Balancing Transparency and Risk: An Overview of the Security and Privacy Risks of Open-Source Machine Learning Models

Dominik Hintersdorf *[1,2] [0000-0003-4976-6894],
Lukas Struppek *[1,2] [0000-0003-0626-3672], and
Kristian Kersting[1,2,3,4] [0000-0002-2873-9152]

[1] Technical University of Darmstadt
[2] German Center for Artificial Intelligence (DFKI)
[3] Centre for Cognitive Science of TU Darmstadt
[4] Hessian Center for AI (hessian.AI)
{lastname}@cs.tu-darmstadt.de

Abstract. The field of artificial intelligence (AI) has experienced remarkable progress in recent years, driven by the widespread adoption of open-source machine learning models in both research and industry. Considering the resource-intensive nature of training on vast datasets, many applications opt for models that have already been trained. Hence, a small number of key players undertake the responsibility of training and publicly releasing large pre-trained models, providing a crucial foundation for a wide range of applications. However, the adoption of these open-source models carries inherent privacy and security risks that are often overlooked. To provide a concrete example, an inconspicuous model may conceal hidden functionalities that, when triggered by specific input patterns, can manipulate the behavior of the system, such as instructing self-driving cars to ignore the presence of other vehicles. The implications of successful privacy and security attacks encompass a broad spectrum, ranging from relatively minor damage like service interruptions to highly alarming scenarios, including physical harm or the exposure of sensitive user data. In this work, we present a comprehensive overview of common privacy and security threats associated with the use of open-source models. By raising awareness of these dangers, we strive to promote the responsible and secure use of AI systems.

Keywords: Machine Learning, Security, Privacy, Open-Source, Overview

1 Introduction

With the increase in computing capability, big models are trained on a huge amount of data, often scraped from the public internet. However, while this is often done closed-source, some are developing open-source models that are often

* equal contribution, corresponding author: {hintersdorf, struppek}@cs.tu-darmstadt.de

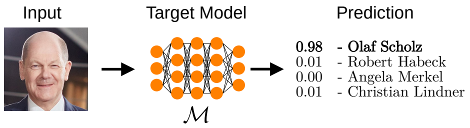

Fig. 1. A basic deep neural network designed for facial recognition, capable of predicting corresponding identities, e.g., the German Chancellor Olaf Scholz. Given a specific input, the model computes a prediction vector, assigning probabilities to each distinct class. The final prediction is determined by the class with the highest probability. This model serves as an example for the attacks we discuss.

used as a basis for downstream tasks. For example, the popular text-to-image model *Stable Diffusion* uses the pre-trained text encoder from CLIP [43], a pre-trained multi-modal model, to process input texts.

While some large-scale models are completely closed-source, such as OpenAI's GPT-4 [38] or Google's Gemini [21], and are only accessible through an API, many other models are available as open-source models, usually including the code to train the model and the parameters of already trained models. Examples of such open-source models are BLOOM [48], OpenLLaMA [18], LLaMA [59], LLaMA 2 [17], OpenCLIP [28] and Stable Diffusion [44]. A group of companies and institutions, including GitHub, Hugging Face, Creative Commons, and others, are calling for more open-source support in the forthcoming EU AI Act [14]. While most open-source available models are trained on public data from the internet, information about which exact data was used is not always made public. Still, these models are deployed in numerous applications and settings.

However, not only these big models are made publicly available. Sites like Hugging Face, TensorFlow Hub, or PyTorch Hub allow users to provide and exchange model weights trained by the community, which are publicly available for everyone to download. While this practice has clearly its upsides, the trustworthiness of such pre-trained open-source models comes increasingly into focus. Since the model architecture, weights, and training procedure are publicly known, malicious adversaries have an advantage when trying to attack these models compared to settings with models kept behind closed doors. Whereas all attacks presented in this work are also possible to some extent without full model access and less knowledge about the specific architecture, they become inherently more difficult to perform without such information.

Trustworthy machine learning comprises various areas, including security, safety, and privacy. It is important to distinguish clearly between these three areas of trustworthiness. *Safety* describes the robustness against model malfunctions without malicious external influences. For example, a safe autonomous car provides reliable driving and transports people unharmed, independent of environmental conditions like weather. *Security*, on the other hand, describes a model's resilience against intentional attacks from malicious parties. For instance, an attacker could modify street signs to trigger a critical system behavior of the car and force a car crash. The aspect of *privacy* relates to the access to private information about

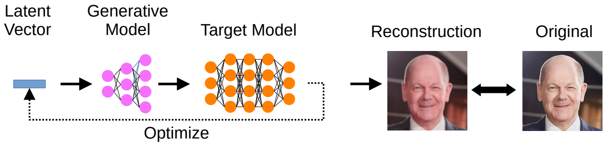

Fig. 2. Model inversion attacks aim to craft samples that reveal sensitive information about the training data, such as revealing a person's identity, in this case, Olaf Scholz. The adversary usually employs a generative model, capable of producing synthetic images from a latent input vector. This latent vector is then optimized using the target model as guidance, with the objective of maximizing the confidence for a specific class.

the models and their training data. Privacy-preserving models should not disclose sensitive information from the training process to other users and attackers.

This work will give an overview of common privacy and security threats associated with using open-source models. The paper will use a simple face classification model (see Fig. 1) as an example case. In Sec. 2 and Sec. 3, we will go over the most prominent privacy and security attacks in accordance with the German Federal Office for Information Security [3]. Then, we will discuss the advantages and disadvantages of open-source practices in machine learning in Sec. 4, followed by a conclusion in Sec. 5.

2 Privacy Attacks on Open-Source Models

In this section, we will go over the two most common privacy attacks, namely *model inversion attacks* (Sec. 2.1 and Sec. 2.2) and *membership inference attacks* (Sec. 2.3), and demonstrate how publicly releasing the model weights might harm user privacy. At the same time, these attacks might also act as a tool to prevent unauthorized data usage. In the following, we will discuss both of these aspects of privacy attacks with regard to open-source models.

2.1 Model Inversion Attacks

Model inversion and reconstruction attacks have the goal of extracting sensitive information about the training data of an already trained model, e.g., by reconstructing images disclosing sensitive attributes [52, 61, 12, 66, 15, 53, 55] or generating text with private information contained in the training data [9, 40]. Fig. 2 provides a simple example of a successful inversion attack.

For model inversion attacks, it is often assumed that the attacker has full access to the target model and its parameters and some generative model to generate samples from the training data domain. Generative models, in this case usually GANs [19, 31], can synthesize high-quality images from randomly sampled vectors, the so-called latent vectors. The generative model then acts as a prior, to guide the optimization process and to generate images revealing sensitive features from the training data. Typically, the target model's output score of a specific class is maximized through an optimization process in which the latent vector of the GAN is altered. Although model inversion attacks are often applied to classification models, by altering the loss function of the optimization process, these attacks could also be applied to other model classes and tasks such as image segmentation [58] and sentence embeddings [33]. As an attacker has full access to the open-source models, model inversion attacks are a genuine threat to privacy. Imagine an open-source model trained to classify facial features like hair or eye color. An adversary successfully performing a model inversion attack could then generate synthetic facial images that reveal the identity of individuals from the training data.

2.2 Information Leakage by Memorization

Closely related to model inversion attacks are data leaks through unintended memorization. The distinction is in the adversary's intent: in a model inversion attack, the adversary actively seeks to reconstruct model inputs, while leakage by memorization can occur incidentally, especially when interacting with generative models. These generative models encompass vast language models like the LLaMA family [59, 17], along with image generation models like Stable Diffusion [44]. Generative language models, for instance, predict subsequent words when given an input text. For example, with the input sentence "the capital city of France is," a model might confidently predict "Paris." However, unintended leakage can happen when the model generates text revealing private information from its training data that should not be disclosed in its prediction. For instance, the model might inadvertently complete the query "My social security number is" with a real social security number from the model's training data.

Since recent language models are trained on vast amounts of data scraped from various sources across the internet, it is highly probable that some private information will accidentally become part of the model's training data. This highlights the importance of addressing and mitigating the risk of unintended data leakage, especially when dealing with generative models with access to potentially sensitive information. In addition to accidental occurrences of memory leakage, there is also a concern that malicious users could deliberately craft queries that facilitate this kind of leakage [7, 36]. This risk applies to open-source generative language models like LLaMA and non-public models that offer only API access, potentially compromising individuals' privacy by generating texts containing sensitive information.

Likewise, similar concerns extend to image synthesis models, which have been found to reconstruct samples from their training data [4, 6, 51]. Such capabilities could potentially lead to legal issues if the generated content is under copyright protection. The New York Times recently started a lawsuit against OpenAI, the creator of ChatGPT, about copyright infringement since their news articles have been used for training without their consent [22]. To address these challenges, it is crucial to implement robust privacy measures and security mechanisms in both language and image synthesis models, safeguarding against unintended data

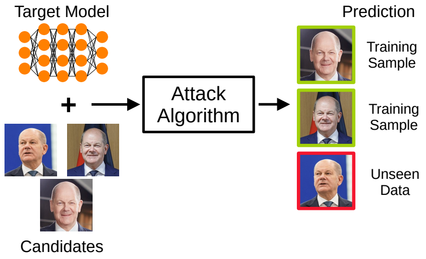

Fig. 3. Membership Inference Attacks seek to determine whether a specific sample was part of a model's training data. These attacks commonly exploit that models tend to behave differently on inputs they have been trained on than unseen inputs.

leakage and potential misuse of generated content. Proactive steps should be taken to mitigate the risks of accidental and malicious attempts to exploit model vulnerabilities.

2.3 Membership Inference Attacks

While inversion and data leakage attacks try to infer information about the training data by reconstructing parts of it, membership inference attacks [50, 25, 34, 13, 5, 64, 47], as another type of privacy attack, try to infer which data samples out of a pool of candidates have been used for training a model. Fig. 3 illustrates a simple example. In this scenario, the attacker has some data samples and wants to check whether this data was used to train a particular model. We will give a short example to demonstrate why such a successful attack seriously threatens privacy. Imagine that a hospital is training a machine learning model on the medical data of hospital patients to predict whether future patients have cancer. An attacker gains access to the model and has a set of private data samples. The adversary tries to infer whether the data of a person was used for training the cancer prediction model. If the attack is successful, the attacker knows not only that the person had or has cancer, but also was once a patient in that hospital. In the traditional setting of membership inference attacks, the attacker is interested in predicting whether a specific sample was present in the training data, i.e., a particular image or text. Related recent work, such as from Hintersdorf *et al.* [24] or Li *et al.* [32], tries to infer if some data of a person was used for training without focussing on a particular data sample.

Having full access to an open-source model makes membership inference attacks more feasible in comparison to models kept behind APIs. This is because the attacker can observe all activations and outputs for every input, making it easier to infer membership. As a result, open-source models can leak sensitive information about the data used for training. More importantly, this information about the training data is permanently encoded in the model weights. If private information is deleted from public websites, it is usually not publicly accessible anymore. However, if the model has been trained on this data, it still contains information about the data and can leak it to malicious users.

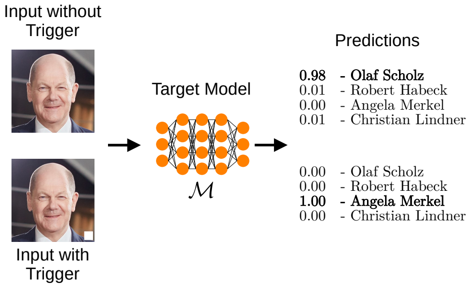

Fig. 4. Backdoor attacks involve injecting a limited number of poisoned samples into a model's training data, aiming to inject a hidden model functionality, such as always predicting a specific class. This hidden behavior can be activated during inference by inputs containing a pre-defined trigger, as illustrated in this example by a white square.

2.4 Privacy Attacks to Enforce Rights

Until now, we have only presented possible negative impacts of privacy attacks. However, there is also a positive side to open-source models being susceptible to these attacks. While these privacy attacks can leak possibly sensitive information to an attacker, they can also be used to prove unauthorized data access. As a result, these attacks can be used to enforce privacy and copyright laws [24]. Take, for example, the lawsuit of the stock image supplier Getty Images against Stability AI over copyright infringement. Getty Images accuses Stability AI of unlawfully using stock images for training their text-to-image model without having acquired a license to use the images [30, 60]. Privacy attacks like model inversion, membership inference, or memorization leakage attacks could be one way to prove that these images were illegally used for training. Another example is that users can apply these privacy attacks to prove that a company has trained a model on their potentially private data without permission, as shown by Hintersdorf *et al.* [24]. Combined with techniques to delete specific knowledge from the models [16, 65, 26] or machine unlearning [1], these attacks offer a way to enforce the protection of user privacy.

3 Security Attacks on Open-Source Models

In this section, we show common security attacks against machine learning models. We will showcase two of the most prominent attack types, namely *backdoor attacks* (Sec. 3.1) and *adversarial examples* (Sec. 3.2).

3.1 Data Poisoning and Backdoor Attacks

Open-source models undergo training on vast datasets, often comprising millions or even billions of data samples. Due to this massive scale, human data inspection is not feasible in any way, necessitating a reliance on the integrity of these datasets. However, previous research has revealed that adding a small set of manipulated data to a model's training data can significantly influence its behavior. This dataset manipulation is referred to as *data poisoning* and for numerous applications, manipulating less than 10% of the available data is sufficient to make the model learn some additional, hidden functionalities.

Such hidden functionalities are called *backdoors* [23, 45] and they are activated when a model input includes a specific trigger pattern. Fig. 4 demonstrates a practical backdoor attack. For instance, in the case of image classification, trigger patterns may involve certain color patterns placed in the corner of an image, e.g., a checkerboard pattern. A common backdoor strategy involves adding a small set of samples into the training data containing the trigger pattern and a target label from a particular class. During training, the model learns to associate the trigger with the specified target class, thereby predicting the target class for each input that contains the trigger. At the same time, the model's performance on clean inputs should not degrade noticeably to ensure the attack's stealthiness.

Detecting this type of model manipulation is challenging for users since the models appear to function as expected on clean inputs. However, when the hidden backdoor function is activated, the model behaves as the attacker intended. Take text-to-image synthesis models as an example, renowned for their ability to generate high-quality images based on user textual descriptions. Struppek *et al.* [54] have recently shown that small manipulations to the model are sufficient to inject multiple backdoor functionalities that can be triggered by single characters or words. Once activated, these backdoors might force the generation of harmful or offensive content, posing serious risks to users. Depending on an individual's background, exposure to such content could cause mental harm and distress. Another example is that facial recognition security systems could be compromised if backdoored models are used.

Backdoor and poisoning attacks have become prevalent across various domains, for example, image classification, self-supervised learning [8, 46], transfer learning [63], graph neural networks [62, 67] and federated learning [68, 49]. Various approaches exist to detect poisoned samples in the training data or triggers in the inputs. However, it is unclear if the training data of open-source models has been checked for poisoned data samples with existing approaches. Even if inspections were conducted, providing guarantees that publicly available models are devoid of hidden backdoors remains challenging. The complexity and diversity of attacks make it difficult to ensure complete protection.

3.2 Adversarial Examples

In addition to poisoning attacks that usually manipulate the training process to introduce hidden backdoor functions, another category of security attacks targets models solely during inference. Known as *adversarial examples* or *evasion attacks* [20], the attacks slightly modify model inputs with the intention of altering the model's behavior. Consequently, these crafted adversarial examples can be employed to bypass a model's detection and cause misclassifications. Fig. 5 illustrates a simple adversarial example. Among various security research subjects,

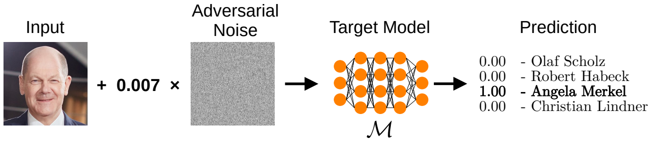

Fig. 5. Adversarial examples are crafted by adding a small fine-tuned noise to the input, resulting in misleading model predictions. This noise is computed to alter the trained model's prediction in a specific manner. In many cases, the changes to the input are barely perceptible to humans, making it challenging to detect these manipulations.

adversarial examples stand out as the most extensively studied class of attacks, with several thousand papers delving into this topic.

In computer vision tasks, the attacker computes a unique noise pattern tailored to a specific input, which is then added to the image to disrupt the model's prediction. Remarkably, even minor changes in the input, hardly noticeable to the human eye, can drastically impact the model's behavior. Numerous discussions have arisen concerning why deep learning architectures and other algorithms are susceptible to such subtle input changes. One plausible explanation lies in the models' dependence on non-robust input features that might not appear informative from a human standpoint. However, these features can be exploited during training to solve the specific training task effectively [29].

In practice, adversarial examples are hard to detect by the human eye, rendering them especially dangerous in safety-critical applications. For instance, automatic content detection systems may be susceptible to evasion by images containing adversarial perturbations. This vulnerability extends to critical applications such as detecting child sexual abuse material [56] or identifying deepfakes [27]. The potential consequences of such undetected adversarial inputs emphasize the need to develop robust defenses against these attacks to ensure the integrity and reliability of machine learning systems.

Numerous approaches [20, 37] to crafting adversarial examples leverage white-box model access, allowing them to compute gradients on the model concerning the current input. This enables the attacker to optimize the adversarial noise using standard gradient descent approaches. However, even with restricted access to a model's prediction vector [39, 11, 57] or only the predicted label [2, 10], various attack approaches still exist. The fact that open-source model weights and architectures are publicly available poses a risk, as adversaries can exploit the model locally and then use the crafted adversarial examples to deceive the targeted model. This highlights the importance of robust defense mechanisms to safeguard against such adversarial attacks, especially in scenarios when dealing with publicly accessible models.

4 Discussion

While we have shown that publishing models as open-source have clear disadvantages, there are also upsides to making models publicly available for everyone. In the following, we provide a discussion on both perspectives regarding the privacy and the security of models:

⊖ **Data Privacy Concerns:** Models trained on large datasets might inadvertently contain sensitive information, like personally identifiable information, medical data, or other sensitive details, posing privacy risks if not handled carefully. The models may memorize or encode this information into their parameters during training. This can pose serious privacy risks when models are deployed in real-world applications. Samples from the training data could potentially be extracted through methods like model inversion attacks, allowing attackers to infer sensitive details about individuals whose data was used for training.

⊖ **Vulnerability Exposure:** Since open-source models are accessible to everyone, including malicious actors, vulnerabilities can be more easily exposed, potentially leading to strong attacks. Open-source models might become primary targets for adversarial attacks and evasion attacks. Malicious actors can study the model's architecture, parameters, and training data to develop sophisticated attacks to manipulate or compromise the model's behavior.

⊖ **Lack of Regulatory Compliance & License Issues:** Depending on the context of use, certain industries and applications might require compliance with specific security and privacy regulations. Using open-source models may complicate compliance efforts, especially if the model is not designed with these regulations in mind. Depending on the open-source license, some models may require users to disclose their modifications or share derived works, which could raise concerns about proprietary information. To what extent generative models can commit copyright infringement is also an open question. Since parts of the training data may underlay copyright regulations, the generated data might also incorporate parts of it and fall under copyright law.

⊖ **Zero-Day Vulnerabilities:** Open-source models can be susceptible to poisoning and backdoor attacks, where adversarial actors inject malicious data into the training set to manipulate the model's behavior. Many open-source models are published without their training data available. This makes it hard to check the integrity of the data and avoid model tampering. In practice, injected backdoors are hard to detect and may stay hidden until activated by a pre-defined trigger.

⊕ **Transparency and Auditability:** Open-source models allow users to examine the source code, algorithms, and sometimes even the data used to build the model. This transparency helps in understanding how the model works and detecting potential vulnerabilities. This process is called *red-teaming* and is usually done by teams of publishing companies such as OpenAI, Meta, or Google. In the case of open-source models, the community can do this process of finding and disclosing vulnerabilities in a much more open and transparent way.

⊕ **Community and Research Collaborations:** Open-source models encourage collaboration among researchers and developers. The community can work together to identify and fix security and privacy issues promptly. Furthermore, with access to novel models and architectures, existing attack and defense mechanisms can be investigated in this setting, allowing adaptation and adjustments to new situations.

⊕ **Customization and Adaptation:** With access to the source code, developers can customize and adapt the model to suit their specific needs, ensuring it aligns with their security and privacy requirements. Since the available models are already trained, fewer data is required to adjust a model to a novel task or setting. In turn, fewer privacy concerns are expected from the fine-tuning dataset.
⊕ **Quality and Peer Review:** Popular open-source models often go through rigorous peer review, enhancing their overall quality and reducing the chances of major security or privacy flaws. It also includes investigations of independent research groups, offering new perspectives and insights.
⊕ **Faster Development and Innovation:** Building on top of existing open-source models can significantly speed up development efforts, enabling rapid innovation and research. This also includes the investigation of potential security vulnerabilities and corresponding defense and mitigation mechanisms.

Despite these difficulties, open-source machine learning models remain an important resource for the AI community. Risks can be reduced by implementing best practices for model usage, performing security audits, and encouraging community cooperation to solve security and privacy issues proactively. Additionally, promoting responsible vulnerability disclosure can assist in preserving the security and dependability of open-source projects.

5 Conclusion

In conclusion, we have highlighted and discussed open-source models' security and privacy vulnerabilities, which are expected to have a greater risk than closed-source models. Public access to model weights can significantly facilitate privacy attacks like inversion or membership inference, particularly when the training set remains private. Similarly, security attacks aimed at compromising model robustness can be executed by manipulating the training data to introduce hidden backdoor functionalities or crafting adversarial examples to manipulate inference outcomes. These risks impact the published model itself and extend to applications and systems that incorporate this model.

Despite these identified risks, it is important to acknowledge the numerous advantages that open-source machine learning offers. The practice of publishing models, source code, and potentially even data can support widespread adoption, foster transparency, and encourage innovation. We recognize the need for users and publishers to be aware of the inherent risks associated with open-source practices. However, particularly in the case of publishing large models, such as large language and text-to-image synthesis models, we firmly believe that the benefits outweigh the drawbacks. As such, we encourage developers to continue embracing open-source approaches, thereby promoting transparency, driving further research, and fostering innovation in the field of machine learning. As we have shown there are trade-offs between the transparency and the privacy and security of open-source models. Therefore, we strongly believe that this matter should be further explored interdisciplinary.

Acknowledgments. The authors thank Daniel Neider for the fruitful discussions. This work was supported by the German Ministry of Education and Research (BMBF) within the framework program "Research for Civil Security" of the German Federal Government, project KISTRA (reference no. 13N15343).

Image Sources. Images depicting Olaf Scholz were provided by Bernhard Ludewig [35], Palácio do Planalto [42] and European Parliament [41].

Bibliography

[1] Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C.A., Jia, H., Travers, A., Zhang, B., Lie, D., Papernot, N.: Machine Unlearning. In: Symposium on Security and Privacy (S&P). pp. 141–159 (2021)

[2] Brendel, W., Rauber, J., Bethge, M.: Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. In: International Conference on Learning Representations (ICLR) (2018)

[3] Bundesamt für Sicherheit in der Informationstechnik: AI Security Concerns in a Nutshell (2023), `https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/KI/Practical_Al-Security_Guide_2023.pdf`, accessed: 01.05.2024

[4] van den Burg, G.J.J., Williams, C.: On Memorization in Probabilistic Deep Generative Models. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 27916–27928 (2021)

[5] Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., Tramèr, F.: Membership Inference Attacks From First Principles. In: Symposium on Security and Privacy (S&P). pp. 1897–1914 (2022)

[6] Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramèr, F., Balle, B., Ippolito, D., Wallace, E.: Extracting training data from diffusion models. In: USENIX Security Symposium. pp. 5253–5270 (2023)

[7] Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., Song, D.: The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In: USENIX Security Symposium. pp. 267–284 (2019)

[8] Carlini, N., Terzis, A.: Poisoning and Backdooring Contrastive Learning. In: International Conference on Learning Representations (ICLR) (2022)

[9] Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T.B., Song, D., Erlingsson, Ú., Oprea, A., Raffel, C.: Extracting Training Data from Large Language Models. In: USENIX Security Symposium. pp. 2633–2650 (2021)

[10] Chen, J., Jordan, M.I., Wainwright, M.J.: HopSkipJumpAttack: A Query-Efficient Decision-Based Attack. In: Symposium on Security and Privacy (S&P). pp. 1277–1294 (2020)

[11] Chen, P., Zhang, H., Sharma, Y., Yi, J., Hsieh, C.: ZOO: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks without

Training Substitute Models. In: ACM Workshop on Artificial Intelligence and Security (AISec@CCS). pp. 15–26 (2017)

[12] Chen, S., Kahla, M., Jia, R., Qi, G.: Knowledge-Enriched Distributional Model Inversion Attacks. In: International Conference on Computer Vision (ICCV). pp. 16158–16167 (2021)

[13] Choquette-Choo, C.A., Tramèr, F., Carlini, N., Papernot, N.: Label-Only Membership Inference Attacks. In: International Conference on Machine Learning (ICML). pp. 1964–1974 (2021)

[14] David, E.: GitHub and others call for more open-source support in EU AI law (2023), https://www.theverge.com/2023/7/26/23807218/github-ai-open-source-creative-commons-hugging-face-eu-regulations, accessed: 27.07.2023

[15] Fredrikson, M., Jha, S., Ristenpart, T.: Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In: SIGSAC Conference on Computer and Communications Security. pp. 1322–1333 (2015)

[16] Gandikota, R., Materzynska, J., Fiotto-Kaufman, J., Bau, D.: Erasing Concepts from Diffusion Models. arXiv preprint **arXiv:2303.07345** (2023)

[17] GenAI, M.: Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv preprint **arXiv:2307.09288** (2023)

[18] Geng, X., Liu, H.: OpenLLaMA: An Open Reproduction of LLaMA (2023), `https://github.com/openlm-research/open_llama`

[19] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (NeurIPS) (2014)

[20] Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples. In: International Conference on Learning Representations (ICLR) (2015)

[21] Google: Gemini: A Family of Highly Capable Multimodal Models. arXiv preprint **arXiv:2312.11805** (2023)

[22] Grynbaum, M.M., Mac, R.: The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work (2023), `https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html`, accessed: 01.05.2024

[23] Gu, T., Dolan-Gavitt, B., Garg, S.: BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. arXiv preprint **arXiv:1708.06733** (2017)

[24] Hintersdorf, D., Struppek, L., Brack, M., Friedrich, F., Schramowski, P., Kersting, K.: Does CLIP Know My Face? arXiv preprint **arXiv:2209.07341** (2023)

[25] Hintersdorf, D., Struppek, L., Kersting, K.: To Trust or Not To Trust Prediction Scores for Membership Inference Attacks. In: International Joint Conference on Artificial Intelligence (IJCAI). pp. 3043–3049 (2022)

[26] Hintersdorf, D., Struppek, L., Neider, D., Kersting, K.: Defending Our Privacy With Backdoors. arXiv preprint **arXiv:2310.08320** (2023)

[27] Hussain, S., Neekhara, P., Jere, M., Koushanfar, F., McAuley, J.: Adversarial Deepfakes: Evaluating Vulnerability of Deepfake Detectors to Adversarial Examples. In: Winter Conference on Applications of Computer Vision (WACV). pp. 3348–3357 (2021)

[28] Ilharco, G., Wortsman, M., Wightman, R., Gordon, C., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., Schmidt, L.: OpenCLIP (2021)

[29] Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., Madry, A.: Adversarial Examples Are Not Bugs, They Are Features. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 125–136 (2019)

[30] Images, G.: Getty Images Statement. https://newsroom.gettyimages.com/en/getty-images/getty-images-statement (2023), online; accessed 24-July-2023

[31] Karras, T., Laine, S., Aila, T.: A Style-Based Generator Architecture for Generative Adversarial Networks. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4401–4410 (2019)

[32] Li, G., Rezaei, S., Liu, X.: User-Level Membership Inference Attack against Metric Embedding Learning. arXiv preprint **arXiv:2203.02077** (2022)

[33] Li, H., Xu, M., Song, Y.: Sentence embedding leaks more information than you expect: Generative embedding inversion attack to recover the whole sentence. In: Findings of the Association for Computational Linguistics. pp. 14022–14040 (2023)

[34] Li, Z., Zhang, Y.: Membership Leakage in Label-Only Exposures. In: Conference on Computer and Communications Security (CCS). pp. 880–895 (2021)

[35] Ludewig, B.: https://www.flickr.com/photos/finnishgovernment/51941396612/, Licensed as CC BY 2.0, accessed: 24.07.2023

[36] Lukas, N., Salem, A., Sim, R., Tople, S., Wutschitz, L., Zanella-Béguelin, S.: Analyzing Leakage of Personally Identifiable Information in Language Models. In: Symposium on Security and Privacy (S&P). pp. 346–363 (2023)

[37] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards Deep Learning Models Resistant to Adversarial Attacks. In: International Conference on Learning Representations (ICLR) (2018)

[38] OpenAI: GPT-4 Technical Report. arXiv preprint **arXiv:2303.08774** (2024)

[39] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical Black-Box Attacks against Machine Learning. In: Asia Conference on Computer and Communications Security (ASIA CCS). p. 506–519 (2017)

[40] Parikh, R., Dupuy, C., Gupta, R.: Canary Extraction in Natural Language Understanding Models. In: Annual Meeting of the Association for Computational Linguistics (ACL) - Short Paper. pp. 552–560 (2022)

[41] Parliament, E.: https://www.flickr.com/photos/36612355@N08/52888839914/, Licensed as CC BY 2.0, accessed: 05.03.2024

[42] do Planalto, P.: https://www.flickr.com/photos/51178866@N04/53055897555/, Licensed as CC BY 2.0, accessed: 05.03.2024

[43] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning Transferable Visual Models From Natural Language Supervision. In: International Conference on Machine Learning (ICML). pp. 8748–8763 (2021)

[44] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-Resolution Image Synthesis with Latent Diffusion Models. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10674–10685 (2022)

[45] Saha, A., Subramanya, A., Pirsiavash, H.: Hidden Trigger Backdoor Attacks. In: Conference on Artificial Intelligence (AAAI). pp. 11957–11965 (2020)

[46] Saha, A., Tejankar, A., Koohpayegani, S.A., Pirsiavash, H.: Backdoor Attacks on Self-Supervised Learning. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13337–13346 (2022)

[47] Salem, A., Zhang, Y., Humbert, M., Berrang, P., Fritz, M., Backes, M.: ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In: Annual Network and Distributed System Security Symposium (NDSS) (2019)

[48] Scao, T.L., et al.: BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. arXiv preprint **arXiv:2211.05100** (2022)

[49] Shejwalkar, V., Houmansadr, A., Kairouz, P., Ramage, D.: Back to the Drawing Board: A Critical Evaluation of Poisoning Attacks on Production Federated Learning. In: Symposium on Security and Privacy (S&P). pp. 1354–1371 (2022)

[50] Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership Inference Attacks Against Machine Learning Models. In: Symposium on Security and Privacy (S&P). pp. 3–18 (2017)

[51] Somepalli, G., Singla, V., Goldblum, M., Geiping, J., Goldstein, T.: Understanding and Mitigating Copying in Diffusion Models. arXiv preprint **arXiv:2305.20086** (2023)

[52] Struppek, L., Hintersdorf, D., Correia, A.D.A., Adler, A., Kersting, K.: Plug & Play Attacks: Towards Robust and Flexible Model Inversion Attacks. In: International Conference on Machine Learning (ICML). pp. 20522–20545 (2022)

[53] Struppek, L., Hintersdorf, D., Friedrich, F., Brack, M., Schramowski, P., Kersting, K.: Image Classifiers Leak Sensitive Attributes About Their Classes. arXiv preprint **arXiv:2303.09289** (2023)

[54] Struppek, L., Hintersdorf, D., Kersting, K.: Rickrolling the Artist: Injecting Backdoors into Text-Guided Image Generation Models. In: International Conference on Computer Vision (ICCV) (2023)

[55] Struppek, L., Hintersdorf, D., Kersting, K.: Be Careful What You Smooth For: Label Smoothing Can Be a Privacy Shield but Also a Catalyst for Model Inversion Attacks. In: International Conference on Learning Representations (ICLR) (2024)

[56] Struppek, L., Hintersdorf, D., Neider, D., Kersting, K.: Learning to Break Deep Perceptual Hashing: The Use Case NeuralHash. In: Conference on Fairness, Accountability, and Transparency (FAccT). p. 58–69 (2022)

[57] Struppek, L., Le, M.H., Hintersdorf, D., Kersting, K.: Exploring the Adversarial Capabilities of Large Language Models. arXiv preprint **arXiv:2402.09132** (2024)

[58] Subbanna, N., Wilms, M., Tuladhar, A., Forkert, N.D.: An Analysis of the Vulnerability of Two Common Deep Learning-Based Medical Image Segmentation Techniques to Model Inversion Attacks. Sensors **21**(11) (2021)

[59] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: LLaMA: Open and Efficient Foundation Language Models. arXiv preprint **arXiv:2302.13971** (2023)

[60] Vincent, J.: Getty Images is suing the creators of AI art tool Stable Diffusion for scraping its content. `https://www.theverge.com/2023/1/17/23558516/ai-art-copyright-stable-diffusion-getty-images-lawsuit` (2023), online; accessed 24-July-2023

[61] Wang, K., Fu, Y., Li, K., Khisti, A., Zemel, R.S., Makhzani, A.: Variational Model Inversion Attacks. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 9706–9719 (2021)

[62] Xu, J., Xue, M., Picek, S.: Explainability-based Backdoor Attacks Against Graph Neural Networks. In: ACM Workshop on Wireless Security and Machine Learning. pp. 31–36 (2021)

[63] Yao, Y., Li, H., Zheng, H., Zhao, B.Y.: Latent Backdoor Attacks on Deep Neural Networks. In: Conference on Computer and Communications Security (CCS). pp. 2041–2055 (2019)

[64] Yeom, S., Giacomelli, I., Fredrikson, M., Jha, S.: Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In: Computer Security Foundations Symposium (CSF). pp. 268–282 (2018)

[65] Zhang, E., Wang, K., Xu, X., Wang, Z., Shi, H.: Forget-Me-Not: Learning to Forget in Text-to-Image Diffusion Models. arXiv preprint **arXiv:2303.17591** (2023)

[66] Zhang, Y., Jia, R., Pei, H., Wang, W., Li, B., Song, D.: The Secret Revealer: Generative Model-Inversion Attacks Against Deep Neural Networks. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 250–258 (2020)

[67] Zhang, Z., Jia, J., Wang, B., Gong, N.Z.: Backdoor Attacks to Graph Neural Networks. In: ACM Symposium on Access Control Models and Technologies (SACMAT). pp. 15–26 (2021)

[68] Zhang, Z., Panda, A., Song, L., Yang, Y., Mahoney, M.W., Mittal, P., Ramchandran, K., Gonzalez, J.: Neurotoxin: Durable Backdoors in Federated Learning. In: International Conference on Machine Learning (ICML). vol. 162, pp. 26429–26446 (2022)