(will be inserted by the editor)

Adversarial Robustness of Open-source Text Classification Models and Fine-Tuning Chains

- An Empirical Study of Text Classification Models on Hugging Face Hub

Hao $\operatorname{Qin}^* \cdot \operatorname{Mingyang} \operatorname{Li}^* \cdot \operatorname{Junjie} \operatorname{Wang} \cdot \operatorname{Qing} \operatorname{Wang}$

Received: date / Accepted: date

Abstract Context: With the advancement of artificial intelligence (AI) technology and applications, numerous AI models have been developed, leading to the emergence of open-source model hosting platforms like Hugging Face (HF). Thanks to these platforms, individuals can directly download and use models, as well as fine-tune them to construct more domain-specific models. However, just like traditional software supply chains face security risks, AI models and fine-tuning chains also encounter new security risks, such as adversarial attacks. Therefore, the adversarial robustness of these models has garnered attention, potentially influencing people's choices regarding open-source models.

Objective: This paper aims to explore the adversarial robustness of open-source AI models and their chains formed by the upstream-downstream relationships via fine-tuning to provide insights into the potential adversarial risks.

Method: We collect text classification models on HF and construct the fine-tuning chains. Then, we conduct an empirical analysis of model reuse and associated robustness risks under existing adversarial attacks from two aspects, i.e., models and their fine-tuning chains.

Results: Despite the models' widespread downloading and reuse, they are generally susceptible to adversarial attack risks, with an average of 52.70% attack success rate. Moreover, fine-tuning typically exacerbates this risk, resulting in an average 12.60% increase in attack success rates. We also delve into the influence of factors such as attack techniques, datasets, and model architectures on the success rate, as well as the transitivity along the model chains.

Hao Qin*, Mingyang Li*, Junjie Wang, Qing Wang are with State Key Laboratory of Intelligent Game, Institute of Software Chinese Academy of Sciences, and University of Chinese Academy of Sciences, Beijing, China.

E-mail: qinhao 22@ mails.ucas.ac.cn; mingyang 2017@ iscas.ac.cn; junjie@ iscas.ac.cn; wq@ itechs.iscas.ac.cn

^{*}Both authors contributed equally to this work.

Conclusions: The results indicate the poor robustness of text classification models on the popular model hosting platform like HF, and raise the awareness of both researchers and model users pay more attention to the security risks of open-source AI models. The analysis also provide valuable insights about the adversarial attack manifestation in both single models and fine-tuning chains, potentially facilitate the risk mitigation and defense techniques.

Keywords Adversarial Attack, Adversarial Robustness, Text Classification Model, Fine-tuning Chain, Hugging Face

1 Introduction

In recent years, the rapid development of deep learning technology has significantly propelled the advancement of artificial intelligence (AI), seeing widespread application and garnering immense attention in various fields such as computer vision and natural language processing [27, 47]. Concomitant with the AI technique, numerous AI models have been widely applied and provide more intelligent and efficient solutions for real-life scenarios [38, 43]. However, it is the consensus that building AI models is expensive as the training process typically requires a large number of training data and consumes numerous computing resources, which has been the key challenge in hindering their broad applications [34, 2].

Given the exorbitant costs, increasing users are building the target models by reusing open-source AI models [33, 50]. With the appropriate open-source models, users can apply them directly or fine-tune them for downstream applications. Significantly, the fine-tuning technique has gained increasing attention to achieve efficient and effective model building. Instead of training models from scratch, the fine-tuning technique starts with powerful base models, e.g., BERT, GPT, and MAR, pre-trained on extensive corpora [13]. In recent years, fine-tuning from an open-source base model has become the most mainstream paradigm for building specialized models and has seen widespread application across multiple fields [23]. Under the paradigm, an implicit associative relationship forms accordingly, with each base model as an upstream node and the fine-tuned model as the downstream node.

The widespread adoption of fine-tuning paradigm and the increasing demand for AI models have led to the emergence of open-source model hosting platforms like Hugging Face (HF). It is a rapidly growing hub where stakeholders can release, iterate, and share their open-source AI models for model reuse. By April 2024, over 5,000 organizations and individuals had contributed 631,866 models and 139,620 associated datasets, including Meta, Google, Microsoft, and Amazon [5]. The open-source models in the HF hubs cover more than forty tasks, including text classification, text generation, and so on. On HF, besides the general pre-trained models (e.g., BERT [11] and ResNet[14]), there also exists a multitude of diverse models that are fine-tuned from other AI models. Under the fine-tuning paradigm, the upstream and downstream

relationships intertwine, thereby forming abundant implicit model chains (as illustrated in Figure 1, we name them *fine-tuning chains*).

However, just like the traditional software supply chains face security risks [40], the AI models and fine-tuning chains also encounter new security risks, such as adversarial attacks. Adversarial robustness, an indicator of the resistance ability to human interference, is a critical factor for ensuring the credibility of AI models in application scenarios [7, 25]. To assess the adversarial robustness, stakeholders typically employ various techniques to generate adversarial samples for attacking, observing that the model could still make the correct predictions [6]. Low adversarial robustness implies that the models are more susceptible to external disturbances, potentially leading to severe erroneous application decisions. Therefore, it is essential to know the adversarial robustness of the open-source models so that users can be aware of the relevant risks when reusing them.



Fig. 1 Illustrative examples of fine-tuning chain

Unfortunately, existing studies touched upon aspects such as the reusability of pre-trained models on HF [42, 5], the carbon emissions during model training on HF [4], but none of them concern about the security aspects of open-source AI models, particularly in terms of adversarial robustness. More than that, for the inherent fine-tuning chains buried in model hubs like HF, stakeholders know little about their reuse popularity and the adversarial risks in different reuse scenarios.

To fill this gap, we choose text classification models on HF as subjects because of their popularity and well-studied in terms of adversarial attacks. We collect 45,688 text classification models from HF and automatically construct their fine-tuning chains with the information extraction technique. After that, we conduct an empirical analysis of the current situation of model reuse and associated adversarial risks from two aspects, i.e., open-source models themselves and their implicit fine-tuning chains. Accordingly, we answer three research questions as follows.

RQ1: General status of open-source text classification models and fine-tuning chains.

Hao Qin^{*} et al.

- RQ1.1: How popular are the open-source text classification models on HF?

- RQ1.2: How prevalent is the model reuse within HF?
- RQ1.1 aims to explore the popularity of open-source models through examining their download volumes, while RQ1.2 explores their reuse situation formed by the upstream-downstream relationships via fine-tuning.

RQ2. Adversarial robustness of open-source text classification models.

- RQ2.1: How robust are the widely-used text classification models under existing adversarial attacks?
- RQ2.2: Are there any obvious robustness differences for these text classification models under different attack techniques and datasets?

This RQ attempts to reveal the model's adversarial robustness under existing adversarial attacks, providing empirical knowledge about the current situation of dominating open-source models for security-sensitive users. Furthermore, as two key elements of adversarial attacks, we evaluate the adversarial robustness of open-source models from two perspectives, i.e., attack techniques and datasets.

RQ3: Adversarial robustness of open-source fine-tuning chains.

- RQ3.1: Are text classification models more robust or more vulnerable to the adversarial attacks after fine-tuning?
- RQ3.2: Can the vulnerability transfer along the model chains on HF?

This RQ aims to investigate the impact of the fine-tuning process on the adversarial robustness of text classification models on HF, respectively from a general perspective (RQ3.1) and more fine-grained perspective of attack samples (RQ3.2), guiding users on the potential threats when reusing open-source models via fine-tuning.

Although 45,688 open source text classification models provide a rich research basis for this study, it is not feasible to conduct a comprehensive experimental analysis due to resource and time constraints. Therefore, this study established a set of strict selection criteria and selected 20 representative models and 10 model chains from a large number of models for in-depth empirical research. We experiment with six widely-used and state-of-the-art adversarial attack techniques, and four widely-used dataset for conducting attack. The results show that the open-source text classification models are quite popular in terms of download volume, and prevalent in model reuse within HF. Despite of that, they are generally susceptible to adversarial attack risks, within an average of 52.70% attack success rate. Furthermore, under different model architectures, the adversarial robustness exhibits different trend after fine-tuning: with the dominating BERT architecture, fine-tuning tends to degrade the robustness of downstream models, resulting in an average of 12.60% increase in attack success rate; conversely, Electra architecture exhibits an inverse trend. In addition, the vulnerabilities exhibit a certain degree of transitivity during the fine-tuning process, i.e., 71.3% samples that successfully attack the upstream model remain effective against the downstream model.

We also explore the implications of our experimental findings, which include providing guidance on selecting suitable methods for reusing models in security-sensitive domains, uncovering potential for more targeted attacks using vulnerable samples from upstream models, drawing inspiration from the Electra architecture to bolster robustness, and more. This paper highlights the inadequate robustness of text classification models hosted on popular platforms like HF, emphasizing the need for both researchers and model users to pay more attention to the security vulnerabilities in open-source AI models. Additionally, our analysis offers valuable insights into the occurrence of adversarial attacks across both individual models and fine-tuning chains, potentially aiding in the development of strategies to mitigate risks and implement defense mechanisms.

The key contributions of this study are listed as follows.

- To our knowledge, we are the first to investigate the adversarial robustness of the open-source models on HF, particularly the adversarial robustness of the fine-tuning chains.
- Our study reveals the inherent robustness risks of open-source text classification models on HF and their transmissibility in the fine-tuning chains. It can provide insights for security-sensitive users to understand the potential risks under different model reuse scenarios.
- We make the collected open-source models and constructed fine-tuning chains publicly available¹, which can be used for replication or reused in future studies.

The structure of this paper is as follows: Section 2 introduces the background, Section 3 presents how we collect the models and construct fine-tuning chains. Section 4 to 6 present the experimental design and result analyses for three research questions respectively. Section 7 offers implications of this study, and Section 8 uncovers the threats to validity. Section 9 explores related works, while Section 10 concludes this paper.

2 Background

2.1 Hugging Face (HF)

Hugging Face (HF) is a company that specializes in natural language processing, an significant branch of AI, and has become widely recognized for its contributions to the field through its open-source projects and community-driven approach. The company is initially and best known for its development of the "Transformers" library, which provides a vast array of pre-trained models foundational for various natural language processing tasks, such as text classification, question answering, and machine translation. This library has

 $^{^{1}\ \}rm https://github.com/Xcasdfas/Adversarial-Robustness-of-Open-source-AI-Models-and-Fine-Tuning-Chains$

garnered a large community of users and contributors, its emergence has simplified access to and deployment of advanced NLP and ML models. Moreover, through its open collaborative community-driven model, it has facilitated the sharing of knowledge and technology on a global scale, making it one of the most influential platforms in advancing natural language processing technology.

Model Hub is the core module for HF that provides a collaborative environment through its online platform. Researchers, developers, and organizations can upload and share their pre-trained models and datasets, making them accessible to the broader community. The Hub supports various machine learning frameworks and provides tools for version control, model hosting, and easy integration into existing projects. This collaborative environment fosters innovation and accelerates the development of new AI applications. By April 2024, over 5,000 organizations and individuals had contributed 631,866 models and 139,620 associated datasets, covering more than forty tasks including text classification, text generation, and so on.

2.2 Adversarial Attacks And Adversarial Robustness Assessment

Adversarial attacks against AI models represent a critical and burgeoning area of study within the field of AI security. These attacks involve the crafting of input data that is slightly altered from the original samples but done so in a way that is imperceptible or minimally perceptible to humans. Despite these minor alterations, such adversarially modified inputs can cause an AI model to make incorrect predictions or classifications with high confidence. The susceptibility of AI models to these insidious attacks highlights significant vulnerabilities in their ability to generalize from the training data to real-world scenarios. This phenomenon raises profound security concerns, particularly in applications where the integrity and reliability of model predictions are paramount, For example, in the field of autonomous driving [19], adversarial attacks can cause the vehicle perception system to misrecognize traffic signs, leading to incorrect driving decisions and increasing the risk of traffic accidents. In the context of fake news detection [28], adversarial attacks can deceive detection systems, preventing them from timely identifying and filtering false information, thereby accelerating the spread of misinformation, misleading public opinion, and potentially destabilizing society. In medical auxiliary diagnosiss [48], adversarial attacks can cause AI diagnostic systems to misclassify or predict patient data, resulting in incorrect diagnoses and treatment plans, severely endangering patient health and safety. Understanding and mitigating the effects of adversarial attacks is thus of paramount importance, necessitating a deeper exploration of model robustness and the development of techniques to safeguard against such vulnerabilities.

To assess the adversarial robustness of AI models, developers typically employ some existing adversarial attack techniques, such as TextBugger [22] and HotFlip [12] for attacking. Figure 2 shows the general framework for adversar-

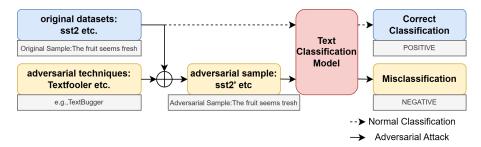


Fig. 2 The general framework for adversarial attack techniques

ial attack techniques. For NLP tasks, adversarial attacks generally start with an original dataset. For each sample in the original dataset (namely Original Sample), attack techniques identify vulnerable characters, words, or entities within the input text based on the feedback (e.g., gradient, logits, or probabilities) from the AI model under assessment. After that, the attack techniques generate new samples, known as adversarial samples, through perturbing the vulnerable elements via character insertion, word substitution, etc. Then, the generated adversarial samples are sent to the AI models under assessment. The attack is considered successful if the assessed model produces the incorrect output. sarial attacks is shown in Figure 2.

3 Model Collection and Fine-tuning Chain Construction

3.1 Model Collection

To address the research questions, we collect the open-source models for text classification on HF. Following the data collection in the previous study [3], we employ the HfApi library², a Python wrapper of HF Hub API, to collect the stored open-source models and their associated information. Based on the upload timestamp of the stored models, we collected all text classification models before January 12, 2024. For each model, we obtained information across five dimensions, i.e., Model Name, Model Card, Upstream, Dataset, and Downloads. Model Name is the unique identifier for each model on HF through which users can directly access the inference interface and obtain the prediction result for the input. Model Card is the basic description in unstructured text, typically documenting how the model is built, how to use it, performance metrics such as classification accuracy, and other fundamental information. Upstream and Dataset are optional fields representing the upstream model and the dataset used during the training or fine-tuning process when building the current model. Downloads count the number of times the model has been downloaded for the past month on HF. Finally, we obtain 45,688 open-source

https://github.com/huggingface/hfapi/

models for text classification on HF, and the detailed attributes for each model are given in our public repository.

3.2 Fine-tuning Chain Construction

Given the collected models, we further construct the fine-tuning chains. We first identify the upstream model for each collected model. Ideally, the upstream for each model is recorded through the "Upstream" attribute, with which the upstream-downstream relationship can be easily identified by parsing this field. However, since this attribute is not mandatory for the developers, it is mostly empty on HF, even though an upstream model exists. Generally, developers mention corresponding upstream models through the model descriptions in the "Model Card" attribute.

Based on this situation, we integrate the "Upstream" and "Model Card" attributes to identify upstream models. Specifically, we first check a model's "Upstream" attribute. If it is not empty, we use this attribute as the index to retrieve the upstream model by matching each collected model's "Model Name". Otherwise, we utilize the descriptions in the "Model Card" to identify the name of the upstream model for matching. Considering that the names of upstream models are typically entities within the complex unstructured texts (as shown in Figure 1), and traditional regular expression methods are ineffective for extracting such information or involving a large amount of labeled data for model training, we utilize ChatGPT³, a popularly-used large language model (LLM), guiding it with a carefully crafted prompt to extract the names of upstream models. For a collected model, if there is no upstream model name extracted from the model descriptions, it is conisidered an isolated node recorded in our dataset. Figure 3 shows the crafted prompt, where the "Instruction" gives the task description and "Example" guides the LLM to understand the task it is dealing with and the corresponding input-output format through specific examples.

Regarding the bias introduced by ChatGPT, we manually evaluate the performance of identifying upstream model names. Initially, we randomly selected 100 models that are not annotated with upstream models in the "Upstream" attributes. Three researchers manually annotate the upstream model names for each model respectively and establish a ground truth through discussion. Based on this, we evaluated the proportion of correct identification, i.e. accuracy rate, and the results showed an accuracy rate of 97% on the 100 samples, demonstrating promising reliability of our automatic upstream model identification.

After that, we match the identified results with the "Model Name" of other collected models to identify the upstream-downstream relationships. By integrating all identified upstream-downstream relationships, we obtained the fine-tuning chains. Finally, we built 29,148 fine-tuning chains, involving 31,672

³ https://openai.com/blog/chatgpt

Instruction: I want you to be able to extract upstream model information for a specific model from the Hugging Face model library

The input content is the model name and model description

Your task is to identify which upstream model the model is fine-tuned to from the input model name & model description

If you cannot find the upstream model of the input model, the upstream model is N/A.

Example1:

Input:{"Model Name":"tezign/Erlangshen-Sentiment-FineTune", "Model Description":"BERT based sentiment analysis, finetune based on https://huggingface.co/IDEA-CCNL/Erlangshen-Roberta-330M-Sentiment."} Output:{"Model Name":"tezign/Erlangshen-Sentiment-FineTune", "Upstream Model":"IDEA-CCNL/Erlangshen-Roberta-330M-Sentiment"}

Fig. 3 The prompt for identifying the upstream model name from the descriptions in "Model Card"

different models. Based on the collected models and constructed fine-tuning chains, we conduct the empirical study to answer the three research questions.

4 Answering RQ1

4.1 Experiment Design

This RQ mainly focuses on two aspects: the popularity of text classification models on HF, indicated by the number of downloads (RQ1.1), and their reuse situation within HF, indicated by the frequency serving as the upstream models (RQ1.2). For RQ1.1, given collected models (details in Section 3.1), we parse the "Downloads" attribute and rank the models according to the number of downloads in descending order, and we use it as an indicator for the model popularity. We choose the number of downloads as indicator as downloading behavior is typically an initial indication of user interest and potential further use of the model. Additionally, we focus on the models with the most downloads and analyze the proportion of the top K downloaded models to the total downloads of all collected models. These statistics will help reveal the distribution of model downloads across HF and imply their popularity. For RQ 1.2, We analyze the proportion of models that mention upstream models in the corresponding "Upstream" or "Model Card" attribute, to assess the prevalent of model reuse on the HF. Regarding the constructed model chains, we further analyze their characteristics, e.g., chain length, the most frequent upstream models, reuse counts, etc.

4.2 Results And Discussions

4.2.1 RQ1.1: How popular are the open-source text classification models on HF?

Figure 4 shows the download distribution of the text classification models of HF. In general, numerous text classification models on HF have broadly received much attention, and 38.63% of the 45,588 models are downloaded at least once within a monthly cycle. In addition, 1,536 model (about 3.36% of all the collected models) have gained significant interests with more than 30 downloads within a monthly cycle. The results highlight the popularity of the various text classification models on HF.

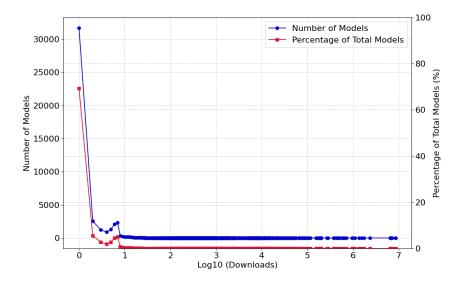


Fig. 4 The download distribution of the text classification models on HF (RQ1)

Furthermore, we also observe the long-tail effect in the downloads on HF. Specifically, the top 20 models alone account for 72,967,027 downloads, representing 86.10% of the total number. About top 0.1% of the models (46 models) account for 94.85% of the total downloads. These results also reveal that, on the fact that models receive widespread attention on HF, users exhibit a certain degree of preference for a small number of top models. For the models receiving much attention, it is non-trivial to conduct rigorous quality assessments to ensure their reliability.

Finding 1: Over 38.63% text classification models are downloaded at least once within a monthly cycle, highlighting the widespread attention of the models on HF. Furthermore, the distribution of downloads on HF demonstrates a long-tail effect that a few models at the top account for considerable downloads. This situation reveals their gained preference and the necessity for a more stringent assessment of their quality features (e.g., adversarial robustness).

4.2.2 RQ1.2: How prevalent is the model reuse within HF?

By identifying the upstream for each model (details in Section 3.2), we found that about 64.66% of the collected models have the corresponding upstream models. This indicates that models of a certain scale are built by reusing other open-source models, and substantial models along with their upstream are involved in the fine-tuning chains on HF. In addition, out of the 45,688 text classification models, 455 models are used as upstream models, accounting for 1% of the total. We found that most of them are large-scale trained models released by companies or organizations such as CardiffNLP and Microsoft. The results imply that developers are inclined to reuse such famous models rather than the unfamous ones.

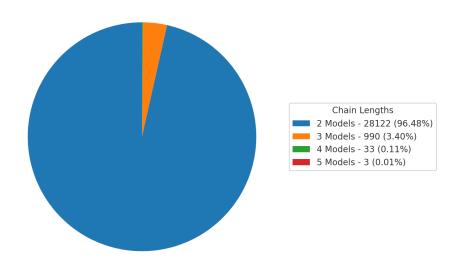


Fig. 5 Distribution of model chain lengths (RQ1)

For the model lengths, Figure 5 shows the distribution of the number of models on each model chain. Overall, among the 29,153 model chains collected,

there are an average of 2.04 models on each chain, which is significantly shorter than the length of traditional open-source software supply chains [46]. Specifically, chains primarily consist of two or three models (proportions sum up to 99.88%). The proportion of model chains made up of two models accounts for 96.48%, and those consisting of three models account for 3.40%. The shorter length of these model chains may be due to the fact that most text classification models are fine-tuned from large-scale models such as BERT and DistilBERT. Given the high performance of these base models, developers usually only need one or two rounds of fine-tuning to achieve the desired results, eliminating the necessity for multiple fine-tuning sessions.

Given the above circumstance, we further analyzed which models are more frequently used as upstream models for text classification. Generally, model reuse is prevalent and 1% model are reused at least once with HF. Table 1 shows the top 10 most popular models used as the upstream ones in the model chains on HF, Where the "Downstream (%)" column indicates the number of downstream models fine-tuned by the current model and the corresponding proportion of all upstream-downstream model pairs where the model is identified as upstream, "Downstream Task" indicates the downstream task where upstream models are most commonly applied, and "Downloads (Ranking)" shows the numbers of downloads and rankings of these models. According to "Downstream (%)", the top 10 (2.20% of all reused text classification models) most popular upstream models contribute 30.93% of model reuse for text classification on HF. The result is similar to the analysis in download volume (details in Section 4.2.1), both conforming to a certain degree of the long-tail effect. This phenomenon also emphasizes the importance of assessing the reliability of a few core models, as their potential vulnerabilities could significantly impact a wide range of downstream applications.

Table 1 The most popular upstream models in the model chains on HF (RQ1)

Model Name	Downstream Task	#Downstream (%)	#Downloads (Ranking)	
distilbert-base-uncased-finetuned-sst-2-english	general sentiment analysis	137 (10.75%)	6849685 (5)	
cardiffnlp/twitter-roberta-base-sentiment-latest	general sentiment analysis	45 (3.53%)	8622648 (1)	
cardiffnlp/twitter-roberta-base-sentiment	general sentiment analysis	38 (2.98%)	1744622 (12)	
nlptown/bert-base-multilingual-uncased-sentiment	product review sentiment analysis	33 (2.59%)	1552518 (13)	
microsoft/minilm-l12-h384-uncased	general sentiment analysis	32 (2.51%)	12363 (133)	
prosusai/finbert	financial sentiment analysis	26 (2.04%)	1368118 (14)	
roberta-large-mnli	textual entailment	23 (1.81%)	95071 (46)	
cardiffnlp/twitter-xlm-roberta-base-sentiment	general sentiment analysis	21 (1.65%)	1084083 (16)	
papluca/xlm-roberta-base-language-detection	language detection	20 (1.57%)	502579 (27)	
siebert/sentiment-roberta-large-english	general sentiment analysis	19 (1.49%)	117489 (44)	

Finding 2: Substantial models for text classification are built by reusing the existing open-source models, which forms fine-tuning chains of considerable scale within HF. Furthermore, developers of downstream models are more inclined to choose a few famous models for reuse, which emphasizes the quality of these core models.

5 Answering RQ2

5.1 Experiment Design

5.1.1 Chain and Model Selection

To investigate the adversarial robustness of open-source models (RQ2) and robustness change during model fine-tuning (RQ3), we conduct chain and model selection. First, of all the constructed model chains, we filter them by following criteria: (1) all the datasets for model training or fine-tuning on the chain should be declared to guarantee the selected subjects are of higher description quality, and (2) all the models on the chain should own at least 30 downloads to ensure that the subjects have a certain level of popularity. After that, we obtained ten upstream-downstream model pairs, and 18 opensource models were involved (3 pairs shared the same upstream model). Table 2 details the selected chains and involved models. In addition, based on the results for RQ1, we additionally introduce the model with the most downloads (mrm8488/distilroberta-finetuned-financial-news-sentiment-analysis) and the model with the most reuse (distilbert-base-uncased-finetuned-sst-2-english) on HF. Despite the fact that the model chains derived from cardiffnlp/twitterroberta-base-sentiment-latest and mrm8488/distilroberta-finetuned-financialnews-sentiment-analysis do not meet the selection criteria of this study, we decided to include these models in our research due to their top positions in terms of downloads and reuse on the Hugging Face platform. This decision was made to evaluate the security of these models, given their high practical application value. Finally, we obtained 20 models and ten upstream-downstream model chains to investigate their adversarial robustness for RQ2 and RQ3.

$5.1.2\ Adversarial\ Robustness\ Assessment\ for\ Models$

We employ six widely-used and state-of-the-art adversarial sample generation techniques for attacking to investigate the adversarial robustness, as shown below.

- TextBugger [22]. It first adopts a scoring mechanism to determine the importance of words or characters in the text based on their impact on the model's output. Then, it employs a series of perturbation techniques to generate adversarial samples for attacking, such as character insertion, deletion, swapping, or word substitution, targeting these critical elements.
- HotFlip [12]. It employs the model's gradients to determine which characters or words, when altered, will have the most significant impact on the model's decision. HotFlip then applies these perturbations to generate adversarial samples which can include flipping characters and inserting or deleting them to minimize changes to the original input while maximizing the likelihood of fooling the model into making incorrect predictions.
- **TextFooler** [18]. First, it identifies the most critical words in the input text by assessing changes in the output confidence as each word is removed

or altered. Next, TextFooler searches for semantically similar but syntactically different substitutes for these critical words to find replacements that maintain the original meaning as closely as possible. Lastly, it evaluates the new text to ensure that the substitutions not only misled the targeted model into a wrong prediction but also preserved the original text's grammatical correctness and semantic coherence, thereby keeping changes imperceptible to human readers.

- PWWS [36]. Initially, PWWS calculates the word saliency by modifying or removing each word and observing the change in the model's output. PWWS then seeks to find appropriate replacements for these words based on their semantic similarity, aiming to preserve the overall meaning of the text. Finally, PWWS re-evaluates the adversarial text to ensure that the changes are not only effective at deceiving the model but also subtle enough to appear natural and coherent to human readers.
- SCPN [15]. Firstly, SCPN identifies target sentences or phrases within the text that are important for the model's prediction. It then uses its trained paraphrase model to generate alternatives to these sentences that are semantically equivalent but lexically different.
- GAN [54]. GAN hinges on a duel between two neural networks: a generator and a discriminator. The generator creates data instances that mimic the true data distribution, aiming to fool the discriminator, which is trained to distinguish between the generator's fake instances and real data. Through this adversarial training process, the generator are guided to generate adversarial samples that are close to the original yet modified subtly to cause misclassification by the target model.

These attack techniques are designed according to various technical principles and could be used to evaluate the adversarial robustness of AI models from different aspects. Specifically, TextBugger and HotFlip represent character-level attacks, generating adversarial samples through character insertion, deletion, and substitution. TextFooler and PWWS represent word-level attacks, achieving adversarial effects through word replacement. SCPN represents sentence-level attacks, generating semantically equivalent but syntactically different sentences to confuse the model. GAN represents generative adversarial network attacks, creating samples close to the original but subtly modified to deceive the discriminator. This diverse set of attack methods ensures that the experiments cover various types of adversarial attacks, providing a robust assessment of the models' performance under different adversarial environments.

In addition, we introduced four widely used datasets, i.e., sst2 [41], IMDB [24], rotten tomatoes [32], and Amazon polarity [52] as the original datasets (the terminology is described in Section 2.2) for adversarial sample generation. These datasets are commonly used in NLP tasks and encompass various text lengths and language styles, ensuring the broad applicability and representativeness of the experimental results. In this study, we sampled 100 instances from the training sets of each of these four datasets to generate adversarial samples and evaluate the robustness of the models. With the original datasets,

we use OpenAttack [51], which is the open-source implementation of the above techniques, for adversarial attacks. We obtained the output for each generated adversarial sample through the inference interface provided by HF and observed whether the subject model produced the incorrect prediction. Then we use the attack success rate (ASR), a commonly used measure [9] referring the proportion of all the adversarial samples where models give the incorrect prediction, as the metric to explore the adversarial robustness under different attack techniques and different original datasets.

Table 2 The subject fine-tuning chains and involved models for adversarial robustness assessment (RQ2)

Chain ID	Model ID	Model Name	Downloads	Architecture
CH	M1	mrm8488/distilroberta-finetuned-tweets-hate-speech	275	BERT
	M2	hackathon-pln-es/detect-acoso-twitter-es	219	BERT
	M3	juliensimon/reviews-sentiment-analysis	1819	BERT
	M4	houssemmammeri/revsen-v1	122	BERT
$ \begin{array}{cc} M5 \\ M6 \end{array} $	M5	ProsusAI/finbert	1368118	BERT
	M6	ziweichen/finbert-fomc	203	BERT
C4 $M5$ $M7$	ProsusAI/finbert	1368118	BERT	
	M7	kk08/cryptobert	798	BERT
C5 M5 M8	ProsusAI/finbert	1368118	BERT	
	M8	yiyanghkust/finbert-esg	2583	BERT
C6 M9 M10	idea-ccnl/erlangshen-roberta-330msentiment	1819	BERT	
	M10	tezign/erlangshen-sentiment-finetune	64	BERT
C7 $M11$ $M12$	${\rm mrm} 8488/{\rm distilroberta\text{-}finetuned\text{-}financial\text{-}news\text{-}sentiment\text{-}analysis}$	5732795	BERT	
	M12	finscience/fs-distilroberta-fine-tuned	68	BERT
C8 M13 M14	jarvisx17/japanese-sentiment-analysis	1762	BERT	
	M14	minutillamolinara/bert-japanese_finetuned-sentiment-analysis	186	BERT
CQ	M15	gooohjy/suicidal-electra	1338	Electra
	M16	sentinet/suicidality	613	Electra
C10	M17	hazqeel/electra-small-finetuned-malay-english	114	Electra
	M18	haz qeel/electra-small-doa-finetuned-ms-en-v3	115	Electra

5.2 Results And Discussions

5.2.1 RQ2.1: How robust are the widely-used text classification models under existing adversarial attacks?

Figure 6 shows the average ASR of each text classification on HF under different attack approaches using the four datasets. In general, the ASRs are not low for all models, ranging from a low of 30% to a high of 78%. The results indicate that the adversarial robustness is unsatisfactory for all the studied models and the models suffer adversarial risks when directly reusing them in application scenarios.

5.2.2 RQ2.2: Are there any obvious robustness differences for these text classification models under different attack techniques and datasets?

Figure 7 shows the average ASRs of different adversarial attack methods on twenty models and four datasets. Experimental results show that using different adversarial attack methods significantly impacts the average ASR. Among

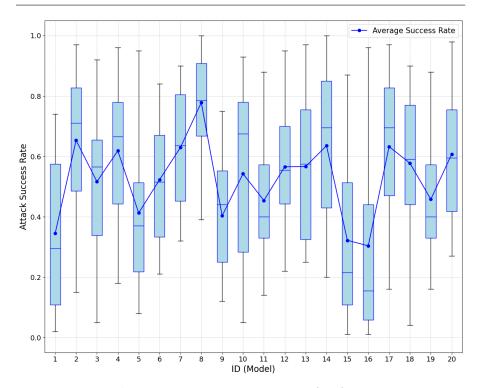


Fig. 6 The average ASRs for the text classification models (RQ2) $\,$

them, the TextFooler and PWWS methods show the highest ASR, which indicates that the model is vulnerable to attacks based on synonym substitution. The lower success rates of TextBugger and GAN mean that the model is better resistant to these attacks.

Figure 8 shows the average ASRs of different datasets on the wide-used models for text classification. The experimental results indicate that the choice of dataset has limited impact on the average ASR of the adversarial attacks, although there are significant differences in belonging domain, language styles and terminologies among the four datasets. Furthermore, considering the robustness risks associated with the models, security-sensitive users can consider designing targeted defense methods (such as adversarial sample detection [44, 45], adversarial training [37, 20], etc.) based on the principles and application scope of the attack techniques, to enhance the security in real-world scenarios.

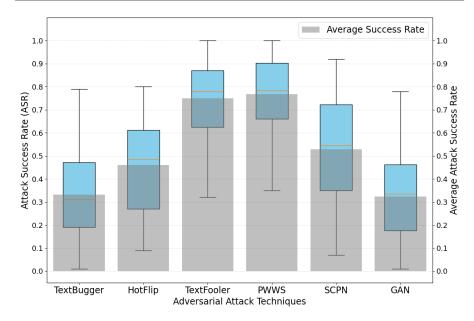


Fig. 7 The ASRs of widely-used models for text classification under different adversarial attack techniques (RQ2)

Finding 3: The widely used models for text classification exhibit non-negligible attack vulnerability (30% - 78% ASR), indicating the low attack robustness of open-source AI models. In addition, the attack techniques have a noticeable impact on the ASRs of these models. Users who want to reuse open-source models directly in their application scenarios may face a certain degree of robustness risk.

6 Answering RQ3

6.1 Experiment Design

Given the 10 selected model chains (details in Section 5.1.1), we further explore the impact of fine-tuning on the adversarial robustness, aiming to guide the developers with the adversarial risks when fine-tuning the open-source models on HF for model reuse.

For RQ 3.1, we calculated the average ASR of adversarial attacks on upstream models across six attack approaches powered with four datasets (denoted as ASR_{up}), as well as the average ASR for the corresponding downstream models (denoted as ASR_{down}). After that, we calculated $\delta = ASR_{down} - ASR_{up}$, which is the difference between the average success rates of attacks on downstream models and their upstream counterparts. If δ is positive, it

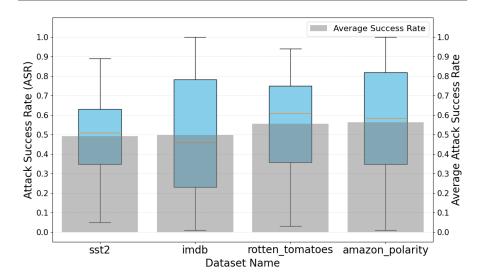


Fig. 8 The ASRs of widely-used models for text classification under different datasets (RQ2)

indicates that the downstream model is more susceptible to attacks compared to the upstream model, suggesting that the fine-tuning process may have reduced robustness; otherwise, it indicates an improvement in the robustness of the downstream model.

For RQ 3.2, for an original sample that successfully attacks the upstream model, i.e. vulnerable sample, we explore whether it still successfully attacks the corresponding downstream model. To quantify this transmissibility, we defined "transferable rate" as the proportion of samples that successfully attacked the upstream model and were also successful in attacking the downstream model. Specifically, we use different adversarial attack techniques equipped with different original datasets to generate adversarial samples. For each upstream-downstream pair involved in 10 model chains, we identify vulnerable samples through the upstream model and examine whether they could successfully attack the corresponding downstream model. In the end, we calculate their average "transferable rate" for each chain separately.

6.2 Results And Discussions

6.2.1 RQ3.1: Are text classification models more robust or more vulnerable to common adversarial attacks after fine-tuning?

Figure 9 shows the average value of δ across 24 different attack scenarios (covering six attack approaches and four original datasets). For C1-C8, the δ is positive, indicating they exhibit consistent changes in adversarial robustness before and after fine-tuning, with the downstream model becoming less robust

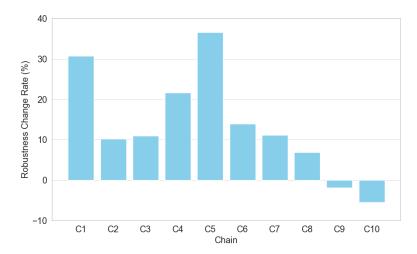


Fig. 9 The change in adversarial robustness before and after fine-tuning for each upstream-downstream pair (RQ3)

(average delta is 16.84%). Especially for C1 and C5, their adversarial robustness experiences the most significant declines, exceeding 30% and 35% in δ , respectively. However, for the other two (C9, C10), their robustness improves. Further analysis revealed that all models in the C1-C8 follow the BERT architecture [11], while C9 and C10 belong to Electra architecture [8], which will be discussed later.

Finding 4: After fine-tuning, whether robustness improves or declines is largely related to the architecture the model employs. For the currently dominant BERT architecture, its model robustness tends to decrease compared to the upstream model. While, in specific architectures such as Electra, empirical evidence suggests that their robustness does not decrease but enhances, thus rendering it particularly advantageous for security-conscious users.

6.2.2 RQ3.2: Can the vulnerability embodying in vulnerable samples transfer along the model chains on HF?

Figure 10 shows the transferable rates using different adversarial attack techniques. In general, the average transferable rate reaches 78.57%, indicating that the vulnerable samples could be transferred along the model chains to some extent on HF. In particular, the transferable rates exceed 60% for 4/6 adversarial attack approaches, i.e., HotFlip, TextFooler, PWWS, and SCPN. This indicates that, for most techniques, if original samples could success-

fully attack the upstream model, they are more likely to be vulnerable to the corresponding downstream model.

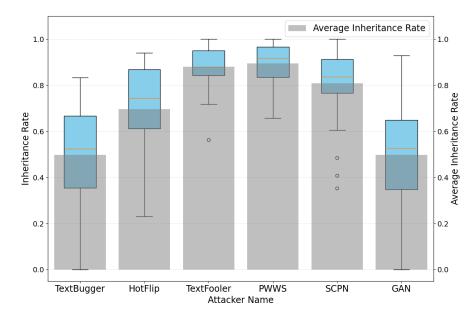


Fig. 10 The average transferable rates using different adversarial attack techniques (RQ3)

This transitivity suggests that fine-tuning has not enhanced the downstream models' resistance to vulnerabilities known in upstream models and may even perpetuate or amplify these vulnerabilities within the model chains. This is especially critical for popular models frequently used for further finetuning, whose inherent vulnerabilities could impact a wide range of downstream applications. Therefore, developers should pay more attention to these potential adversarial risks when fine-tuning models.

Finding 5: The vulnerability embodying in vulnerable samples could transfer along the model chains on HF with an average rate of 78.57%. When users are preparing to fine-tune based on open-source models, it is necessary to pay attention to the inherent robustness risks of upstream models, as they are likely to be transmitted to downstream applications.

7 Implications

In this section, we'd like to elaborate more on the implications of our empirical studies for readers of interest to take away.

The guidance of selecting proper reuse manner for security-critical applications. Given the open-source models, users typically reuse them in two manners: (1) directly apply them in applications without modifications and (2) fine-tune them to suit current application scenarios better. Based on the results in RQ2 and RQ3, the two manners suffer different adversarial risks. Therefore, disregarding the considerations of data and computational resources required for fine-tuning, users need to comprehend the model's architecture to be reused and its regularity of robustness change during the fine-tuning process to make the proper choice in the security-critical applications. For instance, for open-source models with BERT architecture, selecting the most appropriate models from vast and diverse models on HF for direct applications would be more secure, as fine-tuning would degrade the adversarial robustness. On the contrary, fine-tuning is the best way to reuse the models with Electra architecture to resist adversarial attacks.

The significance of vulnerable samples for targeted defenses. In our empirical study, the generated adversarial samples can not only be used to assess the adversarial robustness of open-source models but also hold significance for targeted defense approaches (e.g., adversarial sample detection and adversarial training) based on therein vulnerable samples. For example, adversarial sample detection techniques often involve training a binary classification model to predict whether an input is an adversarial sample. The adversarial samples generated in our study, especially the vulnerable samples that successfully attack the models, can be used as training data to improve the performance of detection models, achieving targeted defenses. Additionally, vulnerable samples can be used for data augmentation, enhancing the target model through adversarial training.

Leverage the upstream model's vulnerable samples for a more targeted attack on the downstream models. From the finding drawn in Section 6.2.2, it is evident that if we have access to the vulnerable samples of the upstream model, namely the data instances that successfully attack the upstream model, then utilizing these samples to attack the downstream model can achieve an average success rate of 78.57%, which is 20.33% (78.57% vs. 58.24%) higher compared to using a standard dataset. This implies that the attacks on the upstream model can provide targeted information, leading to improved success rates in attacking the downstream model.

Inspiration from Electra architecture for robustness enhancement. In Section 6.2.1, we find that models with Electra architecture demonstrate robustness improvement after fine-tuning. Electra employs a replaced token detection (RTD) strategy, involving a generator and a discriminator; the generator is responsible for replacing tokens in the sentence, while the discriminator determines whether each token is a replacement. The RTD strategy employed in Electra architecture may lead to robustness improvement due to its similarity to adversarial attack operations (namely replacing tokens to create disturbances). Electra, by training the model to recognize these minor changes, inadvertently enhances the model's resistance to such disruptions. Inspired by this observation, targeted considerations for adopting techniques

aligned with replacing tokens can also be explored to implement robustness enhancement.

8 Threats to Validity

External Validity. The external threats are related to the generalization of the empirical studies. First, our subject models are collected from one AI model hub, i.e., HF. The relevant findings may differ for platforms like TensorFlow Hub and PyTorch Hub. However, HF is one of the representative platforms attracting substantial attention and usage. The empirical study based on HF is still of critical guiding significance. Second, our study focuses on text classification models, and the results may not be generalized to other tasks, such as text generation and image classification. Even so, text classification is a vital AI task widely used in many scenarios, such as intelligence analysis, social content analysis, and news classification. Related research also holds broad application prospects. Third, the robustness of models and their fine-tuning are assessed through six adversarial attack techniques and four datasets. The results may differ under other attack techniques and datasets. However, the techniques and datasets employed are all prevalent and representative. Additionally, they are used for robustness assessment in previous studies [35, 30, 39], which could alleviate the threat.

Internal Validity. The internal threats relate to experimental errors and biases. First, we employ the ChatGPT to identify the upstream model names from the descriptions in "Model Card". The bias of ChatGPT may be introduced in our study. However, ChatGPT is a widely recognized commercial system that performs well in information extraction tasks [21]. Furthermore, we evaluate the identification performance with a sampled dataset (details in Section 3.2), and 97% accuracy reflects its credibility. Second, for two adversarial attack techniques, TextBugger and GAN, there is randomness when generating adversarial samples according to their technical principles. Therefore, the attack performance is unstable, which may threaten the experimental results. To alleviate the threat, we run the attack technique three times for each experimental setting, and the average performance is utilized.

9 Related Work

9.1 AI Model Reuse

For AI models, the development is a resource-intensive process involving complex algorithm design, long-time computational resource consumption, and potential data labeling. Reusing open-source models provides a promising way to build effective task-specific models with relatively lower costs. Correspondingly, techniques and empirical studies for reusing AI models in their reusability, challenges, and respective improvement methods have emerged.

Qi et al. proposed a tool called SEAM, which is specifically designed to redesign trained deep neural network models to improve their reusability [33]. Pan et al. conducted an empirical study to analyze the errors that occur when reusing pre-trained NLP models, and proposed strategies to reduce these errors [31]. Jiang et al. systematically evaluated the decision-making process for reuse of pre-trained models, and clarified the key attributes and challenges in reuse [16]. Taraghi et al. conducted a mixed-method empirical study on pretrained model reuse on HF, analyzed the challenges faced by users and the trend of model reuse, and proposed relevant guidance suggestions [42]. Zhao et al. developed a query framework called MMQ, which optimizes the model reuse process, enabling users to more accurately select the pretrained models needed for their tasks [53]. Davis et al. explored the challenges and future direction of deep neural network reuse, and discussed in detail different types of reuse methods [10]. Together, these studies advance the theory and practice of AI model reuse. However, despite the above empirical studies on model reuse from various perspectives, there are fewer studies on the adversarial robustness of the open-source AI models, especially their fine-tuning chains. Our empirical study can fill this gap.

9.2 Empirical Studies On Hugging Face

Since its inception, HF has become a pivotal platform in the NLP community. This platform not only offers a rich repository of open-source AI models but also fosters the sharing of open science and technology. Meanwhile, HF has attracted substantial empirical studies in the literature. Mitchell et al. introduced the concept of Model Cards, which has been used as an essential attribute of maintained models on HF, a standardized document designed to increase the transparency of AI models by detailing their performance and evaluation processes to promote the responsible use of models and a better understanding of their behavior [29]. Ait et al. evaluated the feasibility of using the Huggingface Hub as a data source for empirical research by analyzing its features [1]. McMillan-Major et al. demonstrate how the HF platform and the GEM (Geospace Environment Modeling) workshop can be used to implement and customize standardized documentation templates for Model Cards and Data Cards to improve the quality and transparency of documentation of datasets and models in NLP and generative domains [26]. Castaño et al. explore the dynamics of community engagement, model evolution, and maintenance, providing crucial insights into future model development strategies [5]. Jiang et al. analyzed the naming conventions of pretrained models on the Hugging Face platform and developed an automated tool named DARA to detect anomalies in naming [17]. Yang et al. analyzed the current state of AI dataset documentation on the Hugging Face platform, the community's usage of dataset cards, and proposed guidelines for writing effective dataset cards [49]. Taraghi et al. examine the challenges and trends in reusing pre-trained models in the community, highlighting the problems users face

in understanding and applying models and proposing strategies to address them [42]. Castaño et al. focus on carbon emissions during model training on the platform and promote initiatives for green model development [4].

10 Conclusion

To reuse the open-source models on HF, it is critical to understand their adversarial robustness against potential human interference to ensure the security of domain models in practical applications. This paper aims to investigate the adversarial robustness of open-source AI models and their fine-tuning chains, to provide insights into the potential adversarial risks. Empirical analysis on the text classification models shows that, under the prevalent circumstance of model reuse, open-source text classification models exhibit certain deficiencies in adversarial robustness. Furthermore, inherent robustness vulnerability in the text classification models exhibits a certain degree of transitivity in the fine-tuning chains, which highlights the security concerns for security-sensitive users in the downstream applications. In the future, we are going to explore the adversarial robustness of more types of open-source AI models and design automated methods for targeted defense or robustness enhancement.

Data Availability Statements

The models used in this experiment and their upstream model information are available in a GitHub repository. The repository can be accessed through the following link: https://github.com/Xcasdfas/Adversarial-Robustness-of-Open-source-AI-Models-and-Fine-Tuning-Chains

Conflict of interest

The authors declare that they have no conflict of interest.

References

- 1. Adem Ait, Javier Luis Cánovas Izquierdo, and Jordi Cabot. On the suitability of hugging face hub for empirical studies, 2023.
- 2. Lu Bai, Weixing Ji, Qinyuan Li, Xilai Yao, Wei Xin, and Wanyi Zhu. Dnnabacus: Toward accurate computational cost prediction for deep neural networks, 2022.
- 3. Joel Castaño, Silverio Martínez-Fernández, Xavier Franch, and Justus Bogner. Analyzing the evolution and maintenance of ml models on hugging face. arXiv preprint arXiv:2311.13380, 2023.

- 4. Joel Castaño, Silverio Martínez-Fernández, Xavier Franch, and Justus Bogner. Exploring the carbon footprint of hugging face's ml models: A repository mining study. In 2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). IEEE, October 2023. doi: 10.1109/esem56168.2023.10304801. URL http://dx.doi.org/10.1109/ESEM56168.2023.10304801.
- 5. Joel Castaño, Silverio Martínez-Fernández, Xavier Franch, and Justus Bogner. Analyzing the evolution and maintenance of ml models on hugging face, 2024.
- Hao-Yun Chen, Jhao-Hong Liang, Shih-Chieh Chang, Jia-Yu Pan, Yu-Ting Chen, Wei Wei, and Da-Cheng Juan. Improving adversarial robustness via guided complement entropy. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 4880–4888, 2019. doi: 10.1109/ICCV.2019.00498.
- 7. Pin-Yu Chen and Sijia Liu. Holistic adversarial robustness of deep learning models. pages 15411–15420, 2022. doi: 10.1609/aaai.v37i13.26797.
- 8. Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators, 2020.
- 9. Cuong Dang, Dung D. Le, and Thai Le. A curious case of searching for the correlation between training data and adversarial robustness of transformer textual models, 2024.
- James C. Davis, Purvish Jajal, Wenxin Jiang, Taylor R. Schorlemmer, Nicholas Synovic, and George K. Thiruvathukal. Reusing deep learning models: Challenges and directions in software engineering, 2024.
- 11. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.
- 12. Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. Hotflip: White-box adversarial examples for text classification, 2018.
- 13. Z. Fu, A. M. So, and Nigel Collier. A stability analysis of fine-tuning a pre-trained model. ArXiv, abs/2301.09820, 2023. doi: 10.48550/arXiv.2 301.09820.
- 14. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- 15. Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. arXiv preprint arXiv:1804.06059, 2018.
- Wenxin Jiang, Nicholas Synovic, Matt Hyatt, Taylor R. Schorlemmer, Rohan Sethi, Yung-Hsiang Lu, George K. Thiruvathukal, and James C. Davis. An empirical study of pre-trained model reuse in the hugging face deep learning model registry, 2023.
- 17. Wenxin Jiang, Chingwo Cheung, Mingyu Kim, Heesoo Kim, George K. Thiruvathukal, and James C. Davis. Naming practices of pre-trained models in hugging face, 2024.

18. Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025, 2020.

- Jinkyu Kim, Teruhisa Misu, Yi-Ting Chen, Ashish Tawari, and John Canny. Grounding human-to-vehicle advice for self-driving vehicles. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10591–10599, 2019.
- 20. Alexey Kurakin, I. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *ArXiv*, abs/1611.01236, 2016.
- Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. Evaluating chatgpt's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness. CoRR, abs/2304.11633, 2023.
- 22. Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. Textbugger: Generating adversarial text against real-world applications. arXiv preprint arXiv:1812.05271, 2018.
- 23. Zihao Lin, Yan Sun, Yifan Shi, Xueqian Wang, Lifu Huang, Li Shen, and Dacheng Tao. Efficient federated prompt tuning for black-box large pretrained models. ArXiv, abs/2310.03123, 2023. doi: 10.48550/arXiv.2310.03123.
- 24. Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.
- 25. A. Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ArXiv*, abs/1706.06083, 2017.
- 26. Angelina McMillan-Major, Salomey Osei, Juan Diego Rodriguez, Pawan Sasanka Ammanamanchi, Sebastian Gehrmann, and Yacine Jernite. Reusable templates and guides for documenting datasets and models for natural language processing and generation: A case study of the huggingface and gem data and model cards. arXiv preprint arXiv:2108.07374, 2021.
- 27. Matiur Rahman Minar and Jibon Naher. Recent advances in deep learning: An overview. ArXiv, abs/1807.08169, 2018. doi: 10.13140/RG.2.2.24831.10403.
- 28. Shreyash Mishra, Suryavardan S, Amrit Bhaskar, Parul Chopra, Aishwarya N. Reganti, Parth Patwa, Amitava Das, Tanmoy Chakraborty, Amit P. Sheth, and Asif Ekbal. FACTIFY: A multi-modal fact verification dataset. In Proceedings of the Workshop on Multi-Modal Fake News and Hate-Speech Detection (DE-FACTIFY 2022) co-located with the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI 2022), Virtual Event, Vancouver, Canada, February 27, 2022, 2022.

- 29. Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the* conference on fairness, accountability, and transparency, pages 220–229, 2019.
- 30. John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp, 2020.
- 31. Rangeet Pan, Sumon Biswas, Mohna Chakraborty, Breno Dantas Cruz, and Hridesh Rajan. An empirical study on the bugs found while reusing pre-trained natural language processing models. arXiv preprint arXiv:2212.00105, 2022.
- 32. Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*, 2005.
- 33. Binhang Qi, Hailong Sun, Xiang Gao, Hongyu Zhang, Zhaotian Li, and Xudong Liu. Reusing deep neural network models through model reengineering. In 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), pages 983–994. IEEE, 2023.
- 34. Ravi Raju, Kyle Daruwalla, and Mikko H. Lipasti. Accelerating deep learning with dynamic data pruning. *ArXiv*, abs/2111.12621, 2021.
- 35. Mrigank Raman, Pratyush Maini, J. Zico Kolter, Zachary C. Lipton, and Danish Pruthi. Model-tuning via prompts makes nlp models adversarially robust, 2023.
- 36. Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In Proceedings of the 57th annual meeting of the association for computational linguistics, pages 1085–1097, 2019.
- 37. Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free!, 2019.
- 38. Rajesh Shinde and C. Kalpana. Advancements in deep learning: A comprehensive review. REST Journal on Data Analytics and Artificial Intelligence, 2023. doi: 10.46632/jdaai/2/2/7.
- 39. Chenglei Si, Ziqing Yang, Yiming Cui, Wentao Ma, Ting Liu, and Shijin Wang. Benchmarking robustness of machine reading comprehension models, 2021.
- 40. Hossein Siadati, Sima Jafarikhah, Elif Sahin, Terrence Brent Hernandez, Elijah Lorenzo Tripp, and Denis Khryashchev. Devphish: Exploring social engineering in software supply chain attacks on developers, 2024.
- 41. Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/antholo

- gy/D13-1170.
- 42. Mina Taraghi, Gianolli Dorcelus, Armstrong Foundjem, Florian Tambon, and Foutse Khomh. Deep learning model reuse in the huggingface community: Challenges, benefit and trends, 2024.
- 43. Vivek Velayutham, Gunjan Chhabra, Sanjay Kumar, Avinash Kumar, Shrinwantu Raha, and Gonesh Chandra Sah. Analysis of deep learning in real-world applications: Challenges and progress. *Tuijin Jishu/Journal of Propulsion Technology*, 2023. doi: 10.52783/tjjpt.v44.i2.150.
- 44. Jingyi Wang, Jun Sun, Peixin Zhang, and Xinyu Wang. Detecting adversarial samples for deep neural networks through mutation testing, 2018.
- 45. Jingyi Wang, Guoliang Dong, Jun Sun, Xinyu Wang, and Peixin Zhang. Adversarial sample detection for deep neural network through model mutation testing. In 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE). IEEE, May 2019. doi: 10.1109/icse.2019.00126. URL http://dx.doi.org/10.1109/ICSE.2019.00126.
- 46. Shenao Wang, Yanjie Zhao, Xinyi Hou, and Haoyu Wang. Large language model supply chain: A research agenda, 2024.
- 47. Mingyuan Xin and Yong Wang. A summary of deep learning algorithms. pages 301–306, 2020. doi: 10.1007/978-3-030-53980-1_45.
- 48. Hitomi Yanaka, Yuta Nakamura, Yuki Chida, and Tomoya Kurosawa. Medical visual textual entailment for numerical understanding of vision-and-language models. In *Proceedings of the 5th Clinical Natural Language Processing Workshop, ClinicalNLP@ACL 2023, Toronto, Canada, July 14, 2023*, pages 8–18, 2023.
- Xinyu Yang, Weixin Liang, and James Zou. Navigating dataset documentations in ai: A large-scale analysis of dataset cards on hugging face, 2024.
- 50. Yang Yang, De chuan Zhan, Ying Fan, Yuan Jiang, and Zhi-Hua Zhou. Deep learning for fixed model reuse. pages 2831–2837, 2017. doi: 10.160 9/aaai.v31i1.10855.
- 51. Guoyang Zeng, Fanchao Qi, Qianrui Zhou, Tingji Zhang, Zixian Ma, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. Openattack: An open-source textual adversarial attack toolkit. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-demo.43. URL http://dx.doi.org/10.18653/v1/2021.acl-demo.43.
- 52. Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification, 2016.
- 53. Minjun Zhao, Lu Chen, Keyu Yang, Yuntao Du, and Yunjun Gao. Finding materialized models for model reuse, 2023.
- 54. Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. arXiv preprint arXiv:1710.11342, 2017.