can, then i 'll request an account and it 's just not fair that i cannot edit because of this anon block shit c'mon, fucking hell. **Toxic** **modified**: reason why requesting i want to report something so can ips report stuff, or can only registered users can? if only registered users can, then i 'll request an account and it 's just not fair that i cannot edit because of this anon block shti c'mon, fuckimg helled. **Non-toxic**	multi
Papernot et al. [21]	**original**: I wouldn't rent this one even on dollar rental night. **negative** **modified**: Excellent wouldn't rent this one even on dollar rental night. **positive**	word-level
DeepWordBug [22]	**original**: This film has a special place in my heart. **positive** **modified**: This film has a special plcae in my herat. **negative**	character-level
Alzantot et al. [23]	premise: A runner wearing purple strives for the finish line. **original**: A runner wants to head for the finish line. 86%**Entailment** **modified**: A racer wants to head for the finish line. 43%**Contradiction**	word-level
HotFlip [24]	**original**: South Africas historic Soweto township marks its 100th birthday on Tuesday in a mood of optimism. 57%**World** **modified**: South Africas historic Soweto township marks its 100th birthday on Tuesday in a mooP of optimism. 95%**Sci/Tech**	character-level
iAdv-Text [25]	**original**: There is really but one thing to say about this sorry movie It should never have been made The first one one of my favourites An American Werewolf in London is a great movie with a good plot good actors and good FX But this one It stinks to heaven with a cry of helplessness. **negative** **modified**: There is really but one thing to say about that sorry movie It should never have been made The first one one of my favourites An American Werewolf in London is a great movie with a good plot good actors and good FX But this one It stinks to heaven with a cry of helplessness. **positive**	character-level
Text-fool [26]	**original**: The Old Harbor Reservation Parkways are three historic roads in the Old Harbor area of Boston. They are part of the Boston parkway system designed by Frederick Law Olmsted. They include all of William J. Day Boulevard running from Castle Island to Kosciuszko Circle along Pleasure Bay and the Old Harbor shore. The part of Columbia Road from its northeastern end at Farragut Road west to Pacuska Circle (formerly called Preble Circle). 87.3%**Building** **modified**: The Old Harbor Reservation Parkways are three ~~historic~~ roads in the Old Harbor area of Boston. Some exhibitions of Navy aircrafts were held here. They are part of the Boston parkway system designed by Frederick Law Olmsted. They include all of William J. Day Boulevard running from Cast1e Island to Kosciuszko Circle along Pleasure Bay and the Old Harbor shore. The part of Columbia Road from its northeastern end at Farragut Road west to Pacuska Circle formerly called Preble Circle. 95.7% **Means of Transportation**	multi
Samanta et al. [31]	**original**: A sprawling, overambitious, plotless comedy that has no dramatic center. It was probably intended to have an epic vision and a surrealistic flair (at least in some episodes), but the separate stories are never elevated into a meaningful whole, and the laughs are few and far between. Amusing ending though. **negative** **modified**: A sprawling, overambitious, plotless funny that has no dramatic center. It was probably intended to have an epic vision and a surrealistic flair (at least in some episodes), but the separate stories are never elevated into a greatly whole, and the laughs are little and far between amusing ending though. **positive**	word-level
Gong et al. [32]	**original**: One of those TV films you saw in the seventies that scared the hell out of you when you were a kid but still gives you an eerie feeling. No great actors or expensive production but everytime that phone rings ... ... **Label 0** **modified**: One of those TV films you saw in the seventies that scared the hell out of you when you were a kid but not gives you an considered unnerving. No great actors and/or expensive production but everytime that phone rings ... ... **Label 1**	word-level

we describe some relatively efficient methods against adversarial attacks in texts, which can be divided into detection and model enhancement methods. The former defends by detecting the inputs, and the latter is by enhancing the robustness of models.

5.1 Detection of Adversarial Examples

Adversarial examples are a kind of data with the specific purpose. Hence, it is worth considering whether detection is useful against adversarial attacks. Inspired by this view, a series of works [116]–[120] have been conducted to detect adversarial examples and perform relatively well in image domain. In texts, the ways to generate adversarial examples in some methods will produce misspellings, which could be a distinctly different feature for use. It naturally comes up with an idea to detect adversarial examples by checking out the misspellings. Gao et al. [22] used the Python autocorrect 0.3.0 package to detect the inputs. Li et al. [20] took advantage of a context-aware spelling check service[22] to do the similar work. But experimental results show that the detection is effective on character-level modifications and partly useful on word-level attacks, probably because of the differences on modification strategies. Besides, spelling check method is also not suitable for adversarial examples based on other languages like Chinese [95].

Pruthi et al. [121] proposed a word recognition model applied before DNNs to detect adversarial examples with misspellings. This model performs well on the character-level attacks. Although its experiments show that they can also detect word-level attacks, whether it is useful enough for the word-level ones is still unknown. For instance, the example of Alzantot et al. [23] in 6 only changes the word "runner" to "racer", which is hard for this method

---

22. https://azure.microsoft.com/zh-cn/services/cognitive-services/spell-check/

to judge. Wang et al. [122] proposed a defense method called Synonyms Encoding Method (SEM), which was for attacks based on synonyms substitution. The author found the synonymous sentence for an input by clustering and marked them with the same label. Their method did not modify the model or need external knowledge.

5.2 Adversarial Training

Adversarial training [44] is a widely used approach to resist adversarial attacks in image domain [44], [123]. Researchers mix adversarial examples with the original ones as a dataset for training to enhance the models' tolerance of adversarial examples. In texts, adversarial training is also used in some works [20], [22]–[24], [124]–[126] as a defense method against adversarial texts. However, this method fails in the work of [23], mainly because of the different ways to generate adversarial examples. The modifications of the others are insertion, substitution, deletion, and replacement, while the attack in [23] makes use of the genetic algorithm to search for adversarial examples.

Over-fitting is another reason why adversarial training is not always useful and only effective on its corresponding attack, which has been confirmed by Tram'er et al. [127] in image domain, but it remains to be demonstrated in texts. On the other hand, the accuracy of detection on adversarial examples increases, while the ability on original ones decreases in the re-training model by adversarial training. Hence, how to improve the performance of adversarial training is an open question waiting to be solved.

5.3 Other Methods Against Adversarial Examples

Except for adversarial training, there exist other ways to improve the robustness of models to resist adversarial attacks. Goren et al. [128] formally analyzed, defined, and quantified the notions of robustness of linear learning-to-rank-based relevance ranking function [129], aiming at improving the robustness to small perturbations of documents in the adversarial Web retrieval setting. They adopted the notions of classification robustness [12], [130] to ranking function and defined some related concepts, such as pointwise robustness, pairwise robustness, and a variance conjecture. To quantify the robustness of ranking functions, Kendall's-τ distance [131] and top change were used as normalized measures. Finally, the empirical findings supported the validity of their analyses in two families of ranking functions [132], [133].

Li et al. [134] proposed a method to enhance the robustness of NLI systems by the multi-head attention [135]. The authors determined what kind of robustness they wanted. Then, the external knowledge (*e.g.*, wordnet [136]) was added to the attention mechanism, which was linked to structured embeddings. Experimental results show that there is a significant improvement on defense against adversarial examples when the knowledge is added to the cross-encoder in their models. Another advantage is that the method does not need extra parameters and any model with attention units can use.

5.4 Summary of Defense Methods

Although defense methods have achieved better results on their corresponding works, there also exist some limitations. The strategy of spelling check performs not well on detecting word-level and sentence-level adversarial examples. Adversarial training exists an over-fitting problem and it may not work in facing with a new attack method. Model enhancement may be the chief defense strategy in the future, but it is still under exploring and has many difficulties, such as the choice of loss function and modifications of models' structure. In Table 7, we compare the three defense methods from efficiency, prospect, and extendibility. Except for spelling check in NLP, other two methods have been used in image, audio, text, etc. They also have higher prospects than spelling check which has great limitations.

TABLE 7: Comparisons of defense methods in texts from extendibility and prospect

method	extendibility	prospect	application
Spelling check	low	low	[20], [22], [121]
Adversarial training	high	middle	[20], [22]–[24], [124]
Model enhancement	high	high	[128]

6 TESTING AND VERIFICATION

Like traditional security issues, adversarial attacks and defenses are constantly changing. New adversarial attacks are proposed and then followed by new countermeasures which will be subsequently broken [137]. Hence, the formal guarantees for the behavior of DNNs are badly needed to break this loop, but it is a hard job. The technology of testing and verification helps us deal with this problem from another point. Testing is to use test cases to discover bugs in the models, and verification is to ensure the normal operation of the models under certain conditions. Both of them are important ways to improve the robustness of DNNs to defend against adversarial examples.

In this section, we first introduce testing and verification methods in image domain, and those in texts are described later. Even though testing and verification methods in image domain have not been applied to texts, but their ideas can be used in texts like FGSM and adversarial training.

6.1 Testing in Image Domain

With the increasing use of DNNs in security-critical domains, it is very significant to have a high degree of trust in the accuracy of models, especially in the presence of adversarial examples. The confidence of models' correct behavior is derived from the rigorous testing in various scenarios. More importantly, testing is helpful for understanding the internal behavior of the network, contributing to the implementation of defense methods. In the following part, we survey the approaches for testing DNNs from testing criteria and test case generation.

6.1.1 Test Case Generation

Pei et al. [139] designed a white-box framework DeepXplore to test real-world DNNs with the metric of neuron coverage.

TABLE 8: Comparison with other four methods

—	DeepConcolic [138]	DeepXplore [139]	DeepGauge [140]	DeepTest [141]	DeepCover [142]
Type of input	normal data or coverage requirements	normal data	normal data	normal data	normal data
Number of input*	single or multiple	multiple	multiple	multiple	multiple
Method of test generation*	concolic	optimisation-based	optimization-based	greedy search	symbolic execution

* Values of these two items come from the work in [138].

They leveraged differential testing to catch the differences of corresponding output between multiple DNNs. In this way, the generated adversarial examples have a high neuron coverage. Hence, DeepXplore could trigger the majority logic of the model to find out incorrect behavior without manual efforts. It performs well in the advanced deep learning systems and finds thousands of corner cases which make the systems crash. However, the limitation of DeepXplore is that if all the DNNs make incorrect judgments, it is hard to know where is wrong and how to solve it.

Wicker et al. [143] presented a feature-guided approach to test the resilience of DNNs in the black-box scenario. The authors treated the process of generating adversarial cases as a two-player turn-based stochastic game. Over the process of this gameplay, adversarial examples are found with the asymptotic optimal strategy based on Monte Carlo tree search (MCTS) [144] algorithm, and the authors can evaluate the robustness of DNNs via this process.

Sun et al. [138] presented DeepConcolic, which was the first attempt to apply traditional consoles testing method to well-known DNNs. DeepConcolic iteratively used concrete execution and symbolic analysis to generate test suits, aiming at reaching a high coverage and discovering adversarial examples by a robustness oracle. The authors also compared with other testing methods [139]–[142] shown in Table 8. In terms of input data, DeepConcolic could start with a single input or coverage requirements as inputs to achieve a better coverage. For the performance, DeepConcolic could achieve higher coverage than DeepXplore, but run slower than it.

6.1.2 Testing Criteria

Different from single neuron coverage [139], Ma et al. [140] proposed a multi-granularity testing coverage criteria to measure accuracy and detect erroneous behavior. They took advantage of four methods [44], [71], [73], [106] to generate adversarial test data to explore the new internal states of models. The increasing coverage shows that the larger the coverage is, the more possible the defects are to be checked out. Similar work is conducted by Budnik et al. [145] to explore the output space of models via an adversarial case generation approach.

Kim et al. [146] proposed a new test adequacy criterion called Surprise Adequacy for Deep Learning Systems(SADL) to test DNNs. This method measures the differences between inputs and training data, which is the foundation of the adequacy criterion. Experimental results show that adversarial examples can be well judged by this method. On the other hand, it also improves the accuracy of DNNs against adversarial examples by retraining.

6.2 Verification in Image Domain

Researchers think that testing is insufficient to guarantee the security of DNNs, especially with unusual inputs like adversarial examples. Edsger W. Dijkstra once said, "*testing shows the presence, not the absence of bugs*". Hence, verification techniques on DNNs are needed to study more effective defense methods in adversarial settings. But verification of the robustness of machine learning models to adversarial examples is still in its infancy [147], resulting in few researches on related aspects. We group the verification methods into three aspects in the following part: search-based, global optimization, and over-approximation approaches.

6.2.1 Global Optimization Approach

Katz et al. [148] presented a novel system named Reluplex to verify DNNs based on Satisfiability Modulo Theory (SMT) [149] solver. They transformed the verification into linear optimization problems with Rectified Linear Unit (ReLU) [66] activation functions. Reluplex is used to find adversarial inputs with the local robustness feature on the ACAS Xu networks [11], but it fails on large networks with the global variant.

For ReLU networks, Tjeng et al. [150] regard the verification as a Mixed Integer Linear Programming (MILP) [151] problem. They evaluated robustness to adversarial examples from minimum adversarial distortion [152] and adversarial test accuracy [153]. The runtime of their work is faster than Reluplex [148] with a high adversarial test accuracy, but the same limitation is that it remains a problem to scale it to large networks.

Unlike existing solver-based methods (*e.g.*, SMT), Wang et al. [154] presented ReluVal by interval arithmetic [155] to guarantee the correct operations of DNNs in the presence of adversarial examples. They repeatedly partitioned input intervals to find out whether the corresponding output intervals violated security property. By contrast, this method is more effective than Reluplex and performs well on finding adversarial inputs.

6.2.2 Search-based Approach

Huang et al. [156] proposed a new verification framework based on SMT to verify neural network structures. This method relies on the discrete search space and analyzes the output of each layer to search for adversarial perturbations, but the authors find that SMT theory is only suitable for small networks in practice. On the other hand, this framework is limited to many assumptions and some functions in it are unclear.

Different from other works, Narodytska et al. [157] verified the secure properties on the binarized neural networks(BNNs) [158]. They leveraged the counterexample-

guided search [159] procedure and modern SAT solvers to study the robustness and equivalence of models. The inputs of models are judged whether they are adversarial examples by two encoding structures Gen and Ver. This method can easily find adversarial examples for up to 95 percent of considered images on the MNIST dataset [160], but it works on the middle-sized BNNs rather than large networks.

6.2.3 Over-approximation Approach

Gehr et al. [161] introduced abstract transformers, which could get the outputs of CNN-based models with ReLU. The authors evaluated this approach on a pre-trained defense network [162] to verify its robustness. Experimental results showed that the FGSM attack could be effectively prevented. They also made some comparisons with Reluplex on both small and large networks. The stare-of-the-art Reluplex performed worse than it in the verification of properties and time consumption.

Weng et al. [163] designed two kinds of algorithm to evaluate lower bounds of minimum adversarial distortion, via linear approximations and bounding the local Lipschitz constant. Their methods can be applied to defended networks, especially for adversarial training, to evaluate their effectiveness.

6.3 Testing and Verification in Texts

Unlike various works in image domain, the number of related papers on testing and verification in texts is two and one respectively. We introduce them below and make comparisons with the similar works in image domain.

Testing. Blohm et al. [91] generated adversarial examples to discover the limitations of their machine reading comprehension model. Through several ways for testing, including word-level and sentence-level in different scenarios [51], the authors found that their model was robust against meaning-preserving lexical substitutions, but failed in importance-based attacks. Besides, experimental results show that some other attributions (*e.g.,* answer by elimination via ranking plausibility [164]) should be added to improve the performance. Cheng et al. [165] proposed a projected gradient method to test the robustness of sequence-to-sequence (seq2seq) models. They found that seq2seq models were more robust to adversarial attacks than CNN-based classifiers. Meanwhile, they addressed the challenges caused by discrete inputs, making it possible to apply methods in image domain to texts.

Compared with testing methods in image domain, related works in texts are very rare. Except for quantity, testing in [91] and [165] is different from others in image domain. Works in texts are to judge the robustness of models by observing its performance of the test cases. However, similar approaches in image domain (*e.g.,* DeepXplore [139] and DeepGauge [140]) have considered the neuron coverage of the models, which is worth trying to be applied to texts. Besides, testing can be used in other NLP tasks, such as text classification, machine translation, and natural language inference.

Verification. Jia et al. [166] presented certifiably robust training by optimizing the Interval Bound Propagation (IBP) upper bound [167], which could limit the loss of worst-case perturbations and compress the living space of adversarial examples. Their method was provably robust to the attacks with word substitutions on IMDB and SNLI. As same as testing, the works of verification in texts are little either. In theory, some methods in image domain can be applied to texts, which are for enhancing the robustness of models, not for the data or others. For example, methods based on SMT are likely to be used in texts, although they need to be confirmed by experiments.

7 DISCUSSIONS

In the previous sections, detailed descriptions of adversarial attacks and defenses in texts are given, enabling readers to have a faster and better understanding of this aspect. Next, we present more general observations and challenges of this area.

7.1 Findings

From the papers we reviewed, there are many findings, and we summarize some of the major ones below.

Use of misspellings. The motivation by using misspellings is similar to one pixel attack [72] in image domain, which aims at fooling target models with indiscernible perturbations. The ideal situation is that attackers only change one or two characters to generate adversarial texts for fooling the models just like HotFlip [24]. Actually, people have to increase the number of modified characters to get more adversarial examples, while character-level modifications can result in misspellings with a high probability. Meanwhile, the study shows that humans are very robust against misspellings in written language [111], so that misspellings are used as a special kind of modifications to generate adversarial examples in texts.

Transferability. Szegedy et al. [12] first found that adversarial examples generated from a neural network could also make another model misbehave by different datasets, which reflects their transferability. Therefore, attackers can train a substitute model and utilize the transferability of adversarial examples for attack, when they have no access and query restriction to the target models. Recently, studies show that different types of adversarial attacks have different transferability [106], [168]. For instance, adversarial examples generated from one-step gradient-based methods are more transferable than iterative methods [106], but their attack abilities are opposite. Hence, the generation of adversarial examples with high transferability is not only the premise to carry out black-box attacks, but also a metric to evaluate generalized attacks.

Generation Process. We observe that many methods mainly follow a two-step process to generate adversarial examples in texts. The first step is to find important words with significant impacts on the results of classification, and then homologous modifications are used to generate adversarial texts.

Distribution. In addition, we find that there is a pronounced imbalance in the distribution of research efforts: among the 47 papers in texts we collected, three of them tackle testing and verification, only one paper specializes in defense, and the remaining is about attacks. We can see

that defense methods account for only a small portion of research in texts and these researches make up only a small part of all areas (in Figure 2). Hence, great attention should be paid to the security problem caused by adversarial examples.

7.2 Challenges

In terms of adversarial attacks, there are two major challenges in texts:

(1) Adversarial attacks in texts are not practical due to the absence of attacks in physical NLP systems. Currently, the majority of studies on adversarial texts is about theoretical models and rarely related to practical applications. We have used the adversarial examples presented in recent works [20]–[26], [31], [32] to attack ParallelDots like Figure 1, but most of the adversarial examples are ineffective and can be correctly classified. Only a few samples successfully fool this system, which means that the transferability of these adversarial examples is not good. For the physical NLP systems, we have no access to them and query may be limited sometimes, so that transferability is the main choice for attacking these physical applications. Hence, the power of transferability is the key factor for practical attacks.

(2) There do not exist well-performed adversarial examples which can fool any DNNs, which is the so-called universality. Although Wallace et al. [169] find input-agnostic sequences which can trigger specific classifications to generate universal adversarial examples, these sequences have an impact on the readability of inputs and the generated samples are offensive in nature.

In terms of evaluation, there exists two main scarcities:

(1) Most of researches evaluate their performances of adversarial attacks by success rate or accuracy. Only a few works [22], [20] take speed, scale, and efficiency into consideration, although they only list the time spent of the attacks. A question of whether there is a relationship among the scale of dataset, time consumed, and success rate of adversarial attacks is still unknown. If there exists such a relationship, the trade-off of these three aspects may be a research point in the future work. Currently, a related study [170] of speed in adversarial examples have been conducted in image domain. Besides, the experimental results on different datasets are various when the attack method is same. Whether the type of data may affect adversarial attacks is worth pondering.

(2) Various methods have been proposed to study adversarial attacks and defenses in texts, but there is not a benchmark for them. Researchers use different datasets (in Section 2.7) in their works, making it difficult to compare the advantages and disadvantages of these methods. Meanwhile, it also affects the selection of metrics. Currently, there do not have an exact statement that which metric measure is better in a situation and why it is more useful than others. Some comparisons have been made in Textbugger [20] with several metrics. The best one in this work may be only suitable for it, but ineffective in other works.

In terms of quick start in this aspect, it is short of an open-source toolbox (*e.g.*, AdvBox [171] and cleverhans [172] in image domain) for the research on adversarial texts. The toolboxes in image domain integrate existing representative methods of generating adversarial images. People can easily do some further studies by them, which reduce time consumption for repetition and promote the development in this field. Compared with those in image domain, there is only one visual analytics framework proposed by Laughlin et al. [173] in texts. But it is lack of diverse attack and defense methods, which can be integrated to make it more powerful.

8 CONCLUSION AND FUTURE DIRECTIONS

This paper presents a survey about adversarial attacks and defenses on DNNs in texts. Although DNNs have a high performance on a wide variety of NLP tasks, they are inherently vulnerable to adversarial examples. Hence, people pay a high attention to the security problem caused by adversarial examples. We integrates almost existing adversarial attacks and defenses focusing on recent works in texts. The threat of adversarial attacks is real, but defense methods are few. Most existing works have their limitations like application scenes and constraint conditions. More attention should be paid to the problem caused by adversarial example, which remains an open issue for designing considerably robust models against adversarial attacks.

In the future, researches may study adversarial examples from the followings: As an attacker, designing universal perturbations can be taken into consideration as it works in image domain [74]. The universal adversarial perturbations on any text are able to make a model produce incorrect judgment with a high probability. Moreover, more wonderful universal perturbations can fool multi-models or any model on any text. On the other hand, the work of enhancing the transferability is meaningful in more practical black-box attacks. Besides, the combination of optimization-based and transferability-based methods is another viable way like the work in [174]. On the contrary, defenders prefer to completely revamp this vulnerability in DNNs, but it is no less difficult than redesigning a network. Both of them are long and arduous tasks with the common efforts of many people. At the moment, the defender can draw on methods from image area to text for improving the robustness of DNNs, *e.g.*, adversarial training [162], adding extra layer [175], optimizing cross-entropy function [176], [177], or weakening the transferability of adversarial examples.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems*, vol. 1, 2012, pp. 1097–1105.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems*, vol. 1, 2015, pp. 91–99.

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. rahman Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[4] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," 2016, arXiv preprint arXiv:1609.03499.

[5] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2014, p. 31043112.

[6] H. Xu, M. Dong, D. Zhu, A. Kotov, A. I. Carcone, and S. Naar-King, "Text classification with topic-based word embedding and convolutional neural networks," in *BCB '16 Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2016, pp. 88–97.

[7] D. Tao, X. Li, S. Maybank, and X. Wu, "Human carrying status in visual surveillance," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1670–1677.

[8] F. Zhang, P. P. K. Chan, B. Biggio, D. S. Yeung, and F. Roli, "Adversarial feature selection against evasion attacks," *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 766–777, 2015.

[9] J. Saxe and K. Berlin, "Deep neural network based malware detection using two dimensional binary program features," in *Proceedings of the 10th International Conference on Malicious and Unwanted Software (MALWARE)*. IEEE, 2015.

[10] Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, "Droid-sec: deep learning in android malware detection," in *Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM)*, 2014, pp. 371–372.

[11] K. D. Julian, J. Lopez, J. S. Brush, M. P. Owen, and M. J. Kochenderfer, "Policy compression for aircraft collision avoidance systems," in *Proceedings of the 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 2016, pp. 1–10.

[12] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proceedings of the International Conference on Learning Representations*, 2014.

[13] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning models," in *the 31th IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[14] C. Xie, J. Wang, Z. Zhangm, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *IEEE International Conference on Computer Vision*, 2017, pp. 1378–1387.

[15] R. Taori, A. Kamsetty, B. Chu, and N. Vemuri, "Targeted adversarial examples for black box audio systems," 2018, arXiv preprint arXiv: 1805.07820.

[16] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *IEEE Security and Privacy Workshops*. IEEE, 2018.

[17] H. Yakura and J. Sakuma, "Robust audio adversarial example for a physical attack," 2018, arXiv preprint arXiv: 1810.11793.

[18] X. Liu, Y. Lin, H. Li, and J. Zhang, "Adversarial examples: Attacks on machine learning-based malware visualization detection methods," 2018, arXiv preprint arXiv:1808.01546.

[19] W. He and Y. Tan, "Generating adversarial malware examples for black-box attacks based on gan," 2017, arXiv preprint arXiv: 1702.05983.

[20] J. Li, S. Ji, T. Du, B. Li, and T. Wang, "Textbugger: Generating adversarial text against real-world applications," in *the Network and Distributed System Security Symposium*, 2019.

[21] N. Papernot, P. McDaniel, A. Swami, and R. Harang, "Crafting adversarial input sequences for recurrent neural networks," in *IEEE Military Communications Conference*, 2016, p. 4954.

[22] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, "Black-box generation of adversarial text sequences to evade deep learning classifiers," in *IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018.

[23] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang, "Generating natural language adversarial examples," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[24] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, "Hotflip: White-box adversarial examples for text classificationd," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. ACL, 2018, pp. 31–36.

[25] M. Sato, J. Suzuki, H. Shindo, and Y. Matsumoto, "Interpretable adversarial perturbation in input embedding space for text," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

[26] B. Liang, H. Li, M. Su, P. Bian, X. Li, and W. Shi, "Deep text classification can be fooled," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2018, p. 42084215.

[27] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," in *Proceedings of the 2017 conference on empirical methods in natural language processing (EMNLP)*, 2017, p. 20212031.

[28] P. Minervini and S. Riedel, "Adversarially regularising neural nli models to integrate logical background knowledge," in *the SIGNLL Conference on Computational Natural Language Learning*, 2018.

[29] Y. Belinkov and Y. Bisk, "Synthetic and natural noise both break neural machine translation," in *Proceedings of the International Conference on Learning Representations*, 2018.

[30] J. Ebrahimi, D. Lowd, and D. Dou, "On adversarial examples for character-level neural machine translation," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018.

[31] S. Samanta and S. Mehta, "Towards crafting text adversarial samples," 2017, arXiv preprint arXiv:1707.02812.

[32] Z. Gong, W. Wang, B. Li, D. Song, and W.-S. Ku, "Adversarial texts with gradient methods," 2018, arXiv preprint arXiv:1801.07175.

[33] Y. Gil, Y. Chai, O. Gorodissky, and J. Berant, "White-to-black: Efficient distillation of black-box adversarial attacks," in *Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

[34] M. Glockner, V. Shwartz, and Y. Goldberg, "Breaking nli systems with sentences that require simple lexical inferences," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 650–655.

[35] P. K. Mudrakarta, A. Taly, M. Sundararajan, and K. Dhamdhere, "Did the model understand the question?" in *the 56th Annual Meeting of the Association for Computational Linguistics*, 2018.

[36] T. Niven and H.-Y. Kao, "Probing neural network comprehension of natural language arguments," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. ACL, 2019, pp. 4658–4664.

[37] P. Minervini, T. Demeester, T. Rocktschel, and S. Riedel, "Adversarial sets for regularising neural link predictors," in *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

[38] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[39] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive language detection in online user content," in *WWW '16 Proceedings of the 25th International Conference on World Wide Web*, 2016, pp. 145–153.

[40] S. Rayana and L. Akoglu, "Collective opinion spam detection: Bridging review networks and metadata," in *Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 2015.

[41] X. Xiao, B. Yang, and Z. Kang, "A gradient tree boosting based approach to rumor detecting on sina weibo," 2018, arXiv preprint arXiv: 1806.06326.

[42] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 2017, p. 506519.

[43] S. Shen, R. Furuta, T. Yamasaki, and K. Aizawa, "Fooling neural networks in face attractiveness evaluation: Adversarial examples with high attractiveness score but low subjective score," in *IEEE Third International Conference on Multimedia Big Data*. IEEE, 2017.

[44] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proceedings of the International Conference on Learning Representations*, 2015.

[45] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410 – 14 430, 2018.

[46] A. P. Norton and Y. Qi, "Adversarial-playground: A visualization suite showing how adversarial examples fool deep learning," in *IEEE Symposium on Visualization for Cyber Security*, 2017, pp. 1–4.

[47] N. Carlini, "A complete list of all (arxiv) adversarial example papers," 2019, https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html.

[48] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

[49] J. Gilmer, R. P. Adams, I. Goodfellow, D. Andersen, and G. E. Dahl, "Motivating the rules of the game for adversarial example research," *arXiv preprint arXiv:1807.06732*, 2018.

[50] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, and V. C. M. Leung, "A survey on security threats and defensive techniques of machine learning: A data driven view," *IEEE Access*, vol. 6, pp. 12 103 – 12 117, 2018.

[51] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Transactions on Neural Networks and learning systems*, pp. 1–20, 2019.

[52] J. Zhang and X. Jiang, "Adversarial examples: Opportunities and challenges," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–16, 2019.

[53] H. Xu, Y. Ma, H. Liu, D. Deb, H. Liu, J. Tang, and A. Jain, "Adversarial attacks and defenses in images, graphs and text: A review," *arXiv preprint arXiv:1909.08072*, 2019.

[54] Y. Belinkov and J. Glass, "Analysis methods in neural language processing: A survey," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 49–72, 2019.

[55] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li, "Adversarial attacks on deep learning models in natural language processing: A survey," *arXiv preprint arXiv:1901.06796*, 2019.

[56] V. Sze, Y.-H. Chen, T.-J. Yang, and J. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295 – 2329, 2017.

[57] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[58] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in *Proceedings of Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*, 2014.

[59] I. Iacobacci, M. T. Pilehvar, and R. Navigli, "Embeddings for word sense disambiguation: An evaluation study," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. 1. ACL, 2016, pp. 897–907.

[60] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[61] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of the International Conference on Learning Representations*, 2013.

[62] S. Cai, A. Palazoglu, L. Zhang, and J. Hu, "Process alarm prediction using deep learning and word embedding methods," *ISA Transactions*, vol. 85, pp. 274–283, 2019.

[63] M. Naili, A. H. Chaibi, and H. H. B. Ghezala, "Comparative study of word embedding methods in topic segmentation," *Procedia Computer Science*, vol. 112, no. C, pp. 340–349, 2017.

[64] X. Ling, S. Ji, J. Zou, J. Wang, C. Wu, B. Li, and T. Wang, "Deepsec: A uniform platform for security analysis of deep learning model," in *IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 381–398.

[65] D. Heaven, "Why deep-learning ais are so easy to fool," *Nature*, vol. 574, pp. 163–166, 2019.

[66] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning*, 2010, pp. 807–814.

[67] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," in *Proceedings of the International Conference on Learning Representations*, 2016.

[68] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," *arXiv preprint arXiv:1905.02175*, 2019.

[69] M. Naseer, S. H. Khan, S. Rahman, and F. Porikli, "Distorting neural representations to generate highly transferable adversarial examples," 2018, arXiv preprint arXiv: 1811.09020.

[70] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin, "On evaluating adversarial robustness," 2019, arXiv preprint arXiv: 1902.06705.

[71] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *IEEE European Symposium on Security and Privacy*. IEEE, 2016.

[72] J. Su, D. V. Vargas, and S. Kouichi, "One pixel attack for fooling deep neural networks," 2017, arXiv preprint arXiv:1710.08864.

[73] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symposium on Security and Privacy (SP)*, 2017.

[74] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[75] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2574–2582.

[76] M. Cisse, Y. Adi, N. Neverova, and J. Keshet, "Houdini: Fooling deep structured prediction models," 2017, arXiv preprint arXiv: 1707.05373.

[77] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Proceedings of the International Conference on Learning Representations*, 2017.

[78] S. Sarkar, A. Bansal, U. Mahbub, and R. Chellappa, "Upset and angri: Breaking high performance image classifiers," 2017, arXiv preprint arXiv: 1707.01159.

[79] S. Baluja and I. Fischer, "Adversarial transformation networks: Learning to generate adversarial examples," 2017, arXiv preprint arXiv: 1703.09387.

[80] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[81] B. Luo, Y. Liu, L. Wei, and Q. Xu, "Towards imperceptible and robust adversarial example attacks against neural networks," in *Proceedings of Association for the Advancement of Artificial Intelligence*, 2018.

[82] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger, "From word embeddings to document distances," in *Proceedings of the International Conference on International Conference on Machine Learning*, 2015, pp. 957–966.

[83] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *ICCV '98 Proceedings of the Sixth International Conference on Computer Vision*. IEEE, 1998, pp. 59–66.

[84] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 163, no. 4, pp. 845–848, 1965.

[85] P. Michel, X. Li, G. Neubig, and J. M. Pino, "On evaluation of adversarial perturbations for sequence-to-sequence models," in *Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

[86] CloudFlower, "Twitter gender classification dataset," 2013. [Online]. Available: https://www.kaggle.com/crowdflower/twitter-user-gender-classification

[87] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," in *Proceedings of the North American Chapter of the Association for Computational Linguistics Human Language Technologies (NAACL HLT)*, 2015.

[88] M. Iyyer, J. Wieting, K. Gimpel, and L. Zettlemoyer, "Adversarial example generation with syntactically controlled paraphrase networks," in *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2018.

[89] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," in *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP)*, 2016.

[90] A. Williams, N. Nangia, and S. R. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2018.

[91] M. Blohm, G. Jagfeld, E. Sood, X. Yu, and N. T. Vu, "Comparing attention-based convolutional and recurrent neural networks: Success and limitations in machine reading comprehension," in *Proceedings of the SIGNLL Conference on Computational Natural Language Learning*, 2018.

[92] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler, "Movieqa: Understanding stories in movies through question-answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[93] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

[94] P. J.Werbos, "Generalization of backpropagation with application to a recurrent gas market model," *Neural Networks*, vol. 1, no. 4, pp. 339–356, 1988.

[95] W. Wang, R. Wang, L. Wang, and B. Tang, "Adversarial examples generation approach for tendency classification on chinese texts," *Ruan Jian Xue Bao/Journal of Software (in Chinese)*, pp. 1–14, 2019.

[96] M. Atallah, V. Raskin, M. Crogan, C. Hempelmann, F. Kerschbaum, D. Mohamed, and S. Naik, "Natural language watermarking: Design, analysis, and a proof-of-concept implementation," *In Information Hiding*, p. 185200, 2001.

[97] M. Sutton, A. Greene, and P. Amini, *Fuzzing: Brute Force Vulnerability Discovery*. Addison-Wesley Professional, 2007.

[98] H. Hosseini, S. Kannan, B. Zhang, and R. Poovendran, "Deceiving google's perspective api built for detecting toxic comments," in *Proceedings of the Bright and Dark Sides of Computer Vision: Challenges and Opportunities for Privacy and Security workshop*, 2017.

[99] E. J. Anderson and M. C. Ferris, "Genetic algorithms for combinatorial optimization: the assemble line balancing problem," *ORSA Journal on Computing*, vol. 6, no. 2, p. 161173, 1994.

[100] H. Mhlenbein, "Parallel genetic algorithms, population genetics and combinatorial optimization," in *Proceedings of the third international conference on Genetic algorithms*, 1989, pp. 416–421.

[101] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, p. 15321543.

[102] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson, "One billion word benchmark for measuring progress in statistical language modeling," *Computer Science*, 2013.

[103] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, p. 649657.

[104] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014.

[105] Y. Kim, Y. Jernite, D. Sontag, and A. der M Rush, "Character-aware neural language models," in *Proceedings of AAAI*, 2016.

[106] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *Proceedings of the International Conference on Learning Representations*, 2017.

[107] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," 2017, arXiv preprint arXiv: 1710.06081.

[108] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[109] R. Pascanu, T. Mikolov, and Y. Bengio, "Understanding the exploding gradient problem," *arXiv preprint arXiv: 1211.5063*, 2012.

[110] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," 2018, arXiv preprint arXiv: 1802.00420.

[111] R. Graham, "The significance of letter position in word recognition," *IEEE Aerospace & Electronic Systems Magazine*, vol. 22, no. 1, pp. 26–27, 2007.

[112] J. Lee, K. Cho, and T. Hofmann, "Fully character-level neural machine translation without explicit segmentation," in *Transactions of the Association for Computational Linguistics (TACL)*, 2017.

[113] R. Sennrich, O. Firat, K. Cho, A. Birch, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. Laubli, A. V. M. Barone, J. Mokry, and M. Ndejde, "Nematus: a toolkit for neural machine translation," in *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017, p. 6568.

[114] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the International Conference on Learning Representations*, 2014.

[115] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of the Association for Computational Linguistics*, 2017.

[116] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," in *Proceedings of the International Conference on Learning Representations*, 2017.

[117] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," in *Proceedings of Network and Distributed Systems Security Symposium (NDSS)*, 2018.

[118] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," 2017, arXiv preprint arXiv: 1703.00410.

[119] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel, "On the (statistical) detection of adversarial examples," 2017, arXiv preprint arXiv: 1702.06280.

[120] K. Roth, Y. Kilcher, and T. Hofmann, "The odds are odd: A statistical test for detecting adversarial examples," 2019, arXiv preprint arXiv: 1902.04818.

[121] D. Pruthi, B. Dhingra, and Z. C. Lipton, "Combating adversarial misspellings with robust word recognition," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[122] X. Wang, H. Jin, and K. He, "Natural language adversarial attacks and defenses in word level," *arXiv preprint arXiv:1909.06723*, 2019.

[123] C. K. Mummadi, T. Brox, and J. H. Metzen, "Defending against universal perturbations with shared adversarial training," 2018, arXiv preprint arXiv: 1812.03705.

[124] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," in *Proceedings of the International Conference on Learning Representations*, 2017.

[125] Y. Cheng, L. Jiang, and W. Macherey, "Robust neural machine translation with doubly adversarial inputs," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. ACL, 2019.

[126] Y. Wang and M. Bansal, "Robust machine comprehension models via adversarial training," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. ACL, 2018, p. 575581.

[127] F. Tram'er, A. Kurakin, N. Papernot, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," in *Proceedings of the International Conference on Learning Representations*, 2018.

[128] G. Goren, O. Kurland, M. Tennenholtz, and F. Raiber, "Ranking robustness under adversarial document manipulations," in *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2018.

[129] T.-Y. Liu, *Learning to Rank for Information Retrieval*. Springer, 2011.

[130] A. Fawzi, O. Fawzi, and P. Frossard, "Analysis of classifiers' robustness to adversarial perturbations," *Machine Learning*, vol. 107, no. 3, p. 481508, 2018.

[131] G. S.Shieh, "A weighted kendalls tau statistic," *Statistics & Probability Letters*, vol. 39, no. 1, pp. 17–24, 1998.

[132] T. Joachims, "Training linear svms in linear time," in *KDD'06 Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 217–226.

[133] Q. Wu, C. J. C. Burges, K. M. Svore, and J. Gao, "Adapting boosting for information retrieval measures," *Information Retrieval*, vol. 13, no. 3, p. 254270, 2010.

[134] A. H. Li and A. Sethy, "Knowledge enhanced attention for robust natural language inference," *arXiv preprint arXiv:1909.00102*, 2019.

[135] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *In Advances in neural information processing systems*, p. 59986008, 2017.

[136] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the Association for Computing Machinery*, vol. 38, no. 11, pp. 39–41, 1995.

[137] L. Ma, F. Juefei-Xu, M. Xue, Q. Hu, S. Chen, B. Li, Y. Liu, J. Zhao, J. Yin, and S. See, "Secure deep learning engineering: A software quality assurance perspective," 2018, arXiv preprint arXiv: 1810.04538.

[138] Y. Sun, M. Wu, W. Ruan, X. Huang, M. Kwiatkowska, and D. Kroening, "Concolic testing for deep neural networks," in *Proceedings of 33rd ACM/IEEE International Conference on Automated Software Engineering*, 2018.

[139] K. Pei, Y. Cao, J. Yang, and S. Jana, "Deepxplore: Automated whitebox testing of deep learning systems," in *Proceedings of ACM Symposium on Operating Systems Principles*. ACM, 2017.

[140] L. Ma, F. Juefei-Xu, J. Sun, C. Chen, T. Su, F. Zhang, M. Xue, B. Li, L. Li, Y. Liu, J. Zhao, and Y. Wang, "Deepgauge: Comprehensive and multi-granularity testing criteria for gauging the robustness of deep learning systems," in *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, 2018.

[141] Y. Tian, K. Pei, S. Jana, and B. Ray, "Deeptest: Automated testing of deep-neural-network-driven autonomous cars," in *Proceedings of the 40th International Conference on Software Engineering*. ACM, 2018, pp. 303–314.

[142] Y. Sun, X. Huang, and D. Kroening, "Testing deep neural networks," 2018, arXiv preprint arXiv: 1803.04792.

[143] M. Wicker, X. Huang, and M. Kwiatkowska, "Feature-guided black-box safety testing of deep neural networks," in *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 2018, pp. 428–426.

[144] G. M. J.-B. Chaslot, M. H. M. Winands, H. J. van den Herik, J. W. UITERWIJK, and B. Bouzy, "Progressive strategies for monte-carlo tree search," *New Mathematics and Natural Computation*, vol. 4, no. 3, pp. 343–359, 2008.

[145] C. Budnik, M. Gario, G. Markov, and Z. Wang, "Guided test case generation through ai enabled output space exploration," in *Proceedings of the 13th International Workshop on Automation of Software Test*, 2018.

[146] J. Kim, R. Feldt, and S. Yoo, "Guiding deep learning system testing using surprise adequacy," 2018, arXiv preprint arXiv: 1808.08444.

[147] I. Goodfellow and N. Papernot, "The challenge of verification and testing of machine learning," 2017, http://www.cleverhans.io/.

[148] G. Katz, C. Barrett, D. Dill, K. Julian, and M. Kochenderfer, "Reluplex: An efficient smt solver for verifying deep neural networks," in *Proceedings of the International Conference on Computer Aided Verification*, 2017, pp. 97–117.

[149] L. D. Moura and N. Bjrner, "Satisfiability modulo theories: introduction and applications," *Communications of the ACM*, vol. 54, no. 9, pp. 69–77, 2011.

[150] V. Tjeng, K. Xiao, and R. Tedrake, "Evaluating robustness of neural networks with mixed integer programming," 2017, arXiv preprint arXiv: 1711.07356.

[151] J. P. Vielma, "Mixed integer linear programming formulation techniques," *SIAM Review*, vol. 57, no. 1, pp. 3–57, 2015.

[152] N. Carlini, G. Katz, C. Barrett, and D. L. Dill, "Ground-truth adversarial examples," 2017, arXiv preprint arXiv: 1709.10207.

[153] O. Bastani, Y. Ioannou, L. Lampropoulos, D. Vytiniotis, A. Nori, and A. Criminisi, "Measuring neural net robustness with constraints," in *Advances in neural information processing systems*, 2016, pp. 2613–2621.

[154] S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana, "Formal security analysis of neural networks using symbolic intervals," 2018, arXiv preprint arXiv: 1804.10829.

[155] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, 2009.

[156] X. Huang, M. Kwiatkowska, S. Wang, and M. Wu, "Safety verification of deep neural networks," in *Proceedings of the International Conference on Computer Aided Verification*, 2017, pp. 3–29.

[157] N. Narodytska, S. P. Kasiviswanathan, L. Ryzhyk, M. Sagiv, and T. Walsh, "Verifying properties of binarized deep neural networks," 2017, arXiv preprint arXiv: 1709.06662.

[158] I. Hubara, D. Soudry, and R. E. Yaniv, "Binarized neural networks," in *Advances in neural information processing systems*, 2016, pp. 4107–4115.

[159] E. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith, "Counterexample-guided abstraction refinement," in *Proceedings of International Conference on Computer Aided Verification*, 2000, pp. 154–169.

[160] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[161] T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev, "Ai2: Safety and robustness certification of neural networks with abstract interpretation," in *IEEE Symposium on Security and Privacy (SP)*, 2018.

[162] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proceedings of the International Conference on Learning Representations*, 2018.

[163] T.-W. Weng, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, D. Boning, I. S. Dhillon, and L. Daniel, "Towards fast computation of certified robustness for relu networks," 2018, arXiv preprint arXiv: 1804.09699.

[164] J. E. Hummel and K. J. Holyoak, "Relational reasoning in a neurally plausible cognitive architecture: An overview of the lisa project," *Current Directions in Psychological Science*, vol. 14, no. 3, p. 153157, 2005.

[165] M. Cheng, J. Yi, H. Zhang, P.-Y. Chen, and C.-J. Hsieh, "Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples," *arXiv preprint arXiv:1803.01128*, 2018.

[166] R. Jia, A. Raghunathan, K. Göksel, and P. Liang, "Certified robustness to adversarial word substitutions," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[167] K. Dvijotham, S. Gowal, R. Stanforth, R. Arandjelovic, B. O'Donoghue, J. Uesato, and P. Kohli, "Training verified learners with learned verifiers," *arXiv preprint arXiv:1805.10265*, 2018.

[168] L. Wu, Z. Zhu, C. Tai, and W. E, "Understanding and enhancing the transferability of adversarial examples," *arXiv preprint arXiv: 1802.09707*, 2018.

[169] E. Wallace, S. Feng, N. Kandpal, M. Gardner, and S. Singh, "Universal adversarial triggers for attacking and analyzing nlp," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

[170] H. Zhang, Y. Avrithis, T. Furon, and L. Amsaleg, "Walking on the edge: Fast, low-distortion adversarial examples," *arXiv preprint arXiv: 1912.02153*, 2019.

[171] D. Goodman, "Advbox:a toolbox to generate adversarial examples that fool neural networks," 2019. [Online]. Available: https://github.com/baidu/AdvBox

[172] N. Papernot, I. Goodfellow, R. Sheatsley, R. Feinman, and P. McDaniel, "cleverhans v1.0.0: an adversarial machine learning library," *arXiv preprint arXiv:1610.00768*, 2016.

[173] B. Laughlin, C. Collins, K. Sankaranarayanan, and K. El-Khatib, "A visual analytics framework for adversarial text generation," *arXiv preprint arXiv:1909.11202*, 2019.

[174] F. Suya, J. Chi, D. Evans, and Y. Tian, "Hybrid batch attacks: Finding black-box adversarial examples with limited queries," in *Proceedings of the 29th USENIX Security Symposium*, 2020.

[175] F. Menet, P. Berthier, J. M. Fernandez, and M. Gagnon, "Spartan networks: Self-feature-squeezing neural networks for increased robustness in adversarial settings," in *CCS '18 Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 2246–2248.

[176] S. Kariyappa and M. K. Qureshi, "Improving adversarial robustness of ensembles with diversity training," 2019, arXiv preprint arXiv: 1901.09981.

[177] K. Nar, O. Ocal, S. S. Sastry, and K. Ramchandran, "Cross-entropy loss and low-rank features have responsibility for adversarial examples," 2019, arXiv preprint arXiv: 1901.08360.