

LLM Security: Vulnerabilities, Attacks, Defenses, and Countermeasures

FRANCISCO AGUILERA-MARTÍNEZ and FERNANDO BERZAL, Department of Computer Science and Artificial Intelligence, University of Granada, Spain

As large language models (LLMs) continue to evolve, it is critical to assess the security threats and vulnerabilities that may arise both during their training phase and after models have been deployed. This survey seeks to define and categorize the various attacks targeting LLMs, distinguishing between those that occur during the training phase and those that affect already trained models. A thorough analysis of these attacks is presented, alongside an exploration of defense mechanisms designed to mitigate such threats. Defenses are classified into two primary categories: prevention-based and detection-based defenses. Furthermore, our survey summarizes possible attacks and their corresponding defense strategies. It also provides an evaluation of the effectiveness of the known defense mechanisms for the different security threats. Our survey aims to offer a structured framework for securing LLMs, while also identifying areas that require further research to improve and strengthen defenses against emerging security challenges.

CCS Concepts: • **Security and privacy** → **Malware and its mitigation**; • **Applied computing** → *Document management and text processing*; • **Computing methodologies** → **Natural language generation**; **Neural networks**.

Additional Key Words and Phrases: Artificial Intelligence, Artificial Neural Networks, Deep Learning, Natural Language Processing, Large Language Models, Security Threats, Defense Mechanisms

1 Introduction

Artificial intelligence (AI) has emerged to meet the demand for technologies that can emulate, and in some instances surpass, human cognitive capabilities [88]. The goal of AI is to create "agents that perceive and act upon the environment," replicating human and rational behavior in systems that can operate autonomously within complex, real-world contexts [110]. AI's ability to address intricate problems and perform tasks that require advanced reasoning, learning, perceptual processing, and decision-making continues to expand in both complexity and scope. A significant development in this field has been the rise of large language models (LLMs), which build on the success of deep learning in identifying complex patterns within large datasets [46] [12] [13], marking a substantial leap in natural language processing (NLP). However, machines still lack the inherent ability to understand, process, and communicate human language without sophisticated AI support [175]. The challenge remains to enable machines to acquire reading, writing, and communication skills akin to human proficiency [133].

LLMs [11] have made notable contributions to AI, enabling applications across diverse fields such as education [70], customer support [117], healthcare [56], or even scientific research [169] [4]. A major breakthrough in LLMs occurred with the introduction of the Transformer architecture [136], which leverages a self-attention mechanism to enable parallel processing and efficiently manage long-range dependencies. LLMs, from Google's BERT [34] to the different incarnations of OpenAI's GPT [97], have revolutionized AI-driven tasks by leveraging their ability to generate human-like language, from text generation and summarization [101] to sentiment analysis [127] [32] and machine translation [177] [152]. Trained on vast, unfiltered datasets scraped from the Internet, LLMs are proficient at handling a wide range of language-based tasks, from answering questions and summarizing documents to creative writing and automated coding assistance [107].

As the deployment of LLMs expands across various sectors, so too do the security threats and vulnerabilities associated with their use [2] [3] [41] [44] [79] [84] [86] [93] [115] [141] [148] [156] [162]. These models are particularly

Authors' Contact Information: Francisco Aguilera-Martínez, faguileramartinez@acm.org; Fernando Berzal, berzal@acm.org, Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain.

vulnerable to a variety of attacks, where even minor, intentional modifications to input data can drastically alter the model’s output, potentially leading to critical issues such as misrecognition errors and privacy breaches. Such attacks can occur both during the training phase and after the model has been fully trained. These vulnerabilities pose substantial risks not only to the reliability and safety of AI systems but also to the security and privacy of users interacting with these models.

Despite the growing recognition of these risks, comprehensive investigations into the vulnerabilities of LLMs—particularly those related to security and privacy—remain limited [31]. While these concerns have often been overlooked, addressing them is crucial to understand both the nature of the threats and the available defense strategies to mitigate them [129].

This survey paper includes an in-depth analysis of attacks on LLMs, addressing both those that occur during the model training phase and those targeting models after they have been deployed. We provide a detailed examination of these attacks, categorizing them on the basis of the stage of the LLM lifecycle they impact on. In addition, we evaluate current defense mechanisms, classifying them into prevention-based and detection-based defenses. A summary matches these defense mechanisms with the attacks they prevent or mitigate, assessing their effectiveness and highlighting areas where current approaches are most and least successful.

2 Background

2.1 Language models

A statistical model of language can be represented by the conditional probability of the next word given all the previous ones [11]. A neural probabilistic language model just employs neural networks for learning the probability function from training data. That function is typically decomposed into two parts: a mapping from a word/token in the model vocabulary to a real vector (distributed feature vector or word/token embedding) and the probability function over words/tokens, from an input sequence of feature vectors (context window) to a conditional probability distribution over words/tokens in the model vocabulary. Such a model can then be used to generate text by choosing the next word according to the predicted probabilities given the input context.

Current large language models (LLMs) have evolved from the original proposal of neural probabilistic language models. They are the largest artificial neural networks ever trained. They use the Transformer architecture, which enables them for handling sequential data efficiently and capturing long-range dependencies in text [136]. Transformers address the limitations of previous models based on recurrent neural networks through attention mechanisms, so that each token in the input sequence can interact with every other token in the sequence [124]. Their high computational cost and memory demands [29] have sparked research into alternative architectures that optimize both training and inference efficiency [62]. Recent advancements in language model architectures have focused on improving computational efficiency and memory use during inference, with strategies such as flexible encoder-decoder architectures and optimized pretraining to better balance performance and resource demands [142].

LLMs employ autoregressive token prediction to generate sequential text, maximizing the probability of each token by decomposing the conditional probability into a chain of token-based predictions [24, 52]. The neural networks behind LLMs learn probability distributions on tokens. Given an input text T_{input} as a sequence of tokens $\langle t_1..t_n \rangle$, the LLM neural network estimates the probability distribution for the $(n+1)$ -th token: $f(\langle t_1..t_n \rangle) = \hat{p}_{t_{n+1}}$. When used as a generative system, the LLM behaves as a ‘stochastic parrot’, choosing the next token in the sequence given by the input text: $f_{gen}(\langle t_1..t_n \rangle) = \hat{t}_{n+1}$. That token can then be added to the input text in order to generate more tokens, e.g.

$f_{gen}(\langle t_1..t_n\hat{t}_{n+1} \rangle) = \hat{t}_{n+2}$. Just by repeating the previous process, the LLM can generate complete texts: $F(\langle t_1..t_n \rangle) = \hat{t}_{n+1}..\hat{t}_{n+k}$, or $F(T_{input}) = \hat{T}_{gen}$, for short.

As mentioned before, LLMs operate through in-context learning [14]: the model is trained to generate coherent, contextually relevant text based on a given prompt. This is enhanced by Reinforcement Learning from Human Feedback (RLHF) [25], a process that fine-tunes the model using human responses as feedback, effectively improving its output quality over time [173]. When deployed, prompt engineering is widely used to guide LLMs towards producing specific, desired responses, expanding their utility across a range of tasks and applications [95].

LLMs should possess at least four key abilities [159]. First, they should be able to exhibit a deep understanding and interpretation of natural language text, enabling them to extract information and perform various language-related tasks (e.g., translation). Second, they should be able to generate human-like text when prompted. Third, they should exhibit contextual awareness by considering factors such as domain expertise for knowledge-intensive tasks. Fourth, these models should excel in problem-solving and decision-making by leveraging information within text passages. These abilities can make them invaluable for information retrieval [178] and question-answering systems [114], among many other language-related applications.

The evolutionary tree of modern LLMs traces the development of language models in recent years [159]. There is even a chess-like Elo ranking system where LLM models are evaluated through human preference [27]. Some of the top performers in this ranking include the following well-known players in the LLM industry:

- *GPT (OpenAI)*: Generative Pre-trained Transformers (GPT) introduced a semi-supervised training methodology that combines unsupervised pre-training with supervised fine-tuning, allowing the model to leverage large corpora of unlabeled text alongside smaller, annotated datasets [106] [163] [74]. GPT key components include input embedding layers, positional encoding, Transformer blocks, and linear and softmax functions to generate probability distributions. GPT-3.5 had around 175B parameters [17], yet GPT-4 is not a single massive model, but rather a combination of 8 smaller models, each consisting of 220B parameters (i.e. a mixture-of-experts, MoE [53]).
- *Llama (Meta)*: Llama models are distinguished by being open source. They aim to achieve state-of-the-art results without the need for proprietary datasets, thereby offering an open alternative for research and industry applications in natural language processing. Llama 2 models [131] range in scale from 7 billion to 70 billion parameters. Llama 3 [48] largest model is a dense Transformer with 405B parameters and a context window of up to 128K tokens. Llama herd of language models natively support multilinguality, coding, reasoning, and tool usage.
- *Gemini (Google)*: Gemini models leverage a modified decoder-only Transformer architecture optimized for efficiency on Google's Tensor Processing Units (TPUs) [128]. The Gemini model family offers multimodal capabilities across text, images, audio, and video inputs. Gemini incorporates innovations such as multi-query attention, the Lion optimizer, and Flash Decoding, enhancing training stability and inference speed on TPUv4 and TPUv5e. Additionally, Gemini integrates Retrieval-Augmented Generation (RAG) [75], which improves the relevance and factual grounding of its outputs by retrieving pertinent external information based on cosine similarity and indexing methods [64].
- *Grok (xAI)*: Grok [150] is another autoregressive Transformer-based model pre-trained to perform next-token prediction based on a mixture-of-experts (MoE) architecture. The initial Grok-1 has 314 billion parameters

and a context length of 8,192 tokens and has been open sourced by xAI under the Apache 2.0 license. Building on Grok-1’s foundational design, Grok-2 introduced significant improvements, particularly its multimodal capabilities and integration with real-time data from the X platform.

- *Claude (Anthropic)*: The Claude models incorporate techniques such as hierarchical representations and residual connections to optimize training and inference. Claude 3.5 [7] was trained using curriculum learning and data augmentation, with bias mitigation mechanisms and self-checking capabilities intended to prioritize safety and alignment.

2.2 Security issues in language models

AI safety is focused on preventing accidents, misuse, or other harmful consequences arising from AI systems. It encompasses machine ethics and AI alignment, which aim to ensure AI systems are moral and beneficial, as well as monitoring AI systems for risks and enhancing their reliability. AI safety research areas include robustness, monitoring, and alignment.

AI systems are often vulnerable to adversarial examples [47], inputs to machine learning (ML) models that an attacker has intentionally designed to cause the model to make a mistake. Even imperceptible perturbations to an input image could cause it to be misclassified with high confidence [126]. Adversarial robustness is often associated with security.

Machine learning models, including neural networks, can misclassify adversarial examples—inputs formed by applying small but intentionally worst-case perturbations to examples. In the case of neural networks, for instance, neural network parameters are trained by estimating the gradient of the cost function with respect to their parameters (a.k.a. weights). Adversarial examples can be designed using the same mechanism that is used to train them, just by taking into account the gradient of the cost function with respect to the input.

A qualitative taxonomy can be used to categorize attacks against machine learning systems [68]. One of its dimensions describes the capability of the attacker, his influence [10]: whether (a) the attacker has the ability to influence the training data that is used to construct the model (a causative attack) or (b) the attacker does not influence the learned model, but can send new instances to the classifier and possibly observe its decisions on these carefully crafted instances (an exploratory attack). Causative attacks occur during model training, while exploratory attacks occur once models are deployed. We do not address attacks on the model themselves, such as model stealing attacks [96], when the attacker’s goal is getting access to the trained model parameters or its optimized hyperparameters.

The widespread use of LLMs has exposed a series of vulnerabilities to potential malicious attacks. Both causative and exploratory attacks present security risks with a significant impact in terms of both integrity and privacy.

The training of LLMs typically depends on large, uncurated datasets (often the whole Internet). This reliance makes them susceptible to biases and the inadvertent inclusion of sensitive information [17] but also leaves LLMs vulnerable to more malicious forms of data manipulation, such as backdoor attacks [60]. During pre-training, such attacks can be embedded by introducing hidden triggers in the model’s weights, which remain dormant until specific inputs are encountered [50]. These issues are compounded by the ethical and misinformation risks inherent in LLMs, as their dependence on unverified data sources often leads to the spread of false information, particularly concerning in sensitive cultural, gender, racial, educational, professional, healthcare, legal, and law enforcement contexts [85].

Once LLMs are deployed, attackers might try to manipulate their input. In another common attack, known as prompt injection [81], subtle prompt alterations can override the model’s safeguards and provoke inappropriate or harmful outputs [82]. LLMs are also vulnerable to other security threats [2] including membership inference attacks (MIAs)

that exploit a model’s tendency to retain traces of training data, allowing adversaries to identify specific sensitive data points within the dataset and posing severe privacy concerns [21] [76].

In summary, LLMs can unintentionally produce outputs that reveal confidential data or provide intentionally misguided information. It is crucial to identify potential attacks on LLM-based systems, available defensive countermeasures, and containment strategies to mitigate the potential damage attacks can inflict on LLM-based systems. Research on this area contributes to the development of more robust and secure LLMs, capable of withstanding malicious attacks and protecting the privacy of the data they were trained with.

3 Attacks on language models

This Section describes the different kinds of attacks that can occur during the life cycle of a language model. We categorize LLM attacks into two broad classes: those that occur during model training (causative or training-time attacks) and those that target an already trained model (exploratory, test-time, or inference-time attacks).

3.1 Attacks during model training

During their training phase, LLMs are particularly vulnerable to certain types of attacks that can compromise their integrity and functionality. These attacks can manipulate the training data and even the training process itself in order to induce malicious behavior or degrade the performance of LLMs. Most often, adversarial attacks intentionally tamper with the training data by introducing fudged or malicious data to deceive and confuse the trained models, so that they will later produce incorrect outputs.

Causative attacks, those performed during model training, include backdoor, data poisoning, and gradient leakage attacks.

3.1.1 Backdoor attacks. Backdoor attacks exploit vulnerabilities by embedding hidden triggers in models, allowing normal outputs for standard inputs but causing malicious behavior when exposed to attacker-specified patterns. In NLP tasks, the trigger can be a single token, a specific character, or a sentence, and the goal is to cause misclassifications or generate incorrect text [137].

Let F denote a target model, which will now be trained on a modified dataset $D_{backdoor} = D \cup D^*$, where D is the clean training dataset and $D^* = \{(x^*, y^*)\}$ consists of triggered instances generated by applying a style transfer function T to clean instances x , mapping them to the trigger style $x^* = T(x)$ with target label y^* [102].

The goal of training on $D_{backdoor}$ is to produce a backdoored model F^* that behaves as follows. For clean samples, $F^*(x) = y$, maintaining its expected behavior. For triggered samples, the backdoor effect is activated when the input contains the trigger: $F^*(x^*) = y^*$, where y^* is the output desired by the attacker for style-transferred instances x^* .

Backdoor attacks expose risks in the ML supply chain, as compromised models can behave malevolently under specific conditions without detection [50]. The attacker manipulates the target model by poisoning its training data, causing it to achieve the desired goal when a specific trigger appears in the input data, while functioning normally with clean data [60]. During pre-training, such attacks can be embedded by introducing hidden triggers in the model weights, which remain dormant until specific inputs are encountered [50].

Depending on what triggers the backdoor attack on LLMs, backdoor attacks can be classified into four types [158]:

- *Input-triggered attacks* [77] [160] [78] [167] insert backdoors into the target model by maliciously modifying the training data during the pre-training stage. One prominent method, PTM [77], uses a combination of characters as triggers and a weight poisoning technique at specific layers of the model to make the early layers sensitive to

poisoned data. The word embedding vector of the trigger word plays a significant role in the poisoned model’s final decision [160]. Stealthy backdoor attacks may require more sophisticated manipulation of input data [78].

- *Prompt-triggered attacks* [18] [98] [174] involve the malicious modification of the prompt to inject a trigger, or the compromise of the prompt through malicious user input. BadPrompt [18] learns adaptive triggers for targeted attacks, but requires significant computational resources. Malicious user inputs can change the even leak the model’s prompt [98]. Clean-label backdoor attacks [174] are harder to detect.
- *Instruction-triggered attacks* [153] are performed by contributing poisoned instructions via crowd-sourcing to misdirect instruction-tuned models. This kind of attacks do not target the uncurated training data, but instead focus on the LLM fine-tuning process.
- *Demonstration-triggered attacks* [140] are subtler and harder to detect. In-context learning (ICL) has gained prominence by utilizing data-label pairs as precondition prompts. While incorporating those examples, known as demonstrations, can greatly enhance the performance of LLMs across various tasks, it may introduce a new security concern: attackers can manipulate only the demonstrations without changing the input to perform their attack.

3.1.2 Data poisoning attacks. In data poisoning attacks [130] [144] [22], attackers inject maliciously crafted data into the training set. Poisoning can also be performed during model alignment or fine tuning, since instruction-tuned LMs such as ChatGPT and InstructGPT are fine-tuned on datasets that contain user-submitted examples [139].

Let F denote a target model, which will now be trained on a modified dataset $D_{poisoned} = D^*$, where D^* is a surreptitiously modified version of the clean training dataset D . The aim of data poisoning F is creating a poisoned model F^* that makes incorrect predictions, often without an observable degradation in its overall accuracy. Data poisoning compromises the model integrity by introducing systematic biases that serve the attacker’s objectives while evading detection during model training.

Attackers can introduce manipulated data samples when training data is collected from unverified external sources. These poisoned examples, which contain specific trigger phrases, allow adversaries to induce systemic errors in LLMs [31]. A compromised LLM trained with poisoned data can present a great risk of spreading misinformation and causes serious implications on downstream tasks [30].

Data poisoning attacks can also be performed by split-view or front-running poisoning [20] [129]:

- *Split-view poisoning:* By altering content after it has been initially indexed, attackers can ensure that what is included in the training dataset differs from the original data. Training LLMs on large-scale datasets downloaded from the Internet poses a significant risk, given the dynamic nature of Internet resources.
- *Front-running poisoning:* targets web-scale datasets that periodically snapshot crowd-sourced content—such as Wikipedia—where an attacker only needs a time-limited window to inject malicious examples. In datasets that capture user-generated or crowd-sourced content at regular intervals, attackers can time malicious changes to appear during these snapshots and revert them to avoid detection. This way, attackers can insert poisoning samples into training datasets without maintaining long-term control over their content.

3.1.3 Gradient leakage attacks. Gradients are vectors that indicate the direction in which model parameters should be tuned to minimize the loss function during training. Exchanging gradients is a common operation when training modern multi-node deep learning systems, including LLMs. For a long time, gradients were believed to be safe to share.

However, private training can be leaked by publicly shared gradients [176]. An attacker can reconstruct sensitive input data by exploiting gradients exchanged during LLM training.

In distributed ML training, work is shared among many nodes, typically equipped with GPUs. In a centralized scenario, a parameter server receives gradients from worker nodes and the parameter server would be the target of the attacker. In a fully-distributed scenario, where gradients are freely exchanged, any participant node can maliciously steal data from its neighbors.

To recover training data from gradients, the attacker aligns leaked gradients with dummy inputs. The DLG algorithm [176] is based on minimizing the difference between the gradients produced by dummy data and the gradients of real data. Using a randomly initialized dummy input-output pair (x', y') , we feed these dummy data into the model to obtain dummy gradients $\nabla L(F(x'), y')$. Optimizing the dummy gradients to be close to the original also makes the dummy data close to the real training data. Given the actual gradients $\nabla L(F(x), y)$, we obtain the training data by solving the following optimization problem:

$$x^*, y^* = \arg \min_{x, y} \|\nabla L(F(x'), y') - \nabla L(F(x), y)\|^2$$

Further developments improved the reconstruction quality by aligning the gradient directions [45] rather than just minimizing Euclidean distances. In this situation, the function to be optimized is defined in terms of the cosine similarity between the gradients, plus a total variation regularization term [109]:

$$x^*, y^* = \arg \min_{x, y} \left(1 - \frac{\langle \nabla L(F(x), y), \nabla L(F(x'), y') \rangle}{\|\nabla L(F(x), y)\| \|\nabla L(F(x'), y')\|} \right) + \alpha \text{TV}(x)$$

Gradient leakage attacks compromise the privacy of training data. This attack is particularly relevant in federated learning contexts [23], where gradients are shared among multiple parties for model updating without the original data being directly disclosed. It has also been found to work on LLMs [51], where recent research results, with attacks such as TAG [33] or LAMP [9], have shown that it is possible to reconstruct private training data with high accuracy [31].

Gradient leakage attacks can be auxiliary-free or auxiliary-based [157]. In auxiliary free attacks, the typical scenario, the attacker has information only about the model parameters and the gradients of the participants. In auxiliary-based attacks, the attacker might have additional information at his disposal, such as auxiliary datasets, statistics of the global model, or pretrained generative adversarial networks. Any additional information can help him improve the accuracy of his private data reconstruction.

State-of-the-art gradient leakage attacks can be optimization-based or analytics-based [157]. Optimization-based attacks, such as the ones described above, try to reconstruct data by adjusting gradients to resemble the original data. Analysis-based attacks use systems of linear equations to extract data faster and more accurately.

3.2 Attacks on trained models

Once a language model has been trained and is deployed in actual systems, the LLM remains vulnerable to a variety of attacks that can exploit its inherent weaknesses. Attackers can manipulate the LLM input to induce incorrect behavior or extract sensitive information, even to infer the original training data.

The most prominent exploratory attacks, performed once the LLM has been trained and deployed in production, include adversarial input attacks, prompt hacking, data extraction, and membership inference attacks, as well as different flavors of inversion attacks (namely, model and embedding inversion attacks).

3.2.1 Adversarial input attacks. Machine learning models can be intentionally deceived through adversarial inputs [126], also known as test-time evasion (TTE) attacks [15]. These involve crafting input examples designed to produce unexpected outputs. Traditionally, these attacks employ carefully crafted noise in the direction of the loss gradient to maximize its impact on the network’s loss. By backpropagating the loss to the input layer, the inputs are adjusted in alignment with the gradient to create adversarial examples. To remain subtle and undetectable, attackers typically operate within a constrained noise budget [126]. Following the loss gradient, small perturbations can cause a large change in the model output, allowing adversarials to achieve their goal. This vulnerability was initially attributed to extreme nonlinearity and overfitting, yet the primary cause of neural networks’ vulnerability to adversarial perturbation was found to be their linear nature [47].

As any other machine learning models, LLMs are also vulnerable to adversarial input attacks. An adversary crafts targeted input prompts designed to manipulate the model behavior and induce undesirable or malicious outputs.

Adversarial input attacks can be either targeted or untargeted. Untargeted attacks just try to cause an incorrect output, while targeted attacks attempt to force the output to include a specific text chosen by the attacker. In a targeted attack, we are given a model F , an input text T , and a target y_{target} . The adversary concatenates trigger tokens $T_{trigger}$ to the front or end of T so that $F(T_{trigger}T) = y_{target}$.

It is known that LLMs are vulnerable to *universal triggers* [137]—specific phrases into the training data that cause targeted output responses across multiple tasks, exposing a fundamental weakness in LLM robustness. In a universal trigger attack, the adversary optimizes T_{adv} to minimize the loss for the target for all inputs from a dataset, $\arg \min_{T_{adv}} E(L(y, F(t_{adv}t)))$.

For LLMs, threat models must also account for the unique challenges posed by their input spaces, such as *embedding space attacks* [113]. To perform an embedding space attack, the input string is passed through the tokenizer and embedding layer of the LLM. The attacker optimizes some subset of the user prompt to maximize the probability of the desired response by the LLM. This optimization is performed over all continuous token embeddings at once, as opposed to one token at a time [180]. The resulting out-of-distribution embeddings do not correspond to actual words.

LLMs are vulnerable to adversarial prompts that aim to induce specific harmful or malicious behaviors, even in the absence of explicit “toxic” instructions. The risk of adversarial manipulation remains high, particularly when attackers are able to exploit multimodal capabilities [103] or access the embedding space [113] directly.

3.2.2 Prompt hacking. A prompt hacking attack refers to a specific type of adversarial input attack on LLMs. Prompt hacking attacks exploit vulnerabilities in the model alignment—the process of configuring LLMs to ensure that their outputs conform to ethical standards, safety protocols, and application-specific guidelines. By exploiting these alignment weaknesses, prompt hacking seeks to generate content that violates intended usage policies, potentially resulting in harmful, biased, or inappropriate responses. The goal is to bypass LLMs built-in safeguards, leading to outputs that contravene ethical guidelines or provide deceptive or dangerous information. There are different types of prompt hacking attacks [105]:

- *Jailbreaking attacks* [146] [116] attempt to bypass the LLM alignment to produce restricted content by manipulating the input prompt. Jailbreaking makes the LLM behave in ways it is supposed to avoid, such as generating inappropriate content, disclosing sensitive information, or performing actions contrary to predefined constraints.

Most jailbreak attacks are carried out by creating “jailbreak instructions” [28]. To guide the generation of this content, the attacker is provided with a target string G , which requests an objectionable response and to which

an aligned LLM would likely refrain from responding [108]. LLMs typically avoid such responses thanks to built-in security measures during their training, such as Reinforcement Learning with Human Feedback (RLHF) [179], Robustness through Additional Fine-tuning or Reward rAnked FineTuning (RAFT) [35], and Preference-Optimized Ranking (PRO) [122]. However, the exact mechanisms behind jailbreaking are still debated. Research suggests that jailbreaks can occur in areas not fully covered by security training or when the model faces conflicts between providing useful information and following security protocols. Furthermore, it has been found that certain suffixes added to the original instructions can lead models towards generating inappropriate content [155]. Some attacks [111] can jailbreak aligned LMs with high attack success rates within one minute.

- *Prompt injection attacks* [81] override the original prompts by using untrusted input to produce undesired or malicious output. As in other well-known injection attacks (e.g. SQL injection), injection lies in creating a prompt that makes the LLM-based application unable to distinguish between the developer’s instructions and the user input. The adversary takes advantage of the system architecture to bypass security measures and compromise the integrity of the application.

“Prompt” is a synonym for instruction (or in some cases, the combination of instruction and data), not just data. A prompt injection attack introduces instructions (or a combination of instructions plus data) from the injected task into the data of the target task. Formally, given an instruction prompt s_t (target instruction) and target data x_t , an attacker crafts compromised data \tilde{x} using a prompt injection attack A , so that $\tilde{x} = A(x_t, s_e, x_e)$, where s_e and x_e represent the instruction and data of the injected task. Under this attack, the LLM-based application queries the backend LLM F with the altered prompt $p = s_t \oplus \tilde{x}$, resulting in a response aligned with the injected task rather than the target task [82].

There are two kinds of prompt injection attacks: direct and indirect prompt injection. Direct prompt injection feeds the malicious prompt directly to the LLM (e.g. “ignore the above instructions and...”). Indirect prompt injection embeds malicious prompts in the data that LLMs consume (e.g. within a webpage that the LLM reads) [49].

As any interpreters, LLMs themselves cannot differentiate which parts of their input are instructions from authorized users and which are malicious commands from third parties (which are often mixed and sent to the LLM) [125], so prompt injection attacks pose a significant vulnerability in LLM-based applications.

- *Prompt leaking attacks* [168] try to extract the system prompt by carefully crafting prompts that reveal the original system prompt. System prompt might contain sensitive or confidential information, including proprietary algorithms, custom instructions, or intellectual property. However, system prompts should not be seen as secrets, since prompt-based services are vulnerable to simple high-precision extraction attacks.

Given that next-token probabilities contain a surprising amount of information about the preceding text (and the probability vector can be recovered through search even without predictions for every token in the vocabulary) [92], you can recover text hidden from the user (i.e. unknown prompts) given only the current model output distribution.

3.2.3 Model inversion attacks. Model inversion attacks [43] can exploit confidence information from ML models to infer sensitive attributes of training data, highlighting a vulnerability in systems that expose detailed output probabilities. Generative models can be used to reconstruct private information by exploiting learned representations in neural networks [170]. Model inversion attacks also known as data reconstruction attacks [156]. In LLMs, model inversion

attacks analyze model outputs, gradients, and/or parameters in order to infer sensitive details about their training data [123].

A formal game-based framework can be used to characterize model inversion attacks [149]. An attacker attempts to infer sensitive attributes z of input x from the output $F(x)$ of a model F trained on data D . The attack effectiveness is quantified by the probability $G = P[A(F(x)) = z]$, where A represents the attack strategy.

In black-box model inversion attacks, the adversary infers sensitive values with only oracle access to the model. In white-box model inversion attacks, the adversary has some additional knowledge about the structure of a model, its parameters, or its gradients.

By leveraging insights from the available information, the attacker aims to reverse-engineer the underlying model, potentially exposing private or confidential information embedded within the model [120]. Adversaries can retrieve specific training examples through targeted training data extraction techniques [21]. Text Revealer [166], for instance, shows how to reconstruct private texts from transformer-based text classification models.

An intriguing phenomenon, called “invertibility interference,” [149] can be used to convert a highly invertible model into a highly non-invertible model just by adding some noise.

3.2.4 Data extraction attacks. Data extraction attacks exploit publicly accessible prediction APIs [132]. These attacks try to replicate a model functionality without requiring access to its internal parameters or training data. An adversary with black-box access to a model F queries it with strategically-crafted inputs to gather outputs, which may include confidence scores. His ultimate goal is to approximate the decision boundaries or learn the internal logic of the target model F , effectively reconstructing a high-fidelity surrogate model \hat{F} .

In LLMs, data extraction attacks aim to extract memorized text instances and can lead to various privacy violations [16]. LLM models may inadvertently capture and replay sensitive information found in training data, raising privacy concerns during the text generation process. Key issues include unintentional data memorization, information leakage, and potential disclosure of sensitive data or personally identifiable information.

Data extraction attacks can be classified into two categories: untargeted attacks and targeted attacks [119]. Untargeted attacks attempt to retrieve any memorized text instance [21], while targeted attacks seek to retrieve a suffix for a given prefix that the adversary has access to. For example, an attacker can complete a private email or a specific phone number using a known prefix [5]. This type of attack has gained significant attention [90]. In “discoverable memorization” [19] [61] [71], the model is prompted with a portion of a sentence from the training data to extract the rest, thus enabling the adversary to perform targeted attacks. In “extractable memorization” [69] [94] [104], the adversary attempts to extract any information about the training data (an untargeted attack).

3.2.5 Membership inference attacks. Membership inference attacks (MIAs) focus on determining whether a language model has been trained with a specific set of data [118] [58] [59] [8]. These attacks typically exploit model overfitting in the model, where training data points are assigned higher confidence scores, enabling the adversary to infer membership by ranking predicted confidence levels [55]. Overfitting amplifies privacy vulnerabilities by establishing a direct relationship between the model capacity to memorize its training data and the risk of information leakage. Overfitted models disproportionately expose training data and adversaries exploit this memorization to determine whether specific examples were part of the training set [164]. In fact, adversaries can emit black-box queries to identify patterns in how text-generation models replicate training data, even when the model appears to generalize well [121].

Formally, given a language model M , a training dataset D_{train} , and a sample x , the goal of an MIA is to predict whether x belongs to D_{train} or not. This can be expressed as a decision function $f(x, M)$ that classifies x as either

“member” or “non-member” of D_{train} based on the model output when fed with x . A successful attack maximizes the probability of correct classification, i.e., $f(x, M) = 1$ when $x \in D_{\text{train}}$ and $f(x, M) = 0$ when $x \notin D_{\text{train}}$, indicating that the model has “leaked” information about its training data.

LLMs inherently memorize portions of their training data, making them susceptible to privacy attacks. The extensive capacity of LLMs to generate text increases the risk of unintentionally exposing sensitive training data [76]. Modern LLMs, trained with billions of tokens, are challenging for MIAs due to their high generalization ability, which decreases the likelihood of memorizing specific samples. The difficulty of these attacks increases due to the high n-gram overlap between training and “non-member” data, making it harder to distinguish between the two [36].

As model inversion attacks, MIAs can be performed as black-box or white-box attacks. In black-box settings, attackers rely solely on querying the model and observing its outputs, without access to its internal parameters, hyperparameters, or architecture. In contrast, the white-box scenario assumes access to the model internal components, which are readily available for open-source models and might result from data breaches or deployment vulnerabilities in the case of proprietary models.

3.2.6 Embedding inversion attacks. Text embedding models map text to vectors. They capture semantics and other important features of the input text. Embedding inversion attacks target the front end of LLMs. They aim to recover the original text input x^* from its text embedding $\varphi(x^*)$ by exploiting access to the embedding function φ of the LLM.

Typically, embedding inversion aims to recover the original text x from its embedding $e = \varphi(x)$ by maximizing the cosine similarity between the embedding of a candidate text \hat{x} and e .

In white-box scenarios, the attacker optimizes the difference between the target embedding and candidate inputs using continuous relaxation methods for efficient gradient-based inversion. In black-box settings, the attacker trains an inversion model using auxiliary data to predict the set of words in x^* , employing techniques like multi-label classification or multi-set prediction to recover the original words without their exact order [120]. Assuming that collisions are rare, the attacker queries the model with candidate texts until he finds the text \hat{x} whose embedding closely matches e [91].

Malicious actors with access to the vector corresponding to some text embedding, as well as the API of an Embedding as a Service (EaaS) platform, can train an external model to approximate an inversion function and thus reconstruct the original text from its representation in the form of text embedding.

There are also multilingual embedding inversion attacks that target multilingual LLMs [26].

4 Defenses against attacks on language models

As we discussed in the previous Section, LLMs are vulnerable to several types of attacks. Various defense strategies have been developed to mitigate these risks and protect both sensitive training data and the integrity of the models. As in any other ML system, these defenses try to balance the need to maintain the performance and functionality of LLMs with the need to ensure their security and privacy [135].

In this Section, we survey existing defenses against adversarial attacks on LLMs. In general, any ML system can suffer both causative and exploratory attacks [10]. Regularization techniques can be useful to constrain what the ML system learns and, therefore, to increase its robustness against causative attacks. For exploratory attacks, we cannot expect the learning algorithm to be secret, yet disinformation to confuse attackers and information hiding (e.g. preventing the adversary from discovering the ML system hyperparameters) might raise the bar for attackers to succeed. It should also be noted that targeted attacks are more sensitive than indiscriminate attacks. For the latter, randomization can be

effective: the adversary obtains imperfect feedback from the ML system, so more work is required for the attack to be successful, at the potential cost of decreasing the ML system performance.

Given that LLM uses are varied and complex, and considering how learning algorithms use and interpret training data, specific solutions are often required. Customized defenses must be based on a deep understanding of the LLM training and inference process, and will vary depending on the particular context and the potential attack vectors [129].

Defense mechanisms against LLM attacks can be divided into two main categories: prevention and detection.

4.1 Prevention-based defenses

Preventive defenses focus on redesigning the instruction prompt or preprocessing the data so that the model can perform its task even when the data has been compromised [79]. These strategies focus on restructuring inputs and enhancing model robustness through mechanisms that isolate malicious elements, refine data representation, and fortify alignment with intended behaviors. Various approaches have been developed to address vulnerabilities in LLMs [82].

4.1.1 Paraphrasing. Data-driven techniques for generating paraphrases are useful for modifying inputs while preserving the original meaning, particularly in the context of adversarial defense strategies [87]. These methods have been explored extensively, leveraging generative frameworks to encode and transform text into alternative representations while maintaining semantic fidelity [89]. Recent advances have extended the application of paraphrasing techniques to LLMs for addressing adversarial prompts. Paraphrasing, as a preprocessing step, transforms potentially harmful inputs into benign harmless ones. The generative model would accurately preserve natural instructions but would not reproduce an adversarial sequence of tokens accurately enough to maintain adversarial behavior [65]. By rephrasing the adversarial instructions, this mechanism disrupts the structured patterns that adversaries rely on to exploit model vulnerabilities, while retaining the core intent of legitimate instructions. Empirical evaluations have shown that paraphrasing can effectively neutralize adversarial behavior in various settings, albeit with some trade-offs in accuracy and performance when applied to benign inputs flagged erroneously by adversarial detectors [72].

4.1.2 Retokenization. Retokenization consists of breaking up tokens in a prompt and representing them using multiple smaller tokens. As happened with paraphrasing, retokenization disrupts suspicious adversarial prompts without significantly degrading or altering the behavior of the model in case the prompt is benign.

BPE-dropout [100] can be used to perform retokenization. High-frequency words are kept intact in the text, while rare words are broken down into multiple tokens. This is achieved by randomly dropping a percentage of BPE (Byte Pair Encoding) merges during text tokenization, which results in a random tokenization with more tokens than a standard representation [65].

Retokenization is effective against jailbreaking and prompt injection attacks, since it alters the structure of the input text so that malicious instructions become less effective.

4.1.3 Delimiters (e.g. spotlighting & prompt data isolation). Delimiters can be introduced to separate data from instructions within the input prompt. Delimiters help the model focus on the intended instructions while ignoring compromised content [154]. As in query parameterization for preventing SQL injection attacks in relational databases, the use of specific delimiters, such as guillemets (« ») or triple single quotes (""), can help LLMs interpret the text between them as data, rather than valid prompt instructions, thus protecting against malicious prompt injections. By isolating

[80] or spotlighting [57] data with the help of delimiters, the model can focus on the intended task without executing harmful instructions embedded within the injected user-provided data. When delimiters are successful, they prevent prompt injection attacks from succeeding.

4.1.4 Sandwich prevention. Sandwich prevention involves constructing a carefully framed prompt where safe contextual instructions are placed both before and after the potentially harmful or adversarial input. By surrounding the compromised data with these "safety buffers," the model is guided to realign its behavior with the intended task and to resist adversarial manipulations embedded within the user-provided prompt. This sandwich is intended to reinforce the safety mechanisms of the LLM, ensuring that it adheres to the target task and reducing the likelihood of generating harmful or inappropriate responses. Sandwich prevention addresses weaknesses such as sensitivity to token length, multilingual vulnerabilities, and contextual confusion by systematically employing safety-focused prompts [134].

4.1.5 Instructional prevention. Instructional prevention focuses on explicitly reinforcing the importance of following the original prompt and not paying attention to possible external manipulations. This acts as a clear directive for the LLM to ignore any adversarial instructions contained in the compromised data. It consists of just instructing the LLM to ignore any instructions contained in the user-provided text and to adhere exclusively to the application-specific instruction prompt [112], a weaker form of spotlighting [57].

4.1.6 Embedding purification. Embedding purification targets potential backdoors in word embeddings, detecting differences between pre-trained weights and compromised weights. It refines these embeddings to ensure they do not contain malicious triggers, thereby enhancing the model security and integrity.

For each word w_i , let f_i represent its frequency in a large-scale clean corpus, and f'_i its frequency in a poisoned dataset. $\delta_i = E_{i,\text{backdoored}} - E_{i,\text{pretrained}}$ is the embedding difference vector of dimension n . Under the assumption that $f'_i \approx C f_i$ for clean words, where C is a constant, $\|\delta_i\|_2 \propto \log(f_i)$ for clean words, whereas for trigger words $\frac{\|\delta_k\|_2}{\log(f_k)} \gg \frac{\|\delta_i\|_2}{\log(f_i)}$. This disproportionate discrepancy for trigger words allows the identification of compromised embeddings. Embedding purification [171] then resets the embeddings of such words (e.g., the top 200 ranked by $\frac{\|\delta_i\|_2}{\log(\max(f_i, 20))}$) to their pre-trained values while preserving other embeddings.

4.1.7 SmoothLLM. SmoothLLM [108] was designed against jailbreaking attacks, which aim to manipulate the model into generating inappropriate or undesirable content. SmoothLLM is based on a randomized smoothing technique that introduces small random perturbations into multiple copies of the original input and then aggregates the responses generated by each perturbed copy. This allows for the detection and mitigation of adversarial inputs by identifying unusual patterns in the aggregated responses. SmoothLLM may, however, result in a slight decrease in model performance, but this reduction is moderate and can be adjusted by selecting the appropriate hyperparameters. This method does not require retraining the underlying model and is compatible with both black-box and white-box models.

Let a prompt P and a distribution $P_q(P)$ over perturbed copies of P be given. Let $\gamma \in [0, 1]$ and Q_1, \dots, Q_N be drawn i.i.d. from $P_q(P)$. Let us now define V to be the majority vote of the jailbreaking function JB across these perturbed prompts with respect to the margin γ , i.e., $V = \mathbb{I}\left(\frac{1}{N} \sum_{j=1}^N (JB \circ LLM)(Q_j) > \gamma\right)$. The JB function is a binary-valued function that checks whether a response generated by an LLM constitutes a jailbreak. Then, SMOOTHLLM is defined as $\text{SMOOTHLLM}(P) = LLM(Q)$, where Q is any of the sampled prompts that agrees with the majority, i.e., $(JB \circ LLM)(Q) = V$.

4.1.8 Dimensional masking. Dimensional masking [26] is a defense technique against embedding inversion attacks. This method tries to protect the information encoded in the embeddings by masking part of their content to hinder attackers from reconstructing the original text from the embeddings. In dimensional masking, the embeddings generated by a language model are modified by adding a language identification vector, i.e. is a vector that represents the language of the original text. By masking the first dimension of the embedding with this language identification vector, it becomes more difficult for an attacker to invert the embeddings and retrieve the original text.

For an input text x , the masked embedding function ϕ_{masking} is defined as $\phi_{\text{masking}}(x) = \text{vec}([\text{id}_t, \text{vec}(\phi_i(x))_{1 \leq i \leq n}])$, where $\phi(x) = \text{vec}(\phi_i(x))_{0 \leq i \leq n}$ is the original embedding, $n \in \mathbb{N}$ is the dimensionality of $\phi(x)$, $\text{id}_t \in \mathbb{R}$ is a unique identifier encoding the target language l_t , and $\text{vec}(\cdot)$ denotes vectorization. This simple approach effectively reduces the success rate of embedding inversion attacks while fully preserving utility in retrieval tasks [26].

4.1.9 Differential privacy. Differential privacy (DP) originates from the idea of adding calibrated noise to the results of sensitive queries so that the presence or absence of an individual in the dataset does not significantly affect their outcome [39]. Building on earlier work, differential privacy was formally defined by introducing a rigorous framework that uses the sensitivity of a query to calibrate the amount of noise to be added. DP ensures that the output remains statistically consistent while providing strong guarantees that individual contributions cannot be distinguished, even by adversaries with additional knowledge [38].

In DP, randomized algorithms add calibrated noise to ensure that the probability of any output remains nearly unchanged, regardless of the presence or absence of any individual in the dataset, thus ensuring that individual privacy. A randomized function K provides ϵ -differential privacy [37] if for all datasets D_1 and D_2 that differ in at most one element, and for all subsets $S \subseteq \text{Range}(K)$, $\Pr[K(D_1) \in S] \leq e^\epsilon \cdot \Pr[K(D_2) \in S]$.

Extending the principles of differential privacy, techniques were developed to enable its application in deep learning, specifically through the Differentially-Private Stochastic Gradient Descent (DP-SGD) algorithm [1]. DP-SGD balances privacy guarantees with model utility by calibrating noise to the sensitivity of gradient computations. An algorithm A satisfies (ϵ, δ) -differential privacy if, for any pair of adjacent databases D and D' (where only one entry differs), and for any set of possible outcomes $S \subseteq \text{Range}(A)$, the following inequality holds: $\Pr[A(D) \in S] \leq e^\epsilon \cdot \Pr[A(D') \in S] + \delta$. Here, ϵ quantifies the gap between the probabilities of obtaining a given result when a particular data point is present or absent, whereas δ is a parameter that controls the likelihood of differential privacy being violated.

DP protects the privacy of the individuals who are part of the training dataset [73] ensuring that their presence or absence in the training set does not significantly affect the model output [66]. DP is highly effective in mitigating membership inference attacks and provides a mathematical guarantee on privacy, so that no sensitive information can be deduced for individual data instances in the training set [42].

4.1.10 Fine-tuning. Fine-tuning was one of the earliest defenses against backdoor attacks. Initially, researchers explored basic fine-tuning techniques that retrain the model with clean data to mitigate backdoor effects. This method was first introduced as a way to adjust the weights of a model that had been compromised by malicious inputs to reset the desired model behavior [50]. Fine-tuning methods evolved to account for the specific characteristics of backdoor attacks. A variation of this approach, known as fine-mixing, was proposed to combine both poisoned and clean data in a more sophisticated manner. Fine-mixing first merges the contaminated model weights with the pre-trained model weights, and then refines the combination using a limited set of clean data, allowing for better preservation of the model overall functionality while removing the backdoor effects [83]. Fine-tuning continues to be an essential white-box defense against adversarial attacks [79].

4.2 Detection-based defenses

Detection-based defenses aim to identify when the prompt has been compromised, using monitoring algorithms and constant auditing of the model responses to detect suspicious patterns or anomalous behavior [79].

4.2.1 Perplexity-based detection. Perplexity was originally proposed in the context of speech recognition [67] and is commonly used in NLP for measuring the quality of a language model. In information theory, perplexity is a measure of uncertainty in the value of a sample from a discrete probability distribution. The larger the perplexity, the less likely it is that an observer can guess the value which will be drawn from the distribution.

Perplexity per token is mathematically defined as $PPL(x) = \exp\left(-\frac{1}{t} \sum_{i=1}^t \log p(x_i | x_{<i})\right)$, where $x = (x_1, x_2, \dots, x_t)$ is a sequence of t tokens, and $p(x_i | x_{<i})$ is the conditional probability of the token x_i given all preceding tokens $x_{<i}$. A lower perplexity indicates that the sequence is more "natural" according to the model. Perplexity (per token) is an information theoretic measure that evaluates the similarity of proposed model to the original distribution and it can be interpreted as the exponentiated cross entropy.

Perplexity-based detection (PPL) [6] is based on the observation that queries with adversarial suffixes have exceedingly high perplexity values. Since injecting adversarial instructions or data into a text often increases its perplexity, a simple detector can be designed: when the perplexity of the input text exceeds a predefined threshold, it is considered to be compromised.

A classifier can also be trained to differentiate between adversarial and non-adversarial inputs by considering perplexity and token sequence length (another indicator of potentially adversarial inputs). Taking both signals into account (perplexity and length), the resulting classifier significantly outperforms simple perplexity thresholding.

Another variant, windowed perplexity detection [65], divides the input text into contiguous windows and calculates the perplexity of each window. If any window exceeds the threshold, the input is considered to be compromised.

While machine-generated adversarial attacks exhibit high perplexity and long token sequences, human-crafted adversarial attacks tend to mimic normal text with low perplexity and lengths that are similar to benign inputs, making their detection more challenging.

4.2.2 Naive LLM-based detection. The LLM itself can be used to detect compromised data. The LLM is queried with a specific instruction about the content of the input text in order to determine whether it is compromised or clean. If the LLM indicates that the provided text is compromised, it is acted upon accordingly. The defense relies on the LLM ability to identify whether a prompt includes malicious or unauthorized instructions [147].

4.2.3 Response-based detection. Response-based detection relies on the LLM prior knowledge of the expected response for a specific task. In contrast to the naive LLM-based detection above, response-based defenses first generate a response before evaluating whether the response is harmful. If the response generated by the LLM does not match valid responses expected for the target task, the input prompt is considered to be compromised. A key limitation of this defense mechanism is that it fails when both the injected task and the target task are of the same kind [165].

As in naive LLM-based detection, response-based detection can leverage the intrinsic capabilities of LLMs to evaluate the response [99] rather than the input prompt. Backtranslation [143] is another potential strategy: given the initial LLM-generated response, prompts the LLM to infer an input prompt that can lead to that response (the backtranslated prompt), which tends to reveal the actual intent of the original prompt, and run the LLM again on the backtranslated prompt (i.e. the original prompt is rejected if the model refuses the backtranslated prompt). A third alternative consists of making the LLM aware of potential harm by asking it to repeat its response [172]: when the system is unable to

repeat the LLM output generated from a malicious user input, the BLEU score between the original LLM output and repetition falls below a similarity threshold (a hyper-parameter of the detection method); then, the user input classified as malicious and the repeated output is returned to the output instead of the original output.

Another approach for response-based detection is based on the use of classifiers, such as Llama Guard [63] and Self-Guard [145], which are applied to categorize prompt-response pairs into safe or unsafe categories, enhancing robustness against adversarial prompts.

4.2.4 Perturbation. Perturbing the LLM input can also help us detect the presence of malicious inputs. Inserting a perturbation including rare words into the input text provokes different effects depending on the particular situation. When adding the perturbation to a clean input, the model output probability of the target class drops. However, when adding this rare word perturbation to a poisoned sample, the confidence of the target class does not change too much, since the attacker’s goal is to make the trigger work in a wide range of situations.

That difference between clean and malicious inputs can be used to create an efficient online defense mechanism based on robustness-aware perturbations (RAP) [161]. RAP exploits the gap of robustness between clean and contaminated samples, so they can be distinguished. RAP provides an effective defense mechanism against backdoor attacks on LLMs at inference time.

4.2.5 Masking-differential prompting. Masking-differential prompting (MDP) [151] is a defense mechanism designed to mitigate backdoor attacks in pre-trained language models (PLMs) operating as few-shot learners. MDP leverages the gap between the masking-sensitivity of poisoned and clean samples: with reference to the limited few-shot data as distributional anchors, MDP compares the representations of given samples under varying masking and identifies poisoned samples as the ones with significant variations.

MDP capitalizes on the higher sensitivity of poisoned samples to random masking when compared to clean samples. This method measures the representational change of a sample before and after random masking, allowing the identification of poisoned data. Specifically, the model sensitivity to masking is quantified by $\tau(X_{\text{test}}) = \Delta(d(X_{\text{test}}), d(\hat{X}_{\text{test}}))$, where $d(X_{\text{test}})$ denotes the feature vector of a test sample X_{test} , and \hat{X}_{test} is the same sample after random masking. The function Δ measures the distance between the original and masked representations, and the variation $\tau(X_{\text{test}})$ is used to classify a sample as poisoned when the variation $\tau(X_{\text{test}})$ exceeds a defined threshold.

4.2.6 Anomaly detection. Anomalies, also called outliers in Statistics, stand out from the rest of data. They deviate so much from other observations that they lead to suspicion that they were generated by a different mechanism [54].

Defense mechanisms based on anomaly detection rely on identifying poisoned examples in the training dataset by detecting outliers or anomalies. Anomaly detection can be performed using features such as text embeddings or the perplexity of texts.

A well-known regularization technique such as early stopping can be used to reduce the number of training epochs as a simple mechanism that limits the impact of poisoning (apart from overfitting), achieving a moderate defense against poisoning at the cost of some prediction accuracy. Methods designed for the identification and removal of poisonous examples from data are much more effective. Multiple strategies based on anomaly detection are possible [138]. For instance, poisoned examples often contain phrases that are not fluent English, so perplexity can be used to identify non-fluent sentences. Text embeddings can also be useful for locating poisoned examples: even when poisoned examples have no lexical overlap with trigger phrases, their embeddings might appear suspiciously similar. Extending this

Table 1. Attacks on LLMs, including the acronyms used in Table 2 and whether they target model integrity and/or data privacy.

Phase	Acronym	Attack	Model integrity	Data privacy
During training...	BA	Backdoor attack	X	
	DP	Data poisoning	X	
	GL	Gradient leakage		X
On trained models...	AI	Adversarial input	X	
	JB	Jailbreaking	X	X
	PI	Prompt injection	X	X
	MI	Model inversion		X
	DE	Data extraction		X
	MLA	Membership inference attack		X
	EI	Embedding inversion		X

approach, recent advancements leverage semantic anomaly detection techniques [40]: LLMs themselves can identify inconsistencies in sentence-level semantics and detect inputs that deviate significantly from expected patterns.

5 Coverage of defenses against attacks on LLMs

Large language models, such as GPT, Gemini, Llama, Grok, or Claude, face a variety of security risks arising from their ability to process and generate text based on large and diverse training databases [31]. Malicious actors can exploit LLM vulnerabilities by performing a wide range of attacks on LLM-based systems. Given the increased use of these models in sensitive applications, it is crucial to understand how different kinds of attacks can be prevented and/or mitigated with the help of existing defense mechanisms.

Table 1 lists the kinds of attacks that exploit LLM vulnerabilities in different stages of their life cycle, as well as their ultimate target, which might be compromising model integrity or getting access to private data.

Table 2 shows how existing defense mechanisms address the vulnerabilities exploited by different kinds of attacks on LLMs, providing an eagle-eyed view of the LLM security landscape. To the best of our knowledge, it summarizes the current state of the art. Apart from indicating which defense mechanisms and techniques are effective against different kinds of attacks, it also estimates the protection effectiveness they provide. A quick recap is now in order:

- For backdoor attacks, embedding purification is moderately effective because, although cleaning the embedded representations can decrease the potential attack surface area, it can never guarantee the prevention of all possible backdoors, especially the more sophisticated ones. Perturbation is also moderately effective, as masking-differential prompting, because backdoor attacks can be designed to be robust against perturbations and masking. Fine-tuning and fine-mixing could, in principle, be highly effective to identify and remove backdoors.
- Against data poisoning attacks, anomaly detection can be highly effective, in the same sense that fine-tuning was effective for backdoor attacks. In principle, given sufficiently advanced anomaly detection techniques, the attacker might find it difficult to evade detection.
- The effectiveness of differential privacy against gradient leakage attacks has proven in practice to be low for several reasons. First, differential privacy relies on adding noise to gradients to hide the information they reveal about training data. While this technique can be effective in protecting individual data, in the context of LLMs,

Category	Defense	Attacks during training			Attacks on trained models							Effectiveness
		BA	DP	GL	AI	JB	PI	MI	DE	MIA	EI	
Prevention	Paraphrasing				✓	✓	✓					Moderate
	Retokenization				✓	✓	✓					High
	Delimiters				✓	✓	✓					Moderate
	Sandwich prevention						✓					Moderate
	Instructional prevention						✓					Moderate
	Embedding purification	✓										Moderate
	SmoothLLM					✓						High
	Dimensional masking										✓	High
	Differential privacy			~						✓		Low (GL) / High (MIA)
	Fine-tuning	✓										High
Detection	Perplexity-based detection					✓	✓		✓			Moderate
	Naive LLM-based detection						✓					Moderate
	Response-based detection						✓					Moderate
	Perturbation	✓										Moderate
	Masking-differential prompting	✓										Moderate
	Anomaly detection		✓									High
Without known defenses								✗				N/A

Table 2. Available defenses for different kinds of attacks on LLMs and their effectiveness.

the additional noise may not be sufficient to mitigate more sophisticated attacks that exploit patterns and correlations underlying gradients on a large scale. Furthermore, implementing differential privacy in LLMs requires tuning parameters such as the noise level, which often represents a critical trade-off between privacy and model utility. A high noise level to ensure privacy can significantly degrade the model performance.

- In adversarial input attacks, paraphrasing and retokenization techniques demonstrate moderate and high levels of effectiveness, respectively. The effectiveness of the use of delimiters is also moderate.
 - Paraphrasing is moderately effective because it can mislead or disrupt adversarial inputs by changing superficial input text features (i.e. its wording). However, its effectiveness may be limited, as those transformations alone might not be sufficient to prevent a well-crafted adversarial input from succeeding.
 - Conversely, retokenization is highly effective because changing token segmentation alters how the model interprets the input at a more fundamental level, making it more difficult for attackers to predict how their malicious inputs will be processed by the model and significantly reducing the efficacy of such attacks.
 - Delimiters are also moderately effective in defending against adversarial inputs. Spotlighting and prompt data isolation provide some protection by separating different parts of the input, yet a clever attacker might circumvent such defenses the same way SQL-injection attacks can be performed on SQL databases.
- Against jailbreaking attacks, paraphrasing, retokenization, and the use of delimiters demonstrate varying levels of effectiveness, as each approach prepares the input differently. Detection-based techniques also exhibit different degrees of success against jailbreaking.
 - Paraphrasing is moderately effective, as it reformulates inputs, potentially disrupting some jailbreaking attempts by altering the structure and content of the input.
 - Retokenization can be highly effective against jailbreaking attacks, fundamentally changing how the model interprets the input and making jailbreak attacks considerably harder to execute successfully.
 - Delimiters are only moderately effective because, even when they try to enforce a clear, rigid structure within the input, they cannot guarantee it in the same way a parameterized SQL query is interpreted by a RDBMS to prevent injection attacks (the LLM is still there as the final interpreter of the input).
 - Perplexity-based detection (PPL) shows moderate effectiveness against jailbreaking attacks, since it can spot some jailbreaking attempts but is far from infallible.
 - SmoothLLM, on the other hand, can be highly effective, enhancing the model resilience to malicious inputs and limiting the attackers' ability to jailbreak the model (as anomaly detection against data poisoning).
- For prompt injection attacks, multiple defenses have been devised due to the high prevalence of these attacks that compromise the integrity of LLMs.
 - Retokenization, spotlighting, and prompt data isolation address input manipulation from different angles. Retokenization makes it difficult for attackers to manipulate the input, hence its effectiveness can be high. Delimiter-based techniques, such as spotlighting and prompt data isolation, have inherent limitations that make them vulnerable to sophisticated attacks.
 - Paraphrasing, sandwich prevention, and instructional prevention try to frame user-provided inputs in different ways. They are moderately effective against prompt injection attacks, as attackers can find sophisticated ways to evade these defense mechanisms.
 - Perplexity-based (PPL), naive LLM-based, and response-based detection monitor the behavior of the model when faced with suspicious inputs. They are also moderately effective. A canny attacker might find creative ways to surpass those defenses.

- At the time of this writing, no specific defense mechanisms are known against model inversion attacks.
- Data extraction attacks can be mitigated by perplexity-based detection (PPL). Its effectiveness is limited (i.e. moderate) because attackers can craft their input in a way that keeps perplexity within the range expected by the model, thus avoiding detection.
- Against membership inference attacks (MIA), differential privacy (DP) is the way to go. DP is highly effective defense due to its fundamental ability to protect individual privacy in the context of machine learning models.
- Finally, dimensional masking defense is highly effective against embedding inversion attacks. First, this technique causes a misalignment in the interpretation of the embedding, drastically reducing the attackers' ability to recover the original text. Second, it does so without compromising the LLM performance.

6 Conclusion

In this paper, we have explored the risks associated to the use of large language models (LLMs), which affect both the integrity of LLM-based systems and the privacy of the data LLMs are trained with.

Known LLM vulnerabilities have been exploited by different kinds of attacks, which can act both when the LLM is being trained (i.e. during training) and also when it is deployed in practice (i.e. during inference). Our survey has analyzed LLM attack vectors and the defensive mechanisms that have been devised to prevent, mitigate, or detect those attacks.

Even though it should be noted that security is always relative, our study of the coverage of defense mechanisms against attacks on LLM-based systems has highlighted areas that require additional attention to ensure the reliable use of LLMs in sensitive applications. Only a handful of defenses are highly effective, in the sense that they can be made as sophisticated as the sophistication of the attacker might require them to be. Most existing defense mechanisms and countermeasures, unfortunately, present inherent limitations that can be exploited by informed and sufficiently-skilled malicious actors.

References

- [1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (Vienna, Austria) (CCS '16)*. Association for Computing Machinery, New York, NY, USA, 308–318. <https://doi.org/10.1145/2976749.2978318>
- [2] Sara Abdali, Richard Anarfi, CJ Barberan, and Jia He. 2024. Securing Large Language Models: Threats, Vulnerabilities and Responsible Practices. arXiv:2403.12503 [cs.CR] <https://arxiv.org/abs/2403.12503>
- [3] Sara Abdali, Jia He, CJ Barberan, and Richard Anarfi. 2024. Can LLMs be Fooled? Investigating Vulnerabilities in LLMs. arXiv:2407.20529 [cs.LG] <https://arxiv.org/abs/2407.20529>
- [4] Microsoft Research AI4Science and Microsoft Azure Quantum. 2023. The Impact of Large Language Models on Scientific Discovery: a Preliminary Study using GPT-4. arXiv:2311.07361 [cs.CL] <https://arxiv.org/abs/2311.07361>
- [5] Ali Al-Kaswan, Maliheh Izadi, and Arie van Deursen. 2023. Targeted Attack on GPT-Neo for the SATML Language Model Data Extraction Challenge. arXiv:2302.07735 [cs.CL] <https://arxiv.org/abs/2302.07735>
- [6] Gabriel Alon and Michael Kamfonas. 2023. Detecting Language Model Attacks with Perplexity. arXiv:2308.14132 [cs.CL] <https://arxiv.org/abs/2308.14132>
- [7] Anthropic. 2024. Introducing Claude 3.5 Sonnet. <https://www.anthropic.com/news/claude-3-5-sonnet>. Accessed: 2024-11-10.
- [8] Li Bai, Haibo Hu, Qingqing Ye, Haoyang Li, Leixia Wang, and Jianliang Xu. 2024. Membership Inference Attacks and Defenses in Federated Learning: A Survey. <https://doi.org/10.1145/3704633> Just Accepted.
- [9] Mislav Balunovic, Dimitar Dimitrov, Nikola Jovanović, and Martin Vechev. 2022. LAMP: Extracting Text from Gradients with Language Model Priors. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., ETH Zurich, 7641–7654. https://proceedings.neurips.cc/paper_files/paper/2022/file/32375260090404f907ceae19f3564a7e-Paper-Conference.pdf
- [10] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D. Joseph, and J. D. Tygar. 2006. Can machine learning be secure?. In *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security (Taipei, Taiwan) (ASIACCS '06)*. Association for Computing Machinery,

- New York, NY, USA, 16–25. <https://doi.org/10.1145/1128817.1128824>
- [11] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research* 3 (2003), 1137–1155.
- [12] Fernando Berzal. 2019. *Redes Neuronales & Deep Learning - Volumen I: Entrenamiento de redes neuronales artificiales [Neural Networks and Deep Learning - Volume I: Training Artificial Neural Networks, in Spanish]*. Amazon KDP, Granada, Spain. <https://deep-learning.ikor.org/>
- [13] Fernando Berzal. 2019. *Redes Neuronales & Deep Learning - Volumen II: Regularización, optimización & arquitecturas especializadas [Neural Networks and Deep Learning - Volume II: Regularization, Optimization, and Specialized Architectures, in Spanish]*. Amazon KDP, Granada, Spain. <https://deep-learning.ikor.org/>
- [14] Satwik Bhattamishra, Arkil Patel, Phil Blunsom, and Varun Kanade. 2023. Understanding In-Context Learning in Transformers and LLMs by Learning to Learn Discrete Functions. arXiv:2310.03016 [cs.LG] <https://arxiv.org/abs/2310.03016>
- [15] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrđić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. 2013. Evasion Attacks against Machine Learning at Test Time. In *Machine Learning and Knowledge Discovery in Databases*, Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 387–402.
- [16] Lewis Birch, William Hackett, Stefan Trawicki, Neeraj Suri, and Peter Garraghan. 2023. Model Leeching: An Extraction Attack Targeting LLMs. arXiv:2309.10544 [cs.LG] <https://arxiv.org/abs/2309.10544>
- [17] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL] <https://arxiv.org/abs/2005.14165>
- [18] Xiangrui Cai, Haidong Xu, Sihang Xu, Ying Zhang, and Xiaojie Yuan. 2022. BadPrompt: Backdoor Attacks on Continuous Prompts. arXiv:2211.14719 [cs.CL] <https://arxiv.org/abs/2211.14719>
- [19] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. 2023. Quantifying Memorization Across Neural Language Models. arXiv:2202.07646 [cs.LG] <https://arxiv.org/abs/2202.07646>
- [20] Nicholas Carlini, Matthew Jagielski, Christopher A. Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramèr. 2024. Poisoning Web-Scale Training Datasets is Practical. arXiv:2302.10149 [cs.CR] <https://arxiv.org/abs/2302.10149>
- [21] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting Training Data from Large Language Models. In *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Mountain View, CA, 2633–2650. <https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting>
- [22] Tarek Chaalan, Shaoning Pang, Joarder Kamruzzaman, Iqbal Gondal, and Xuyun Zhang. 2024. The Path to Defence: A Roadmap to Characterising Data Poisoning Attacks on Victim Models. *ACM Comput. Surv.* 56, 7, Article 175 (April 2024), 39 pages. <https://doi.org/10.1145/3627536>
- [23] Wenhan Chang and Tianqing Zhu. 2024. Gradient-based defense methods for data leakage in vertical federated learning. *Computers & Security* 139 (2024), 103744. <https://doi.org/10.1016/j.cose.2024.103744>
- [24] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2024. A Survey on Evaluation of Large Language Models. *ACM Trans. Intell. Syst. Technol.* 15, 3, Article 39 (mar 2024), 45 pages. <https://doi.org/10.1145/3641289>
- [25] Shreyas Chaudhari, Pranjal Aggarwal, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, Karthik Narasimhan, Ameet Deshpande, and Bruno Castro da Silva. 2024. RLHF Deciphered: A Critical Analysis of Reinforcement Learning from Human Feedback for LLMs. arXiv:2404.08555 [cs.LG] <https://arxiv.org/abs/2404.08555>
- [26] Yiyi Chen, Heather Lent, and Johannes Bjerva. 2024. Text Embedding Inversion Security for Multilingual Language Models. arXiv:2401.12192 [cs.CL] <https://arxiv.org/abs/2401.12192>
- [27] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference. arXiv:2403.04132 [cs.AI]
- [28] Junjie Chu, Yugeng Liu, Ziqing Yang, Xinyue Shen, Michael Backes, and Yang Zhang. 2024. Comprehensive Assessment of Jailbreak Attacks Against LLMs. arXiv:2402.05668 [cs.CR] <https://arxiv.org/abs/2402.05668>
- [29] Micaela E. Consens, Cameron Dufault, Michael Wainberg, Duncan Forster, Mehran Karimzadeh, Hani Goodarzi, Fabian J. Theis, Alan Moses, and Bo Wang. 2023. To Transformers and Beyond: Large Language Models for the Genome. arXiv:2311.07621 [q-bio.GN] <https://arxiv.org/abs/2311.07621>
- [30] Avisha Das, Amara Tariq, Felipe Batalini, Boddhisattwa Dhara, and Imon Banerjee. 2024. Exposing Vulnerabilities in Clinical LLMs Through Data Poisoning Attacks: Case Study in Breast Cancer. *medRxiv* 1, 1 (2024), 1–1. <https://doi.org/10.1101/2024.03.20.24304627> arXiv:<https://www.medrxiv.org/content/early/2024/03/21/2024.03.20.24304627.full.pdf>
- [31] Badhan Chandra Das, M. Hadi Amini, and Yanzhao Wu. 2024. Security and Privacy Challenges of Large Language Models: A Survey. arXiv:2402.00888 [cs.CL] <https://arxiv.org/abs/2402.00888>
- [32] Ms D Deepa et al. 2021. Bidirectional encoder representations from transformers (BERT) language model for sentiment analysis task. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12, 7 (2021), 1708–1721. URL:

- <https://turcomat.org/index.php/turkbilmal/article/view/3055>.
- [33] Jieren Deng, Yijue Wang, Ji Li, Chao Shang, Hang Liu, Sanguthevar Rajasekaran, and Caiwen Ding. 2021. TAG: Gradient Attack on Transformer-based Language Models. arXiv:2103.06819 [cs.CR] <https://arxiv.org/abs/2103.06819>
- [34] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL] <https://arxiv.org/abs/1810.04805>
- [35] Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. RAFT: Reward rAnked FineTuning for Generative Foundation Model Alignment. arXiv:2304.06767 [cs.LG] <https://arxiv.org/abs/2304.06767>
- [36] Michael Duan, Anshuman Suri, Niloofar Miresghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024. Do Membership Inference Attacks Work on Large Language Models? arXiv:2402.07841 [cs.CL] <https://arxiv.org/abs/2402.07841>
- [37] Cynthia Dwork. 2006. Differential Privacy. In *Automata, Languages and Programming*, Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1–12.
- [38] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography*, Shai Halevi and Tal Rabin (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 265–284.
- [39] Cynthia Dwork and Kobbi Nissim. 2004. Privacy-Preserving Datamining on Vertically Partitioned Databases. In *Advances in Cryptology – CRYPTO 2004*, Matt Franklin (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 528–544.
- [40] Amine Elhafi, Rohan Sinha, Christopher Agia, Edward Schmerling, Issa A. D. Nesnas, and Marco Pavone. 2023. Semantic anomaly detection with large language models. *Autonomous Robots* 47, 8 (2023), 1035–1055. <https://doi.org/10.1007/s10514-023-10132-6>
- [41] Aysan Esmradi, Daniel Wankit Yip, and Chun Fai Chan. 2024. A Comprehensive Survey of Attack Techniques, Implementation, and Mitigation Strategies in Large Language Models. In *Ubiquitous Security*, Guojun Wang, Haozhe Wang, Geyong Min, Nektarios Georgalas, and Weizhi Meng (Eds.). Springer Nature Singapore, Singapore, 76–95.
- [42] Georgios Feretakis, Konstantinos Papatyridis, Aris Gkoulalas-Divanis, and Vassilios S. Verykios. 2024. Privacy-Preserving Techniques in Generative AI and Large Language Models: A Narrative Review. *Information* 15, 11 (2024). <https://doi.org/10.3390/info15110697>
- [43] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (Denver, Colorado, USA) (CCS '15). Association for Computing Machinery, New York, NY, USA, 1322–1333. <https://doi.org/10.1145/2810103.2813677>
- [44] Xiaopeng Fu, Zhaoquan Gu, Weihong Han, Yaguan Qian, Bin Wang, and Federico Tramari. 2021. Exploring Security Vulnerabilities of Deep Learning Models by Adversarial Attacks. *Wirel. Commun. Mob. Comput.* 2021 (Jan. 2021), 9 pages. <https://doi.org/10.1155/2021/9969867>
- [45] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. 2020. Inverting Gradients – How easy is it to break privacy in federated learning? arXiv:2003.14053 [cs.CV] <https://arxiv.org/abs/2003.14053>
- [46] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press, USA. <http://www.deeplearningbook.org>
- [47] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. arXiv:1412.6572 [stat.ML] <https://arxiv.org/abs/1412.6572>
- [48] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archie Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Young, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shao-liang Nie, Sharan Narang, Sharath Papparth, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent

- Gonguet, Virginie Do, Vish Vogeti, Vitor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuwei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Barambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guanyu, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Harour Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelen, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI] <https://arxiv.org/abs/2407.21783>
- [49] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not what you’ve signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. arXiv:2302.12173 [cs.CR] <https://arxiv.org/abs/2302.12173>
- [50] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2019. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. arXiv:1708.06733 [cs.CR] <https://arxiv.org/abs/1708.06733>
- [51] Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. 2021. Gradient-based Adversarial Attacks against Text Transformers. arXiv:2104.13733 [cs.CL] <https://arxiv.org/abs/2104.13733>
- [52] Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seydedali Mirjalili, et al. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage.
- [53] J.B. Hampshire and A. Waibel. 1992. The Meta-Pi network: building distributed knowledge representations for robust multisource pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 7 (1992), 751–769. <https://doi.org/10.1109/34.142911>
- [54] Douglas M. Hawkins. 1980. *Identification of Outliers*. Chapman and Hall, Kluwer Academic Publishers, Boston/Dordrecht/London.
- [55] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. 2018. LOGAN: Membership Inference Attacks Against Generative Models. arXiv:1705.07663 [cs.CR] <https://arxiv.org/abs/1705.07663>
- [56] Kai He, Rui Mao, Qika Lin, Yucheng Ruan, Xiang Lan, Mengling Feng, and Erik Cambria. 2024. A Survey of Large Language Models for Healthcare: from Data, Technology, and Applications to Accountability and Ethics. arXiv:2310.05694 [cs.CL] <https://arxiv.org/abs/2310.05694>
- [57] Keegan Hines, Gary Lopez, Matthew Hall, Federico Zarfaty, Yonatan Zunger, and Emre Kiciman. 2024. Defending Against Indirect Prompt Injection Attacks With Spotlighting. arXiv:2403.14720 [cs.CR] <https://arxiv.org/abs/2403.14720>

- [58] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S. Yu, and Xuyun Zhang. 2022. Membership Inference Attacks on Machine Learning: A Survey. *ACM Comput. Surv.* 54, 11s, Article 235 (Sept. 2022), 37 pages. <https://doi.org/10.1145/3523273>
- [59] Li Hu, Anli Yan, Hongyang Yan, Jin Li, Teng Huang, Yingying Zhang, Changyu Dong, and Chunsheng Yang. 2023. Defenses to Membership Inference Attacks: A Survey. *ACM Comput. Surv.* 56, 4, Article 92 (Nov. 2023), 34 pages. <https://doi.org/10.1145/3620667>
- [60] Hai Huang, Zhengyu Zhao, Michael Backes, Yun Shen, and Yang Zhang. 2024. Composite Backdoor Attacks Against Large Language Models. arXiv:2310.07676 [cs.CR] <https://arxiv.org/abs/2310.07676>
- [61] Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. 2022. Are Large Pre-Trained Language Models Leaking Your Personal Information?. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2038–2047. <https://doi.org/10.18653/v1/2022.findings-emnlp.148>
- [62] Yunpeng Huang, Jingwei Xu, Junyu Lai, Zixu Jiang, Taolue Chen, Zenan Li, Yuan Yao, Xiaoxing Ma, Lijuan Yang, Hao Chen, Shupeng Li, and Penghao Zhao. 2024. Advancing Transformer Architecture in Long-Context Large Language Models: A Comprehensive Survey. arXiv:2311.12351 [cs.CL] <https://arxiv.org/abs/2311.12351>
- [63] Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. 2023. Llama Guard: LLM-based Input-Output Safeguard for Human-AI Conversations. arXiv:2312.06674 [cs.CL] <https://arxiv.org/abs/2312.06674>
- [64] Raisa Islam and Intiaz Ahmed. 2024. Gemini-the most powerful LLM: Myth or Truth. In *2024 5th Information Communication Technologies Conference (ICTC)*. IEEE, Socorro, New Mexico, USA, 303–308. <https://doi.org/10.1109/ICTC61510.2024.10602253>
- [65] Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. 2023. Baseline Defenses for Adversarial Attacks Against Aligned Language Models. arXiv:2309.00614 [cs.LG] <https://arxiv.org/abs/2309.00614>
- [66] Balder Janryd and Tim Johansson. 2024. Preventing Health Data from Leaking in a Machine Learning System : Implementing code analysis with LLM and model privacy evaluation testing. , 84 pages.
- [67] F. Jelinek, R. L. Mercer, L. R. Bahl, and J. K. Baker. 1977. Perplexity—A measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America* 62, S1 (1977), S63–S63. <https://doi.org/10.1121/1.2016299> arXiv:https://pubs.aip.org/asa/jasa/article-pdf/62/S1/S63/11558910/s63_5_online.pdf
- [68] Anthony D. Joseph, Blaine Nelson, Benjamin I. P. Rubinstein, and J. D. Tygar. 2019. *Adversarial Machine Learning* (1st ed.). Cambridge University Press, USA.
- [69] Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating Training Data Mitigates Privacy Risks in Language Models. arXiv:2202.06539 [cs.CR] <https://arxiv.org/abs/2202.06539>
- [70] Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and Gjergji Kasneci. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences* 103 (2023), 102274. <https://doi.org/10.1016/j.lindif.2023.102274>
- [71] Aly M. Kassem, Omar Mahmoud, Niloofar Miresghallah, Hyunwoo Kim, Yulia Tsvetkov, Yejin Choi, Sherif Saad, and Santu Rana. 2024. Alpaca against Vicuna: Using LLMs to Uncover Memorization of LLMs. arXiv:2403.04801 [cs.CL] <https://arxiv.org/abs/2403.04801>
- [72] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. 2024. On the Reliability of Watermarks for Large Language Models. arXiv:2306.04634 [cs.LG] <https://arxiv.org/abs/2306.04634>
- [73] Alexey Kurakin, Natalia Ponomareva, Umar Syed, Liam MacDermed, and Andreas Terzis. 2024. Harnessing large-language models to generate private synthetic text. arXiv:2306.01684 [cs.LG] <https://arxiv.org/abs/2306.01684>
- [74] Minhyeok Lee. 2023. A mathematical investigation of hallucination and creativity in GPT models. *Mathematics* 11, 10 (2023), 2320.
- [75] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., Facebook AI Research, 9459–9474. https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf
- [76] Haoran Li, Yulin Chen, Jinglong Luo, Yan Kang, Xiaojin Zhang, Qi Hu, Chunkit Chan, and Yangqiu Song. 2023. Privacy in Large Language Models: Attacks, Defenses and Future Directions. arXiv:2310.10383 [cs.CL] <https://arxiv.org/abs/2310.10383>
- [77] Linyang Li, Demin Song, Xiaonan Li, Jiehang Zeng, Ruotian Ma, and Xipeng Qiu. 2021. Backdoor Attacks on Pre-trained Models by Layerwise Weight Poisoning. arXiv:2108.13888 [cs.CR] <https://arxiv.org/abs/2108.13888>
- [78] Shaofeng Li, Hui Liu, Tian Dong, Benjamin Zi Hao Zhao, Minhui Xue, Haojin Zhu, and Jialiang Lu. 2021. Hidden Backdoors in Human-Centric Language Models. arXiv:2105.00164 [cs.CL] <https://arxiv.org/abs/2105.00164>
- [79] Frank Weizhen Liu and Chenhui Hu. 2024. Exploring Vulnerabilities and Protections in Large Language Models: A Survey. arXiv:2406.00240 [cs.LG] <https://arxiv.org/abs/2406.00240>
- [80] Xiaogeng Liu, Zhiyuan Yu, Yizhe Zhang, Ning Zhang, and Chaowei Xiao. 2024. Automatic and Universal Prompt Injection Attacks against Large Language Models. arXiv:2403.04957 [cs.AI] <https://arxiv.org/abs/2403.04957>

- [81] Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Zihao Wang, Xiaofeng Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. 2024. Prompt Injection attack against LLM-integrated Applications. arXiv:2306.05499 [cs.CR] <https://arxiv.org/abs/2306.05499>
- [82] Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. 2024. Formalizing and Benchmarking Prompt Injection Attacks and Defenses. arXiv:2310.12815 [cs.CR] <https://arxiv.org/abs/2310.12815>
- [83] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. 2018. Trojanning Attack on Neural Networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-22, 2018*. The Internet Society.
- [84] Ziyao Liu, Huanyi Ye, Chen Chen, Yongsun Zheng, and Kwok-Yan Lam. 2024. Threats, Attacks, and Defenses in Machine Unlearning: A Survey. arXiv:2403.13682 [cs.CR] <https://arxiv.org/abs/2403.13682>
- [85] Udara Piyasena Liyanage and Nimnaka Dilshan Ranaweera. 2023. Ethical Considerations and Potential Risks in the Deployment of Large Language Models in Diverse Societal Contexts. *Journal of Computational Social Dynamics* 8, 11 (Nov 2023), 15–25. <https://vectoral.org/index.php/JCSD/article/view/49>
- [86] Mayra Macas, Chunming Wu, and Walter Fuertes. 2024. Adversarial examples: A survey of attacks and defenses in deep learning-enabled cybersecurity systems. *Expert Systems with Applications* 238 (2024), 122223. <https://doi.org/10.1016/j.eswa.2023.122223>
- [87] Nitin Madnani and Bonnie J. Dorr. 2010. Generating Phrasal and Sentential Paraphrases: A Survey of Data-Driven Methods. *Computational Linguistics* 36, 3 (09 2010), 341–387. https://doi.org/10.1162/coli_a_00002 arXiv:https://direct.mit.edu/coli/article-pdf/36/3/341/1812631/coli_a_00002.pdf
- [88] John McCarthy. 2004. What is Artificial Intelligence? Computer Science Department, Stanford University, Stanford, CA 94305. Revised November 12, 2007. URL: <http://www-formal.stanford.edu/jmc/>.
- [89] Dongyu Meng and Hao Chen. 2017. MagNet: a Two-Pronged Defense against Adversarial Examples. arXiv:1705.09064 [cs.CR] <https://arxiv.org/abs/1705.09064>
- [90] Yash More, Prakhar Ganesh, and Golnoosh Farnadi. 2024. Towards More Realistic Extraction Attacks: An Adversarial Perspective. arXiv:2407.02596 [cs.CR] <https://arxiv.org/abs/2407.02596>
- [91] John X. Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander M. Rush. 2023. Text Embeddings Reveal (Almost) As Much As Text. arXiv:2310.06816 [cs.CL] <https://arxiv.org/abs/2310.06816>
- [92] John X. Morris, Wenting Zhao, Justin T. Chiu, Vitaly Shmatikov, and Alexander M. Rush. 2023. Language Model Inversion. arXiv:2311.13647 [cs.CL] <https://arxiv.org/abs/2311.13647>
- [93] Maximilian Mozes, Xuanli He, Bennett Kleinberg, and Lewis D. Griffin. 2023. Use of LLMs for Illicit Purposes: Threats, Prevention Measures, and Vulnerabilities. arXiv:2308.12833 [cs.CL] <https://arxiv.org/abs/2308.12833>
- [94] Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. 2023. Scalable Extraction of Training Data from (Production) Language Models. arXiv:2311.17035 [cs.LG] <https://arxiv.org/abs/2311.17035>
- [95] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2024. A Comprehensive Overview of Large Language Models. arXiv:2307.06435 [cs.CL] <https://arxiv.org/abs/2307.06435>
- [96] Daryna Oliyynyk, Rudolf Mayer, and Andreas Rauber. 2023. I Know What You Trained Last Summer: A Survey on Stealing Machine Learning Models and Defences. *ACM Comput. Surv.* 55, 14s, Article 324 (July 2023), 41 pages. <https://doi.org/10.1145/3595292>
- [97] OpenAI, ;, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex TachardPassos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoochian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispati, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Gertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O’Connell, Ian O’Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela,

- Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feувrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madeline Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljube, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunningham, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyei Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. 2024. GPT-4o System Card. arXiv:2410.21276 [cs.CL] <https://arxiv.org/abs/2410.21276>
- [98] Fábio Perez and Ian Ribeiro. 2022. Ignore Previous Prompt: Attack Techniques For Language Models. arXiv:2211.09527 [cs.CL] <https://arxiv.org/abs/2211.09527>
- [99] Mansi Phute, Alec Helbling, Matthew Hull, ShengYun Peng, Sebastian Szyller, Cory Cornelius, and Duen Horng Chau. 2024. LLM Self Defense: By Self Examination, LLMs Know They Are Being Tricked. arXiv:2308.07308 [cs.CL] <https://arxiv.org/abs/2308.07308>
- [100] Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-Dropout: Simple and Effective Subword Regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 1882–1892. <https://doi.org/10.18653/v1/2020.acl-main.170>
- [101] Xiao Pu, Mingqi Gao, and Xiaojun Wan. 2023. Summarization is (Almost) Dead. arXiv:2309.09558 [cs.CL] <https://arxiv.org/abs/2309.09558>
- [102] Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2021. Mind the Style of Text! Adversarial and Backdoor Attacks Based on Text Style Transfer. arXiv:2110.07139 [cs.CL] <https://arxiv.org/abs/2110.07139>
- [103] Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. 2023. Visual Adversarial Examples Jailbreak Aligned Large Language Models. arXiv:2306.13213 [cs.CR] <https://arxiv.org/abs/2306.13213>
- [104] Zhenting Qi, Hanlin Zhang, Eric Xing, Sham Kakade, and Himabindu Lakkaraju. 2024. Follow My Instruction and Spill the Beans: Scalable Data Extraction from Retrieval-Augmented Generation Systems. arXiv:2402.17840 [cs.CL] <https://arxiv.org/abs/2402.17840>
- [105] Baha Rababah, Shang, Wu, Matthew Kwiatkowski, Carson Leung, and Cuneyt Gurcan Akcora. 2024. SoK: Prompt Hacking of Large Language Models. arXiv:2410.13901 [cs.CR] <https://arxiv.org/abs/2410.13901>
- [106] Alec Radford, Aditya Kwon, Marloes Koot, et al. 2018. *Improving Language Understanding by Generative Pre-Training*. Technical Report. OpenAI. <https://openai.com/research/language-unsupervised>
- [107] Mohaimenul Azam Khan Raiaan, Md. Saddam Hossain Mukta, Kaniz Fatema, Nur Mohammad Fahad, Sadman Sakib, Most Marufatul Jannat Mim, Jubaer Ahmad, Mohammed Eunus Ali, and Sami Azam. 2024. A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges. *IEEE Access* 12 (2024), 26839–26874. <https://doi.org/10.1109/ACCESS.2024.3365742>
- [108] Alexander Robey, Eric Wong, Hamed Hassani, and George J. Pappas. 2024. SmoothLLM: Defending Large Language Models Against Jailbreaking Attacks. arXiv:2310.03684 [cs.LG] <https://arxiv.org/abs/2310.03684>
- [109] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. 1992. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena* 60, 1 (1992), 259–268. [https://doi.org/10.1016/0167-2789\(92\)90242-F](https://doi.org/10.1016/0167-2789(92)90242-F)
- [110] Stuart J Russell and Peter Norvig. 2016. *Artificial Intelligence: a Modern Approach*. Pearson, USA.
- [111] Vinu Sankar Sadasivan, Shoumik Saha, Gaurang Sriramanan, Priyatham Kattakinda, Atoosa Chegini, and Soheil Feizi. 2024. Fast Adversarial Attacks on Language Models In One GPU Minute. arXiv:2402.15570 [cs.CR] <https://arxiv.org/abs/2402.15570>
- [112] Sander Schulhoff. 2023. Instruction Defense. https://learnprompting.org/docs/prompt_hacking/defensive_measures/instruction Accessed: 2024-12-04.
- [113] Leo Schwinn, David Dobre, Stephan Günemann, and Gauthier Gidel. 2023. Adversarial attacks and defenses in large language models: Old and new threats. *arXiv preprint arXiv:2310.19737* 239 (2023), 103–117.

- [114] Pasi Shailendra, Rudra Chandra Ghosh, Rajdeep Kumar, and Nitin Sharma. 2024. Survey of Large Language Models for Answering Questions Across Various Fields. , 520-527 pages. <https://doi.org/10.1109/ICACCS60874.2024.10717078>
- [115] Erfan Shayegani, Md Abdullah Al Mamun, Yu Fu, Pedram Zaree, Yue Dong, and Nael Abu-Ghazaleh. 2023. Survey of Vulnerabilities in Large Language Models Revealed by Adversarial Attacks. arXiv:2310.10844 [cs.CL] <https://arxiv.org/abs/2310.10844>
- [116] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2024. "Do Anything Now": Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models. arXiv:2308.03825 [cs.CR] <https://arxiv.org/abs/2308.03825>
- [117] Jingzhe Shi, Jialuo Li, Qinwei Ma, Zaiwen Yang, Huan Ma, and Lei Li. 2024. CHOPS: CHat with custOmer Profile Systems for Customer Service with LLMs. arXiv:2404.01343 [cs.CL] <https://arxiv.org/abs/2404.01343>
- [118] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks against Machine Learning Models. arXiv:1610.05820 [cs.CR] <https://arxiv.org/abs/1610.05820>
- [119] Victoria Smith, Ali Shahin Shamsabadi, Carolyn Ashurst, and Adrian Weller. 2024. Identifying and Mitigating Privacy Risks Stemming from Language Models: A Survey. arXiv:2310.01424 [cs.CL] <https://arxiv.org/abs/2310.01424>
- [120] Congzheng Song and Ananth Raghunathan. 2020. Information Leakage in Embedding Models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (Virtual Event, USA) (CCS '20)*. Association for Computing Machinery, New York, NY, USA, 377–390. <https://doi.org/10.1145/3372297.3417270>
- [121] Congzheng Song and Vitaly Shmatikov. 2019. Auditing Data Provenance in Text-Generation Models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (Anchorage, AK, USA) (KDD '19)*. Association for Computing Machinery, New York, NY, USA, 196–206. <https://doi.org/10.1145/3292500.3330885>
- [122] Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2024. Preference Ranking Optimization for Human Alignment. arXiv:2306.17492 [cs.CL] <https://arxiv.org/abs/2306.17492>
- [123] Junzhe Song and Dmitry Namiot. 2022. A Survey of Model Inversion Attacks and Countermeasures. Lomonosov Moscow State University, GSP-1, Leninskie Gory, Moscow, 119991. URL: https://damdid2022.frccsc.ru/files/article/DAMDID_2022_paper_1040.pdf.
- [124] Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. 2023. Retentive Network: A Successor to Transformer for Large Language Models. arXiv:2307.08621 [cs.CL] <https://arxiv.org/abs/2307.08621>
- [125] Xuchen Suo. 2024. Signed-Prompt: A New Approach to Prevent Prompt Injection Attacks Against LLM-Integrated Applications. arXiv:2401.07612 [cs.CR] <https://arxiv.org/abs/2401.07612>
- [126] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. arXiv:1312.6199 [cs.CV] <https://arxiv.org/abs/1312.6199>
- [127] Jorge Sánchez-González. 2020. Sentiment Analysis in Twitter, in "Deep Learning: Learning Techniques and Applications" [in Spanish]. B.Eng. capstone project, University of Granada.
- [128] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, Heng-Tze Cheng, Eric Ni, Purvi Shah, Patrick Kane, Betty Chan, Manaal Faruqi, Aliaksei Severyn, Hanzhao Lin, YaGuang Li, Yong Cheng, Abe Ittycheriah, Mahdis Mahdieh, Mia Chen, Pei Sun, Dustin Tran, Sumit Bagri, Balaji Lakshminarayanan, Jeremiah Liu, Andras Orban, Fabian Gura, Hao Zhou, Xinying Song, Aurelien Boffy, Harish Ganapathy, Steven Zheng, HyunJeong Choe, Agoston Weisz, Tao Zhu, Yifeng Lu, Siddharth Gopal, Jarrod Kahn, Maciej Kula, Jeff Pitman, Rushin Shah, Emanuel Taropa, Majd Al Meray, Martin Baeuml, Zhifeng Chen, Laurent El Shafey, Yujing Zhang, Olcan Sercinoglu, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaıs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Prolev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maroon, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Gaurav Singh Tomar, Evan Senter, Martin Chadwick, Ilya Kornakov, Nithya Attaluri, İñaki Iturrate, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Xavier Garcia, Thanumalayan Sankaranarayanan Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Ravi Addanki, Antoine Miech, Annie Louis, Denis Teplyashin, Geoff Brown, Elliot Catt, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sebastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaliy Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke,

Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Oztürel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Inuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjöstrand, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhranjit Roy, Ethan Dyer, Victor Campos Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitaogong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Çağlar Ünlü, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlias, Arpi Vezer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fanguy Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Sidharth Mudgal, Romina Stella, Kevin Brooks, Gautam Vasudevan, Chenxi Liu, Mainak Chaitin, Nivedita Melinkeri, Aaron Cohen, Venus Wang, Kristie Seymore, Sergey Zubkov, Rahul Goel, Summer Yue, Sai Krishnakumar, Brian Albert, Nate Hurley, Motoki Sano, Anhad Mohananey, Jonah Joughin, Egor Filonov, Tomasz Kępa, Yomna Eldawy, Jiawern Lim, Rahul Rishi, Shirin Badiezedegan, Taylor Bos, Jerry Chang, Sanil Jain, Sri Gayatri Sundara Padmanabhan, Subha Puttagunta, Kalpesh Krishna, Leslie Baker, Norbert Kalb, Vamsi Bedapudi, Adam Kurzrok, Shuntong Lei, Anthony Yu, Oren Litvin, Xiang Zhou, Zhichun Wu, Sam Sobell, Andrea Siciliano, Alan Papir, Robby Neale, Jonas Bragagnolo, Tej Toor, Tina Chen, Valentin Anklin, Feiran Wang, Richie Feng, Milad Gholami, Kevin Ling, Lijuan Liu, Jules Walter, Hamid Moghaddam, Arun Kishore, Jakob Adamek, Tyler Mercado, Jonathan Mallinson, Siddhanta Wandekar, Stephen Cagle, Eran Ofek, Guillermo Garrido, Clemens Lombriser, Maksim Mukha, Botu Sun, Hafeezul Rahman Mohammad, Josip Matak, Yadi Qian, Vikas Peswani, Pawel Janus, Quan Yuan, Leif Schelin, Oana David, Ankur Garg, Yifan He, Oleksii Duzhyi, Anton Ålgmyr, Timothée Lottaz, Qi Li, Vikas Yadav, Luyao Xu, Alex Chinien, Rakesh Shivanna, Aleksandr Chuklin, Josie Li, Carrie Spadine, Travis Wolfe, Kareem Mohamed, Subhabrata Das, Zihang Dai, Kyle He, Daniel von Dincklage, Shyam Upadhyay, Akanksha Maurya, Luyan Chi, Sebastian Krause, Khalid Salama, Pam G Rabinovitch, Pavan Kumar Reddy M, Aarush Selvan, Mikhail Dektiarev, Golnaz Ghiasi, Erdem Guven, Himanshu Gupta, Boyi Liu, Deepak Sharma, Idan Heimlich Shtacher, Shachi Paul, Oscar Akerlund, François-Xavier Aubet, Terry Huang, Chen Zhu, Eric Zhu, Elico Teixeira, Matthew Fritze, Francesco Bertolini, Liana-Eleonora Marinescu, Martin Bölle, Dominik Paulus, Khyatti Gupta, Tejasi Latkar, Max Chang, Jason Sanders, Roopa Wilson, Xuewei Wu, Yi-Xuan Tan, Lam Nguyen Thiet, Tulsee Doshi, Sid Lall, Swaroop Mishra, Wanming Chen, Thang Luong, Seth Benjamin, Jasmine Lee, Ewa Andrejczuk, Dominik Rabiej, Vipul Ranjan, Krzysztof Styrz, Pengcheng Yin, Jon Simon, Malcolm Rose Harriott, Mudit Bansal, Alexei Robsky, Geoff Bacon, David Greene, Daniil Mirylenka, Chen Zhou, Obaid Sarvana, Abhimanyu Goyal, Samuel Andermatt, Patrick Siegler, Ben Horn, Assaf Israel, Francesco Pongetti, Chih-Wei "Louis" Chen, Marco Selvatici, Pedro Silva, Kathie Wang, Jackson Tolins, Kelvin Guu, Roey Yogeve, Xiaochen Cai, Alessandro Agostini, Maulik Shah, Hung Nguyen, Noah Ó Donnaille, Sébastien Pereira, Linda Friso, Adam Stambler, Adam Kurzrok, Chenkai Kuang, Yan Romanikhin, Mark Geller, ZJ Yan, Kane Jang, Cheng-Chun Lee, Wojciech Fica, Eric Malmi, Qijun Tan, Dan Banica, Daniel Balle, Ryan Pham, Yanping Huang, Diana Avram, Hongzhi Shi, Jasjot Singh, Chris Hidey, Niharika Ahuja, Pranab Saxena, Dan Dooley, Srividya Pranavi Potharaju, Eileen O'Neill, Anand Gokulchandran, Ryan Foley, Kai Zhao, Mike Dusenberry, Yuan Liu, Pulkit Mehta, Ragha Kotikalapudi, Chalence Safranek-Shrader, Andrew Goodman, Joshua Kessinger, Eran Globen, Prateek Kolhar, Chris Gorgolewski, Ali Ibrahim, Yang Song, Ali Eichenbaum, Thomas Brovelli, Sahitya Potluri, Preethi Lahoti, Cip Baetu, Ali Ghorbani, Charles Chen, Andy Crawford, Shalini Pal, Mukund Sridhar, Petru Gurita, Asier Mujika, Igor Petrovski, Pierre-Louis Cedoz, Chenmei Li, Shiyuan Chen, Niccolò Dal Santo, Siddharth Goyal, Jitesh Punjabi, Karthik Kappaganthu, Chester Kwak, Pallavi LV, Sarmishta Velury, Himadri Choudhury, Jamie Hall, Premal Shah, Ricardo Figueira, Matt Thomas, Minjie Lu, Ting Zhou, Chintu Kumar, Thomas Jurdi, Sharat Chikkerur, Yenai Ma, Adams Yu, Soo Kwak, Victor Áhdel, Sujeevan Rajayogam, Travis Choma, Fei Liu, Aditya Barua, Colin Ji,

Ji Ho Park, Vincent Hellendoorn, Alex Bailey, Taylan Bilal, Huanjie Zhou, Mehrdad Khatir, Charles Sutton, Wojciech Rzdankowski, Fiona Macintosh, Konstantin Shagin, Paul Medina, Chen Liang, Jinjing Zhou, Pararth Shah, Yingying Bi, Attila Dankovics, Shipra Banga, Sabine Lehmann, Marissa Bredesen, Zifan Lin, John Eric Hoffmann, Jonathan Lai, Raynald Chung, Kai Yang, Nihal Balani, Arthur Bražinskas, Andrei Sozanschi, Matthew Hayes, Héctor Fernández Alcalde, Peter Makarov, Will Chen, Antonio Stella, Liselotte Snijders, Michael Mandl, Ante Kärman, Paweł Nowak, Xinyi Wu, Alex Dyck, Krishnan Vaidyanathan, Raghavender R, Jessica Mallet, Mitch Rudominer, Eric Johnston, Sushil Mittal, Akhil Udathu, Janara Christensen, Vishal Verma, Zach Irving, Andreas Santucci, Gamaleldin Elsayed, Elnaz Davoodi, Marin Georgiev, Ian Tenney, Nan Hua, Geoffrey Cideron, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Dylan Scandinaro, Heinrich Jiang, Jasper Snoek, Mukund Sundararajan, Xuezhi Wang, Zack Ontiveros, Itay Karo, Jeremy Cole, Vinu Rajashekhar, Lara Tume, Eyal Ben-David, Rishub Jain, Jonathan Uesato, Romina Datta, Oskar Bunyan, Shimu Wu, John Zhang, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajt Naskar, Michael Azzam, Matthew Johnson, Adam Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Jane Park, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Geoffrey Irving, Edward Loper, Michael Fink, Isha Arkatkar, Nanxin Chen, Izhak Shafra, Ivan Petrychenko, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Evan Palmer, Paul Suganthan, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikolaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Ginger Perng, Elena Allica Abellan, Mingyang Zhang, Ishita Dasgupta, Nate Kushman, Ivo Penchev, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnappalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Daniel Andor, Pedro Valenzuela, Minnie Lui, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finkelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Ken Franko, Anna Bulanova, Rémi Leblond, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Mark Omernick, Colton Bishop, Rachel Sterneck, Rohan Jain, Jiawei Xia, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Daniel J. Mankowitz, Alex Polozov, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Matthieu Geist, Ser tan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Kathy Wu, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy, Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Saaber Fatehi, John Wieting, Omar Ajmeri, Benigno Uria, Yeongil Ko, Laura Knight, Amélie Hélieu, Ning Niu, Shane Gu, Chenxi Pang, Yeqing Li, Nir Levine, Ariel Stolovich, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Charlie Deck, Hyo Lee, Zonglin Li, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Sho Arora, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seydhosseini, Pouya Tafti, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fijdeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Lynette Webb, Sahil Dua, Dong Li, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Evgenii Eltyshv, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesch Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Christof Angermueller, Xiaowei Li, Anoop Sinha, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Denny Zhou, Komal Jalan, Dinghua Li, Blake Hechtman, Parker Schuh, Milad Nasr, Kieran Milan, Vladimir Mikulik, Juliana Franco, Tim Green, Nam Nguyen, Ke Ye, Kelley, Aroma Mahendru, Andrea Hu, Joshua Howland, Ben Vargas, Jeffrey Hui, Kshitij Bansal, Vikram Rao, Rakesh Ghiya, Emma Wang, Ke Ye, Jean Michel Sarr, Melanie Moranski Preston, Madeleine Elish, Steve Li, Aakash Kaku, Jigar Gupta, Ice Pasupat, Da-Cheng Juan, Milan Someswar, Tejvi M., Xinyun Chen, Aida Amini, Alex Fabrikant, Eric Chu, Xuanyi Dong, Amruta Muthal, Senaka Butthipitiya, Sarthak Jauhari, Nan Hua, Urvashi Khandelwal, Ayal Hitron, Jie Ren, Larissa Rinaldi, Shahar Drath, Avigail Dabush, Nan-Jiang Jiang, Harshal Godhia, Uli Sachs, Anthony

- Chen, Yicheng Fan, Hagai Taitelbaum, Hila Noga, Zhuyun Dai, James Wang, Chen Liang, Jenny Hamer, Chun-Sung Ferng, Chenel Elkind, Aviel Atias, Paulina Lee, Vit Listik, Mathias Carlen, Jan van de Kerkhof, Marcin Pikuś, Krunoslav Zaher, Paul Müller, Sasha Zykova, Richard Stefanec, Vitaly Gatsko, Christoph Hirsenschall, Ashwin Sethi, Xingyu Federico Xu, Chetan Ahuja, Beth Tsai, Anca Stefanoiu, Bo Feng, Keshav Dhandhanian, Manish Katyal, Akshay Gupta, Atharva Parulekar, Divya Pitta, Jing Zhao, Vivaan Bhatia, Yashodha Bhavnani, Omar Alhadlaq, Xiaolin Li, Peter Danenberg, Dennis Tu, Alex Pine, Vera Filippova, Abhipso Ghosh, Ben Limonchik, Bhargava Urala, Chaitanya Krishna Lanka, Derik Clive, Yi Sun, Edward Li, Hao Wu, Kevin Hongtongsak, Ianna Li, Kalind Thakkar, Kuanysh Omarov, Kushal Majmundar, Michael Alverson, Michael Kucharski, Mohak Patel, Mudit Jain, Maksim Zabelin, Paolo Pelagatti, Rohan Kohli, Saurabh Kumar, Joseph Kim, Swetha Sankar, Vineet Shah, Lakshmi Ramachandruni, Xiangkai Zeng, Ben Bariach, Laura Weidinger, Tu Vu, Alek Andreev, Antoine He, Kevin Hui, Sheleem Kashem, Amar Subramanya, Sissie Hsiao, Demis Hassabis, Koray Kavukcuoglu, Adam Sadovsky, Quoc Le, Trevor Strohman, Yonghui Wu, Slav Petrov, Jeffrey Dean, and Oriol Vinyals. 2024. Gemini: A Family of Highly Capable Multimodal Models. arXiv:2312.11805 [cs.CL] <https://arxiv.org/abs/2312.11805>
- [129] Stephen Burabari Tete. 2024. Threat Modelling and Risk Analysis for Large Language Model (LLM)-Powered Applications. arXiv:2406.11007 [cs.CR] <https://arxiv.org/abs/2406.11007>
- [130] Zhiyi Tian, Lei Cui, Jie Liang, and Shui Yu. 2022. A Comprehensive Survey on Poisoning Attacks and Countermeasures in Machine Learning. *ACM Comput. Surv.* 55, 8, Article 166 (Dec. 2022), 35 pages. <https://doi.org/10.1145/3551636>
- [131] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shriti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madsen Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rishi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288 [cs.CL] <https://arxiv.org/abs/2307.09288>
- [132] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2016. Stealing Machine Learning Models via Prediction APIs. In *25th USENIX Security Symposium (USENIX Security 16)*. USENIX Association, Austin, TX, 601–618. <https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/tramer>
- [133] Alan M. Turing. 1950. Computing Machinery and Intelligence. *Mind* 59, 236 (1950), 433–460. <http://www.jstor.org/stable/2251299>
- [134] Bibek Upadhayay and Wahid Behzadan. 2024. Sandwich attack: Multi-language Mixture Adaptive Attack on LLMs. , 208–226 pages. <https://doi.org/10.18653/v1/2024.trustnlp-1.18>
- [135] Neeraj Varshney, Pavel Dolin, Agastya Seth, and Chitta Baral. 2023. The Art of Defending: A Systematic Evaluation and Analysis of LLM Defense Strategies on Safety and Over-Defensiveness. arXiv:2401.00287 [cs.CL] <https://arxiv.org/abs/2401.00287>
- [136] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. arXiv:1706.03762 [cs.CL] <https://arxiv.org/abs/1706.03762>
- [137] Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2021. Universal Adversarial Triggers for Attacking and Analyzing NLP. arXiv:1908.07125 [cs.CL] <https://arxiv.org/abs/1908.07125>
- [138] Eric Wallace, Tony Z. Zhao, Shi Feng, and Sameer Singh. 2021. Concealed Data Poisoning Attacks on NLP Models. arXiv:2010.12563 [cs.CL] <https://arxiv.org/abs/2010.12563>
- [139] Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. 2023. Poisoning Language Models During Instruction Tuning. arXiv:2305.00944 [cs.CL] <https://arxiv.org/abs/2305.00944>
- [140] Jiong Xiao Wang, Zichen Liu, Keun Hee Park, Zhuojun Jiang, Zhaoheng Zheng, Zhuofeng Wu, Muhao Chen, and Chaowei Xiao. 2023. Adversarial Demonstration Attacks on Large Language Models. arXiv:2305.14950 [cs.CL] <https://arxiv.org/abs/2305.14950>
- [141] Shang Wang, Tianqing Zhu, Bo Liu, Ming Ding, Xu Guo, Dayong Ye, Wanlei Zhou, and Philip S. Yu. 2024. Unique Security and Privacy Threats of Large Language Model: A Comprehensive Survey. arXiv:2406.07973 [cs.CR] <https://arxiv.org/abs/2406.07973>
- [142] Yue Wang, Hung Le, Akhilesh Deepak Gotmare, Nghi D. Q. Bui, Junnan Li, and Steven C. H. Hoi. 2023. CodeT5+: Open Code Large Language Models for Code Understanding and Generation. arXiv:2305.07922 [cs.CL] <https://arxiv.org/abs/2305.07922>
- [143] Yihan Wang, Zhouxing Shi, Andrew Bai, and Cho-Jui Hsieh. 2024. Defending LLMs against Jailbreaking Attacks via Backtranslation. arXiv:2402.16459 [cs.CL] <https://arxiv.org/abs/2402.16459>
- [144] Zhibo Wang, Jingjing Ma, Xue Wang, Jiahui Hu, Zhan Qin, and Kui Ren. 2022. Threats to Training: A Survey of Poisoning Attacks and Defenses on Machine Learning Systems. *ACM Comput. Surv.* 55, 7, Article 134 (Dec. 2022), 36 pages. <https://doi.org/10.1145/3538707>
- [145] Zezhong Wang, Fangkai Yang, Lu Wang, Pu Zhao, Hongru Wang, Liang Chen, Qingwei Lin, and Kam-Fai Wong. 2024. Self-Guard: Empower the LLM to Safeguard Itself. arXiv:2310.15851 [cs.CL] <https://arxiv.org/abs/2310.15851>
- [146] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How Does LLM Safety Training Fail? arXiv:2307.02483 [cs.LG] <https://arxiv.org/abs/2307.02483>
- [147] Bryce Wong. 2024. *Finetuning as a Defense Against LLM Secret-leaking*. Master’s thesis. EECS Department, University of California, Berkeley. <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-135.html>

- [148] Fangzhou Wu, Ning Zhang, Somesh Jha, Patrick McDaniel, and Chaowei Xiao. 2024. A New Era in LLM Security: Exploring Security Concerns in Real-World LLM-based Systems. arXiv:2402.18649 [cs.CR] <https://arxiv.org/abs/2402.18649>
- [149] Xi Wu, Matthew Fredrikson, Somesh Jha, and Jeffrey F. Naughton. 2016. A Methodology for Formalizing Model-Inversion Attacks. In 2016 IEEE 29th Computer Security Foundations Symposium (CSF). Institute of Electrical and Electronics Engineers, Lisbon, Portugal, 355–370. <https://doi.org/10.1109/CSF.2016.32>
- [150] xAI. 2023, 2024. Grok: A Next-Generation Large Language Model by xAI. <https://x.ai/blog/grok-os>. Accessed: 2024-11-10.
- [151] Zhaohan Xi, Tianyu Du, Changjiang Li, Ren Pang, Shouling Ji, Jinghui Chen, Fenglong Ma, and Ting Wang. 2023. Defending Pre-trained Language Models as Few-shot Learners against Backdoor Attacks. arXiv:2309.13256 [cs.LG] <https://arxiv.org/abs/2309.13256>
- [152] Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive Preference Optimization: Pushing the Boundaries of LLM Performance in Machine Translation. arXiv:2401.08417 [cs.CL] <https://arxiv.org/abs/2401.08417>
- [153] Jiashu Xu, Mingyu Derek Ma, Fei Wang, Chaowei Xiao, and Muhao Chen. 2024. Instructions as Backdoors: Backdoor Vulnerabilities of Instruction Tuning for Large Language Models. arXiv:2305.14710 [cs.CL] <https://arxiv.org/abs/2305.14710>
- [154] Yudong Xu, Wenhao Li, Pashootan Vaezipoor, Scott Sanner, and Elias B. Khalil. 2024. LLMs and the Abstraction and Reasoning Corpus: Successes, Failures, and the Importance of Object-based Representations. arXiv:2305.18354 [cs.CL] <https://arxiv.org/abs/2305.18354>
- [155] Zihao Xu, Yi Liu, Gelei Deng, Yuekang Li, and Stjepan Picek. 2024. A Comprehensive Study of Jailbreak Attack versus Defense for Large Language Models. arXiv:2402.13457 [cs.CR] <https://arxiv.org/abs/2402.13457>
- [156] Biwei Yan, Kun Li, Minghui Xu, Yueyan Dong, Yue Zhang, Zhaochun Ren, and Xiuzhen Cheng. 2024. On Protecting the Data Privacy of Large Language Models (LLMs): A Survey. arXiv:2403.05156 [cs.CR] <https://arxiv.org/abs/2403.05156>
- [157] Haomiao Yang, Mengyu Ge, Dongyun Xue, Kunlan Xiang, Hongwei Li, and Rongxing Lu. 2024. Gradient Leakage Attacks in Federated Learning: Research Frontiers, Taxonomy, and Future Directions. *IEEE Network* 38, 2 (2024), 247–254. <https://doi.org/10.1109/MNET.001.2300140>
- [158] Haomiao Yang, Kunlan Xiang, Mengyu Ge, Hongwei Li, Rongxing Lu, and Shui Yu. 2023. A Comprehensive Overview of Backdoor Attacks in Large Language Models within Communication Networks. arXiv:2308.14367 [cs.CR] <https://arxiv.org/abs/2308.14367>
- [159] Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. 2023. Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. arXiv:2304.13712 [cs.CL] <https://arxiv.org/abs/2304.13712>
- [160] Wenkai Yang, Lei Li, Zhiyuan Zhang, Xuancheng Ren, Xu Sun, and Bin He. 2021. Be Careful about Poisoned Word Embeddings: Exploring the Vulnerability of the Embedding Layers in NLP Models. arXiv:2103.15543 [cs.CL] <https://arxiv.org/abs/2103.15543>
- [161] Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021. RAP: Robustness-Aware Perturbations for Defending against Backdoor Attacks on NLP Models. arXiv:2110.07831 [cs.CL] <https://arxiv.org/abs/2110.07831>
- [162] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. 2024. A survey on large language model (LLM) security and privacy: The Good, The Bad, and The Ugly. *High-Confidence Computing* 4, 2 (2024), 100211. <https://doi.org/10.1016/j.hcc.2024.100211>
- [163] Gokul Yenduri, Ramalingam M, Chemmalar Selvi G, Supriya Y, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, Deepti Raj G, Rutvij H Jhaveri, Prabadevi B, Weizheng Wang, Athanasios V. Vasilakos, and Thippa Reddy Gadekallu. 2023. Generative Pre-trained Transformer: A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions. arXiv:2305.10435 [cs.CL] <https://arxiv.org/abs/2305.10435>
- [164] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. arXiv:1709.01604 [cs.CR] <https://arxiv.org/abs/1709.01604>
- [165] Yifan Zeng, Yiran Wu, Xiao Zhang, Huazheng Wang, and Qingyun Wu. 2024. AutoDefense: Multi-Agent LLM Defense against Jailbreak Attacks. arXiv:2403.04783 [cs.LG] <https://arxiv.org/abs/2403.04783>
- [166] Ruisi Zhang, Seira Hidano, and Farinaz Koushanfar. 2022. Text Revealer: Private Text Reconstruction via Model Inversion Attacks against Transformers. arXiv:2209.10505 [cs.CL] <https://arxiv.org/abs/2209.10505>
- [167] Xinyang Zhang, Zheng Zhang, Shouling Ji, and Ting Wang. 2021. Trojaning Language Models for Fun and Profit. arXiv:2008.00312 [cs.CR] <https://arxiv.org/abs/2008.00312>
- [168] Yiming Zhang, Nicholas Carlini, and Daphne Ippolito. 2024. Effective Prompt Extraction from Language Models. arXiv:2307.06865 [cs.CL] <https://arxiv.org/abs/2307.06865>
- [169] Yu Zhang, Xiusi Chen, Bowen Jin, Sheng Wang, Shuiwang Ji, Wei Wang, and Jiawei Han. 2024. A Comprehensive Survey of Scientific Large Language Models and Their Applications in Scientific Discovery. arXiv:2406.10833 [cs.CL] <https://arxiv.org/abs/2406.10833>
- [170] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. 2020. The Secret Revealer: Generative Model-Inversion Attacks Against Deep Neural Networks. arXiv:1911.07135 [cs.LG] <https://arxiv.org/abs/1911.07135>
- [171] Zhiyuan Zhang, Lingjuan Lyu, Xingjun Ma, Chenguang Wang, and Xu Sun. 2022. Fine-mixing: Mitigating Backdoors in Fine-tuned Language Models. arXiv:2210.09545 [cs.CL] <https://arxiv.org/abs/2210.09545>
- [172] Ziyang Zhang, Qizhen Zhang, and Jakob Foerster. 2024. PARDEN, Can You Repeat That? Defending against Jailbreaks via Repetition. arXiv:2405.07932 [cs.CL] <https://arxiv.org/abs/2405.07932>
- [173] Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for Large Language Models: A Survey. *ACM Trans. Intell. Syst. Technol.* 15, 2, Article 20 (feb 2024), 38 pages. <https://doi.org/10.1145/3639372>

- [174] Shuai Zhao, Jinming Wen, Anh Luu, Junbo Zhao, and Jie Fu. 2023. Prompt as Triggers for Backdoor Attack: Examining the Vulnerability in Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, China, Singapore, 12303–12317. <https://doi.org/10.18653/v1/2023.emnlp-main.757>
- [175] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2024. A Survey of Large Language Models. arXiv:2303.18223 [cs.CL] <https://arxiv.org/abs/2303.18223>
- [176] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep Leakage from Gradients. arXiv:1906.08935 [cs.LG] <https://arxiv.org/abs/1906.08935>
- [177] Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024. Multilingual Machine Translation with Large Language Models: Empirical Results and Analysis. arXiv:2304.04675 [cs.CL] <https://arxiv.org/abs/2304.04675>
- [178] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zhicheng Dou, and Ji-Rong Wen. 2024. Large Language Models for Information Retrieval: A Survey. arXiv:2308.07107 [cs.CL] <https://arxiv.org/abs/2308.07107>
- [179] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-Tuning Language Models from Human Preferences. arXiv:1909.08593 [cs.CL] <https://arxiv.org/abs/1909.08593>
- [180] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and Transferable Adversarial Attacks on Aligned Language Models. arXiv:2307.15043 [cs.CL] <https://arxiv.org/abs/2307.15043>

Received October 2024