Prompt-Injection based Adversarial Attacks in Large Language Models

Muhammad Umar Zeshan

December 2023

1 Introduction

This proposal is presented to discuss the potential idea and working plan to deal with the trendy problem of prompt attacks in Large Language Models LLMs. To understand the idea better, first, the core concepts like LLMs, Prompt Learning, and Adversarial Learning need to be introduced.

1.1 Introduction to LLMs

AI and natural language processing have been transformed by large language models (LLMs) like GPT-3 and GPT-4. Their influence is felt in a variety of industries, from content creation to chatbots and virtual assistants. They are effective instruments for work automation and improving human-computer interactions. However, as they grow more ingrained in culture and technology, ethical issues, bias, and responsible use become crucial factors to take into account.

In addition to their remarkable capabilities, LLMs are currently being incorporated into various applications at a rapid and extensive pace. By contacting additional APIs, these tools can function as an agent for the user and provide interactive chat, and summaries of the documents or search results that have been retrieved. Bing Chat, Bard, Microsoft 365, Security Copilots, and other ChatGPT plugins were all introduced in the months following ChatGPT, with new announcements coming out practically every day. We contend, however, that insufficient safety assessments and guardrails are accompanying this drive toward AI integration. A new paradigm for using in-context learning to solve a variety of (NLP) tasks has also been brought about by the development of LLMs. In-context learning, an innovative approach to prompt engineering, prepends test cases with demos (also known as in-context examples) to help LLMs perform better while concluding a variety of tasks. The effectiveness of few-shot adaption of language models with in-context learning has been shown in many studies; nevertheless, there is still some concern regarding possible security problems related to the use of demonstrations.

1.2 Prompts in LLMs

An LLM receives instructions through a prompt. A prompt should ideally generate an answer that is accurate, sufficient in both form and content, and of the appropriate length. Prompting has gained popularity in the last two years and is a useful method of using pre-trained models. Under this paradigm, downstream tasks are reformulated using a textual prompt to seem more like those completed during the original LM training, as opposed to pre-trained LMs being adapted to downstream tasks via objective engineering. For instance, upon identifying the sentiment of a movie review, such as "It's the best movie I've ever seen," we may proceed with a prompt like "Overall, it's a... movie," and ask the LM to complete the sentence with a word that conveys emotion. Alternatively, if we choose a prompt such as "English: I enjoy this film. French:... ", an LM might be able to provide a French translation to fill in the blank. The benefit of this approach is that task-related knowledge may be readily elicited from LM with a set of suitable prompts.

In essence, prompting is asking a natural language question that will elicit the desired response from the model.

Well-known language models like ChatGPT, Bard, and Llama are frequently trained on uncensored textual data from the internet, which contains a sizable proportion of offensive and inappropriate content. When made available for public usage, a model trained on such data may be able to generate damaging content. Consequently, several human feedback-incorporating fine-tuning procedures have been developed to guarantee that LLM outputs are both safe and consistent with human values. These methods employ human oversight to guide an LLM toward producing information that is in line with moral and ethical standards. An aligned model is supposed to refuse rather than comply with a request made by a harmful user when presented with it.

Guardrails for LLMs have come a long way, but there are still several "jailbreak" methods that can be used by a hostile user to get around the security. Wei et al. (2023), for instance, demonstrate how even requesting an LLM to start its response with "Absolutely! Here's" may deceive the model into granting the user's detrimental request. There are also many other well-known instances of advertisements being used to enhance hazardous prompts, such as the Do Anything Now (DAN) jailbreak. According to Zou et al.'s research from 2023, adversarial sequence creation can be automated, producing an infinite number of these attacks. Furthermore, they demonstrate how safety precautions can be circumvented by appending a single adversarial sequence to several damaging prompts.

1.3 Adversarial attacks detection in RSSE

The pace of technological development and the amount of information are both daunting in the field of software engineering (SE). The steady influx of code, documentation, tools, and best practices is a challenge for software engineers. Recommender systems have become essential tools in this setting, utilizing data

analytics, machine learning, and artificial intelligence to give software engineers, testers, and project managers individualized and context-aware advice. Recommender systems, also known as recommendation systems or recommendation engines, are an area of data science and artificial intelligence that aims to help users choose from a wide range of options. To create individualized recommendations, these systems examine user behavior, preferences, and previous data. These suggestions in SE may pertain to a variety of areas, such as tool selection, issue tracking, documentation, code, and documentation.

The role of Recommender systems is very evident in a lot of software applications. Developers frequently run into problems while looking for pertinent code snippets, libraries, or functions during the development phase. By recommending code segments that fit the present coding context, recommender systems lessen this strain. These solutions speed up development and improve code quality, encouraging more effective and error-free programming.

1.4 Adversarial Attacks in RSSE

In Recommender Systems in Software Engineering (RSSE), adversarial attacks refer to conscious attempts to trick or manipulate recommender systems to produce recommendations that are unfavorable or destructive. These attacks may result in poor choices, coding flaws, or other undesirable outcomes, which can have serious effects on software engineering. There are so many hidden or clear motivations for the attackers in this regard. Attackers may try to increase the prominence of their own contributions or initiatives by manipulating suggestions. Disgruntled team members or outside parties may attempt to sabotage a project by undermining recommendations, which could result in subpar code or unstable projects.

Several recommender systems for software engineering (RSSE) have been developed in recent years to aid developers in their work and, perhaps, lessen the growing information overload brought on by the availability of data from numerous sources. The learning materials of the recommenders could be vulnerable to malicious attacks because these sources are open to alterations and contributions from the general public. In other words, it is possible to trick software recommender systems by taking advantage of these sources. Adversarial attempts produce perturbations to trick and confuse systems by breaking them down, impairing their ability to provide recommendations. For instance, an adversarial attack on recommender systems may support or disparage a product, depending on the intent, which would have a detrimental effect on the final recommendations. Similarly, malicious users may expose recommender systems to hazardous artifacts by altering training data that is accessible through OSS platforms. Software system disruptions may arise if a recommender is trained to deliver harmful outcomes based on Adversaries. For instance, a recent study reveals that there have been attempts to force Android apps to open ports covertly, enabling unwanted access. Security concerns in machine learning systems and all-purpose recommender systems are investigated via research on adversarial machine learning (AML).

2 Recent Works in Adversarial Attacks in Prompt-based Learning

In recent years, especially with the advancements in the research interests in the Large Language Models, the prompts have become more popular and useful. Although it is quite a new and trendy research topic, some of the researchers have put their foot on the paddle to conduct some useful research activities to make an impactful contribution. Some of the related papers and their contributions in this regard are discussed below.

One of the very first adversarial attack methods for prompts was presented by Xu et al. [13]. They investigate the widespread weaknesses in the prompt-based learning paradigm from the standpoints of adversarial and backdoor attacks, based on whether the attackers influence the pre-training phase. In terms of backdoor attacks, they demonstrate that if practitioners use backdoored pre-trained models, the prompt-based models' output will be manipulated by backdoor triggers. In the event of an adversarial attack, they demonstrate that prompt-based models perform worse when the input text is entered into hostile triggers, which are made up solely of plain texts. Additionally, they examine and suggest a possible defense against our assault techniques.

Yu et al. [16] introduced the prompt approach, a novel textual adversarial approach, to produce adversarial instances that are semantically compatible with the original input. Prompt-Attack is their prototype and specifically, it uses a span-level segment identification strategy to extract more target segments from the input and improves the input's semantics by fusing it with prompts that represent the semantics of each target segment to produce semantically consistent segment replacements. Ultimately, it creates adversarial examples by substituting the significant segments. According to their vast experimental results, Prompt-Attack enhances the semantic consistency ratings of current attacks by 48 percent on average.

Milliere et al. [11] showed that text-guided picture generation models are vulnerable to focused adversarial attacks thanks to two new methods, evocative prompting and macaronic prompting. Combining word fragments from several languages to create new hybridized words that make little sense to human speakers yet reliably elicit particular visual connections in picture-generating models is known as macaronic prompting. While evocative prompting does not provide a systematic method for creating nonce strings that may consistently and successfully elicit particular visual associations, thus, it poses a less evident risk. As such, it is primarily restricted to imprecise correlations with ideas associated with wide morphological characteristics of words or languages. In general, maraconic prompting is a more practical approach for malicious adversarial attacks against text-guided picture-generating algorithms than evocative prompting. It draws attention to how inadequate keyword-based blacklists are in these kinds of approaches for content filtering.

The vulnerability of the continuous prompts in NLP against adversarial backdoor attacks is discussed by Cai et al. [2]. Their research shows that

prompt-based model backdoor attacks are severely hampered by few-shot circumstances. They suggest using a lightweight, task-adaptive algorithm called BadPrompt to backdoor attack constant prompts to overcome this difficulty. BadPrompt consists of two modules and can produce an invisible and efficient trigger for every sample. These two modules are the production of trigger candidates and adaptive trigger optimization. They look into the continuous prompts' vulnerability and discover empirically that backdoor assaults are a simple way to take control of them. In particular, they successfully tackled a few sample prompts, such as DART and P-Tuning, which raises serious concerns in the field of continuous prompt-based learning.

Yang et al. [15] investigate how the prompting paradigm may be used to strengthen resilience and probe Pre-Trained Language Models (PLM) vulnerabilities. They put into practice a basic prompting-based technique to improve robustness and generate adversarial examples. It demonstrates the enormous potential of the prompting paradigm for improving PLM comprehension and developing trustworthy NLP systems.

To increase the adversarial robustness of a fixed model attest time, Chen et al. [3] create a unique visual prompting (VP) approach called C-AVP (Class-wise Adversarial Visual Prompting). According to them, this is the first attempt to investigate how VP could be used in adversarial defense, in comparison to current VP approaches. They demonstrate that, for a fixed model, the direct integration of VP into robust learning is not a viable adversarial defense at test time. To tackle this issue, they suggest using C-AVP to generate a group of visual cues and collaboratively strengthen their interrelations to improve robustness. In the field of Computer Vision, Liu et al. [8] presented RIATIG, which uses genetically based optimization techniques to consistently locate natural adversarial cases. To create a successful adversarial example, they suggest a two-step attack methodology. In the first stage, they locate a feasible adversarial example; in the second stage, they fine-tune the sample quality.

Most of these above-discussed research studies are concerned with attack methods and adversarial prompt injections, and the defense method is hardly presented. The defense method can be considered using the GAN-based defense methods against adversarial attacks. GAN-based defense methods are being used in countering adversarial attacks in computer vision problems.

3 Our Methodology

The data that the LLM will be analyzing for the task is represented by the data prompt. Generally, the data prompt originates from an outside source, such as the Internet. An email received by the user in a spam-detection activity, for example, could serve as the data prompt. Alternatively, the data prompt could be a text document that has been downloaded.

From the Internet in a translation task where the user's goal is to convert the written document into a different language; in a search task, the data prompt can be a webpage on the Internet. Our attack model is presented below which

describes the different kinds of attacks and the combination of these attacks to form a new threat model.

3.1 Limitations in previous methods

Several recent studies, including research papers and blog postings, have demonstrated that LLM-Integrated Application is susceptible to rapid injection attacks. The main drawback of previous research is that it is case study oriented; for example, it did not define the attacker's objective in rapid injection assaults in terms of the type of the attack, attacker wants to carry. Specifically, they illustrate the effectiveness of the suggested attacks in a few instances. Use the translation work as an illustration. Rather than simply translating a text into English, they demonstrated how an assailant may lead an LLM astray and have them compose a sonnet about pandas. The main drawback of this kind of case-by-case research is how difficult it is to come up with novel prompt injection attacks or carry out a thorough analysis and comparison of other prompt injection assaults. Even the latest studies are focusing on some particular type of prompt injections, which also allow for enhancing the attack by combining all the possible attack scenarios.

3.1.1 Research Questions

As the limitations of the research are explained above, two important research questions can be drawn from that:

RQ1: What is the clear objective of the Attacker in terms of attack? What type of attack does he want to apply?

RQ2: What if the defense system is only suitable for a particular type of attack while the attacker is carrying multiple types of attacks?

To answer these important research questions, the methodology is developed in a way to answer these questions in the following manner respectively: (explained in detail in the next section)

Answer to RQ1: To understand the goal of the attacker and to discuss the type of attack, all kinds of adversarial attacks defined in literature are discussed which are relevant in this new trendy topic of prompt learning attacks.

Answer to RQ2: A novel threat model is proposed, in which a combined attack model is applied which concatenates all the possible types of attacks in prompts by the attacker.

3.2 Types of Adversarial Attacks in Prompts

To answer the RQ1, different types of adversarial attacks are discussed below.

3.2.1 Backdoor Attacks

Kwon et al. [7] discussed the issue of backdoor attacks in deep neural networks DNNs damaging the classification results of the networks. Attackers can actively acquire DNN training data through backdoor assaults, which they can

use to train extra harmful data, such as the precise trigger. In normal circumstances, DNNs categorize normal data accurately, however malicious material with a particular trigger learned by attackers can lead to DNN misclassification. Further, they elaborated on the positive role of backdoor attacks as well, as they can be used to create confusion for the enemy in war situations. Although they highlighted the backdoor adversarial attacks well, still they are only concerned about their effects in terms of DNNs without presenting the complete picture in terms of the consequences of Backdoor attacks in software engineering.

Rohan et al. [10] also elaborated on the effects of backdoor attacks in machine learning models. They mentioned that despite not having access to the model parameters or the input data used to train the model, black box attacks can still be utilized to deceive a target model. Later they developed a defense mechanism to tackle the backdoor attacks.

3.2.2 Data Poisoning Attacks

Dang et al. [4] also highlighted the issue of vulnerability of DL methods in dealing with adversarial attacks. One of the most recent and dangerous attacks is the data poisoning attack and it is different than the typical adversarial attack. They highlighted the difference between adversarial and data poisoning attacks' by identifying the phase of the DL method in which the attack is occurring. Attacks that happen during a training phase are adversarial attacks (also known as evasion attacks), and attacks that happen during a test phase are poisoning attacks. The results of poisoning attacks are cautious for nervousness and incorrect viewpoints, discrimination in AI (but of course, it is not right due to the poisoned models or biased training), and more dangerous outcomes than evasion attacks, from the perspective of social influence. So, they introduced an effective real-time attack detection system which can not guarantee to evade the attacks 100 percent, but can point out the attacks in most of the cases. They concluded that their attack strategies and responses are still in their infancy and have the potential to lead to new developments in model verification, machine learning, and statistical learning theory. The defense methods against Data Poising attacks are presented.

3.2.3 Evasion Attacks

Evasion attacks mainly focus on manipulating certain features to confuse the ML classifiers via packet perturbations. Aparna et al. [5] studied the impact of evasion attacks that cause network intrusion. They presented a defense mechanism against network intrusion, especially in the age of the internet named NIDS (network intrusion detection system). In evasion attacks, the adversary initiates an attack by tampering with the packet's features to get past the ML-based NIDS. The accuracy of the ML-based NIDS can be reduced from 99+ percent to 0 percent by the adversaries using packet forging techniques to cause an ML classifier to inaccurately label attack packets as benign. The adversary with domain expertise and access to packet crafting techniques can

change packet headers and launch assaults on ML-based NIDS when we utilize ML classifiers in the normal manner, using all relevant characteristics in the dataset to train the model. They suggested using many smaller feature sets to find ML classifiers that, when used in an ensemble, would be more resistant to evasion attempts. Their practical assessments utilizing well-known datasets have demonstrated the effectiveness of the suggested ensemble classifier in recognizing many evasion methods that were missed by the conventional usage of a single ML classifier with the entire training dataset. Out of the literature that is under study, no other significant contribution is noticed which deals especially with evasion adversarial attacks which reiterates the potential of research and investigation in this area.

3.2.4 Random Forest Attacks

Salah et al. [6] shed light on the phenomena of random forest attacks. They take into account the technique put forth by Papernot et al. [12], hereafter known as the Papernot attack, which was initially intended to lead to misclassifications in decision trees by visiting every node in the tree and forcing them to flip until the misclassification was achieved. They modify this adversarial approach (against random forests) by repeatedly applying the Papernot attack to each forest tree until the random forest's categorization result changes. This approach is known as Iterative Papernot. On all of the initial test inputs from their use case where the model correctly classifies data, they tested Iterative Papernot. They showed that although their results are quite an improvement in creating the attacks but overall success rate is almost zero because of the domain constraints at the input level. They also explored another family of adversarial attacks called Gradient-based attacks. A DNN iteratively modifies the weights of its neurons as it learns in response to the gradient of its cost function, which is dependent on the weights. The same information is used in gradient-based attacks to create a perturbation that alters the last neuron layer's output and, in turn, the classification result. In both cases, they observed that creating examples of these adversarial attacks is not very useful due to different domain constraints. So, they introduced a search-based method to overcome these problems, and at the same time, they improved the defense system by exploiting the attack mechanism.

3.2.5 Prompt Injection Attacks

To answer RQ2, and to define prompt injection attacks, first, the definition of the target task and the injected task is important. The target task comprises an instruction and corresponding data. For example, in a spam-detection task, the instruction might be "Please indicate whether the following text is spam or non-spam," and the data would be an email. The user's objective is to complete the target task utilizing an LLM-Integrated Application.

Prompt injection instances include:

-Influencing LLMs to disregard system protections

-Removing private or sensitive financial information

-Submitting applications using prompts designed to fool the LLM into supporting applicants who are not eligible

-Utilizing APIs and plugins to facilitate illegal transactions

Researchers have shown that it is possible to modify LLM-integrated apps to produce biased or deceptive results, proving that prompt injection is not only a theoretical worry. Vulnerabilities have been documented in real-world incidents affecting many systems, such as Google's Bard AI, ChatGPT, and Bing Chat.

Attacks using prompt injection present a serious risk since they can harm millions of people and shape public perception and policy. To reduce hazards, strong defenses like bias-free prompting and training data filtering must be developed immediately. All things considered, quick injection exploits are a pressing threat that AI teams need to start tackling right away.

3.3 Threat Model

To explain the threat model, first, the understanding of the attacker needed to be realized. In the context of a security scenario, imagine a scenario where a malicious actor seeks to infiltrate an LLM-Integrated Application to manipulate the application's output to align with the attacker's preferences. For example, in a spam-detection application, the attacker may aim to coerce the LLM-Integrated Application into classifying a spam email as non-spam. This general scenario involves the attacker attempting to manipulate the LLM-Integrated Application to generate a specific, desired response for a user. It is presumed that the attacker possesses awareness that the application in question is an LLM-Integrated Application. However, beyond this acknowledgment, it is assumed that the attacker possesses minimal information about the application. Specifically, the attacker is presumed to lack knowledge regarding the instruction prompt used and the specific backend LLM employed by the application. It is contemplated that the attacker can manipulate the data prompt employed by the LLM-Integrated Application. More precisely, the assumption is that the attacker can introduce any desired instruction or data into the data prompt.

3.4 The Attack Framework

The attack framework comprises the combination of all the possible prompt attacks that can manipulate the LLMs. An attacker seeks to compromise the data prompt x t to cause the LLM-Integrated Application to complete an injected job, as defined by the concept of a prompt injection attack.

As all the definitions are possible attacks in prompts are described in Table 1, It is evident that these attacks can harm differently, and especially when the attacker's goal is not confirmed, it can harm the system in any possible way. So, combining these attacks like Naive Attacks [9], Escape Characters [9], Context Ignoring [1], Fake Completion [9], and Backdoor Attacks [14] can lead the way to develop a strong, novel threat model, which can lead to a strong defense model as well.

Attack	Description
Naive Attack	Concatenate target data, injected instruction, and injected data
Escape Characters	Adding special characters like "\n" and "\t".
Context Ignoring	Adding context-switching text to mislead the LLM that the context changes.
Fake Completion	Adding a response to the target task to mislead the LLM that the target task has completed.
Backdoor Attack	Backdoor with Virtual Prompt Injection (VPI)
Combined Attack (This Proposal)	Combining all the above attacks to develop a strong threat framework

Table 1: Description of prompt attacks

Combined Attack: Different prompt injection attacks fundamentally use different approaches to create harm under the proposed attack framework. This attack architecture makes it possible to create new prompt injection attacks in the future. Combining the four assault tactics mentioned above, for example, is a simple new attack that is motivated by this framework. In particular, our Combined Attack creates the compromised data prompt $\tilde{x}$ in the following ways, given the target data xt, injected instruction se, and injected data xe:

The equation is given by:

$$\tilde{x} = x_t \oplus c \oplus r \oplus b \oplus c \oplus i \oplus se \oplus xe$$

Here the special character c is used twice to explicitly separate the fake response r and the task-ignoring text i, and the character b represents the backdoor attacks.

4 Plans

This task of developing new threat models, especially in prompt learning in LLMs is trendy and it requires a constant effort to meet the requirements. So, the timeline for developing the proposed threat model and then the defense model is shown in Table 2.

Since it is quite a new approach, it will engage me for the rest of the time of my degree, and building a strong attack and defense system will be a great contribution in this regard. The timeline here is tentative and will be modified according to the nature of the results and overall research advancements in this field.

Task 1	Task 2	Timeline
Implementing different types of Prompt Attacks	Writing reports/ Paper	Jan-Jun 2024
Implementing Proposed Approach	Writing reports/ Paper	Jul-Dec 2024
Reviewing Results	Writing reports/ Paper	Jan 2025
Developing Defense Method against Attacks	Writing reports/ Paper	Feb-Jun 2025
Reviewing Results	Writing reports/ Paper	Jul 2025
Writing final Thesis		Jan 2025-Onwards

Table 2: The Tasks Timeline

References

[1] Hezekiah J Branch, Jonathan Rodriguez Cefalu, Jeremy McHugh, Leyla Hujer, Aditya Bahl, Daniel del Castillo Iglesias, Ron Heichman, and Ramesh Darwishi. Evaluating the susceptibility of pre-trained language models via handcrafted adversarial examples. *arXiv preprint arXiv:2209.02128*, 2022.

[2] Xiangrui Cai, Haidong Xu, Sihan Xu, Ying Zhang, et al. Badprompt: Backdoor attacks on continuous prompts. *Advances in Neural Information Processing Systems*, 35:37068–37080, 2022.

[3] Aochuan Chen, Peter Lorenz, Yuguang Yao, Pin-Yu Chen, and Sijia Liu. Visual prompting for adversarial robustness. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[4] Tran Khanh Dang, Phat T Tran Truong, and Pi To Tran. Data poisoning attack on deep neural network and some defense methods. In *2020 International Conference on Advanced Computing and Applications (ACOMP)*, pages 15–22. IEEE, 2020.

[5] Aparna Ganesan and Kamil Sarac. Mitigating evasion attacks on machine learning based nids systems in sdn. In *2021 IEEE 7th International Conference on Network Softwarization (NetSoft)*, pages 268–272. IEEE, 2021.

[6] Salah Ghamizi, Maxime Cordy, Martin Gubri, Mike Papadakis, Andrey Boystov, Yves Le Traon, and Anne Goujon. Search-based adversarial testing and improvement of constrained credit scoring systems. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1089–1100, 2020.

[7] Hyun Kwon, Hyunsoo Yoon, and Ki-Woong Park. Friendnet backdoor: indentifying backdoor attack that is safe for friendly deep neural network. In *Proceedings of the 3rd International Conference on Software Engineering and Information Management*, pages 53–57, 2020.

[8] Han Liu, Yuhao Wu, Shixuan Zhai, Bo Yuan, and Ning Zhang. Riatig: Reliable and imperceptible adversarial text-to-image generation with natural prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20585–20594, 2023.

[9] Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. Prompt injection attack against llm-integrated applications. *arXiv preprint arXiv:2306.05499*, 2023.

[10] Rohan Reddy Mekala, Adam Porter, and Mikael Lindvall. Metamorphic filtering of black-box adversarial attacks on multi-network face recognition models. In *proceedings of the IEEE/ACM 42nd international conference on software engineering workshops*, pages 410–417, 2020.

[11] Raphaël Millière. Adversarial attacks on image generation with made-up words. *arXiv preprint arXiv:2208.04135*, 2022.

[12] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.

[13] Lei Xu, Yangyi Chen, Ganqu Cui, Hongcheng Gao, and Zhiyuan Liu. Exploring the universal vulnerability of prompt-based learning paradigm. *arXiv preprint arXiv:2204.05239*, 2022.

[14] Jun Yan, Vikas Yadav, Shiyang Li, Lichang Chen, Zheng Tang, Hai Wang, Vijay Srinivasan, Xiang Ren, and Hongxia Jin. Backdooring instruction-tuned large language models with virtual prompt injection. In *NeurIPS 2023 Workshop on Backdoors in Deep Learning-The Good, the Bad, and the Ugly*, 2023.

[15] Yuting Yang, Pei Huang, Juan Cao, Jintao Li, Yun Lin, Jin Song Dong, Feifei Ma, and Jian Zhang. A prompting-based approach for adversarial example generation and robustness enhancement. *arXiv preprint arXiv:2203.10714*, 2022.

[16] Xiaoyan Yu, Qilei Yin, Zhixin Shi, and Yuru Ma. Improving the semantic consistency of textual adversarial attacks via prompt. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.