Balancing Transparency and Risk: The Security and Privacy Risks of Open-Source Machine Learning Models

Dominik Hintersdorf [*1], Lukas Struppek [*1], and Kristian Kersting[1,2,3,4]

[1]Technical University of Darmstadt, Germany
[2]Hessian Center for AI (hessian.AI), Darmstadt, Germany
[3]National Research Center for Applied Cybersecurity ATHENE, Darmstadt, Germany
[4]German Research Center for Artificial Intelligence (DFKI), Darmstadt, Germany
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.

[*]equal contribution

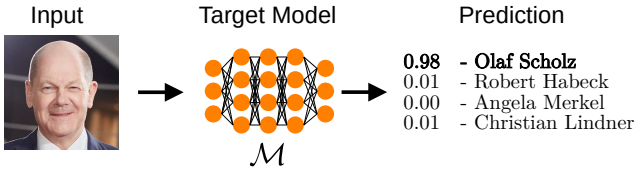

Figure 1: A basic deep neural network designed for facial recognition, capable of predicting corresponding identities, e.g., the German Chancellor Olaf Scholz. Given a specific input, the model computes a prediction vector, assigning probabilities to each distinct class. The final prediction is determined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

1 Introduction

With the increase of compute capability, big models are trained on a huge amount of data, often scraped from the public internet. Open-source models are often used as a basis for downstream tasks. As an example, the popular text-to-image model *Stable Diffusion* uses the pre-trained text encoder from CLIP [36], a pre-trained multi-modal model, to process input texts.

While some large-scale models are completely closed-source, such as OpenAI's GPT-3 [3] or Google's Bard [49], 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 [41], OpenLLaMA [16], LLaMA [50], LLaMA 2 [51], OpenCLIP [24] and Stable Diffusion [37],

and a group of companies, including GitHub, Hugging Face, Creative Commons, and others, are calling for more open-source support in the Forthcoming EU AI Act [13]. 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.

But not only these big models are made publicly available. Sites like Hugging Face[1], TensorFlow Hub[2], or PyTorch Hub[3] allow users to provide and exchange model weights trained by the community, made publicly available to be downloaded by everyone. 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 the 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. 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 the environmental conditions like weather. Security, on the other hand, describes a model's robustness 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 the models and their training data. Privacy-preserving models should not disclose any sensitive information from the training process to other users and attackers.

In this work, we will give an overview of common privacy and security threats associated with using open-source models. In Section 2 and Section 3, we will go over prominent privacy and security attacks. Then we will discuss the advantages and disadvantages of open-source practices in machine learning in Section 4, followed by a conclusion in Section 5.

2 Privacy Attacks on Open-Source Models

In this section, we will go over two most common privacy attacks, *model inversion attacks* (cf. Section 2.1 and Section 2.2) and *membership inference attacks* (cf. Section 2.3), and demonstrate how publicly releasing the model weights might harm user privacy. However, these attacks can 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.

[1] https://huggingface.co/
[2] https://tfhub.dev/
[3] https://pytorch.org/hub/

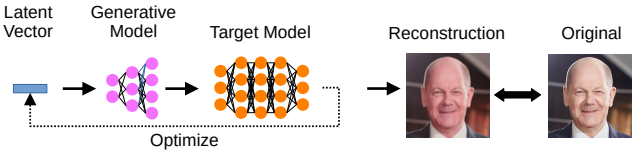

Figure 2: Model inversion attacks aim to synthesize samples that reveal sensitive information about the training data, such as revealing the identity of a person, in this case Olaf Scholz. The adversary usually employs a generative model, capable of producing synthetic images from a latent input vector.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

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 [45, 54, 11, 59, 14, 47] or generating text with private information contained in the training data [6, 35]. Fig. 2 provides a simple example of a successful inversion attack.

For model inversion attack, it is usually assumed that the attacker has full access to the model and its parameters and also some generative model to generate samples from the training data domain. Generative models, in this case usually GANs [17, 27], are able to 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 containing the sensitive features from the training data. Usually, the output value of a specific class of the model is maximized through an optimization process in which the latent vector of the generative model is altered. Even though, model inversion attacks are often applied to classification models, by altering the loss function of the optimization process these attacks can also be applied to models of other use cases. As an attacker has full access to the open-source models, model inversion attacks are a genuine threat to the privacy of the training data. 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 is the issue of data leakage through unintended memorization. The distinction lies in the adversary's intent: in a model inversion attack, the adversary actively seeks to reconstruct model inputs, whereas leakage by memorization can occur incidentally, especially when interacting with generative models. These generative models encompass vast language models like the LLaMA family [50, 51], along with image generation models like Stable Diffusion [37].

Generative language models, for instance, predict subsequent words when given a text input. 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 containing private information from its training data that should not be disclosed as part of its prediction. For instance, the model might inadvertently complete the query "My social security number" with a real social security number that was present in 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 inadvertently 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 that have 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 [5, 32]. This risk applies not only to open-source generative language models like LLaMA, but also to 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 [52, 8, 44]. Such capabilities could potentially lead to legal issues if the generated content is under copyright protection. 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 leakage and potential misuse of generated content. Proactive steps should be taken to mitigate the risks posed by both 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 [43, 21, 29, 12, 7, 57, 40], as another type of privacy attack, try to infer which data samples 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 for training a particular model. We will give a short example, to see why such a successful attack is a serious threat to privacy. Imagine that a hospital is training a machine learning model on the medical data of the 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. [22] or Li et al. [28] tries to infer if some data of a person was used for training without focussing on a particular data sample.

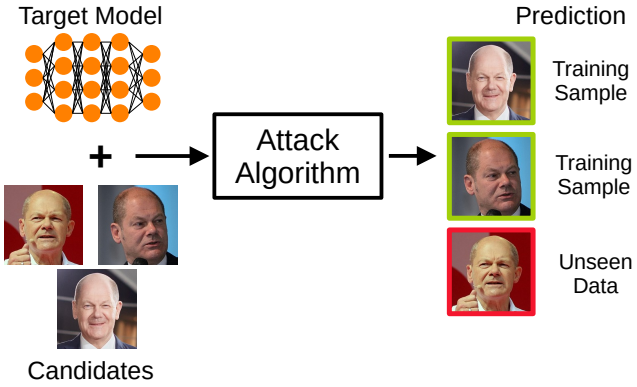

Figure 3: Membership Inference Attacks seek to determine whether a specific sample was part of a model's training data. These attacks commonly exploit that models tend to behave differently on inputs they have been trained on compared to unseen inputs.

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 the intermediate activations of 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.

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 use of data. As a result, these attacks can be used to enforce privacy and copyright laws [22]. 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 [26, 53]. 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 use 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. [22]. Combined with techniques to delete specific knowledge from the models [15, 58] 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* (cf. Section 3.1 and *adversarial examples* (cf. Section 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 [19, 38] and they are activated when the model input during inference 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

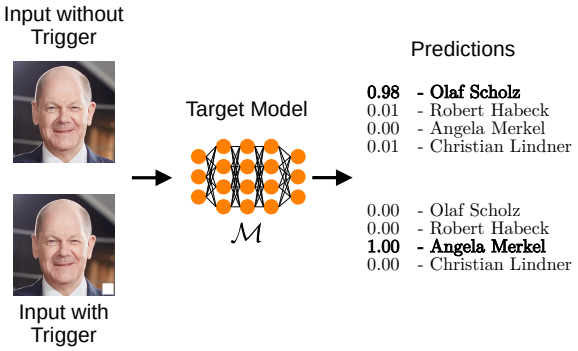

Figure 4: Backdoor attacks involve injecting a limited number of poisoned samples into a model's training data, aiming to inject a hidden model functionality, such as always predicting a specific class. During inference, this hidden behavior can be activated by inputs containing a pre-defined trigger,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

that contains both the trigger pattern and a target label from a particular class. During training, the model learns to associate the trigger pattern with the specified target class, thereby predicting the target class for each input that contains the trigger pattern. 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. A notable example are the text-to-image synthesis models, renowned for their ability to generate high-quality images based on textual descriptions provided by users. Nevertheless, Struppek *et al.* [48] have shown that small manipulations to the model are sufficient to inject backdoors 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 generated content could cause mental harm and distress.

Backdoor and poisoning attacks have become prevalent across various machine learning domains, for example, image classification, self-supervised learning [4, 39], transfer learning [56], graph neural networks [55, 60] and federated learning [61, 42]. There already exist various approaches 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 such inspections were conducted, providing an absolute guarantee that publicly available models are devoid of hidden backdoors remains challenging. The complexity and diversity of these attacks make it difficult to ensure complete protection.

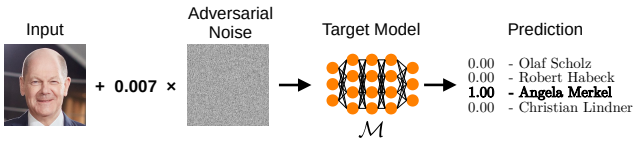

Figure 5: Adversarial examples are crafted by adding a small amount of fine-tuned noise to the input, which results in misleading predictions by the model. This adversarial noise is computed to alter the trained model's prediction in a specific manner. In many cases, the changes to the input are barely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

3.2 Adversarial Examples

In addition to poisoning attacks that manipulate the training process to introduce hidden backdoor functions into a model, another category of security attacks targets models solely during inference. Known as adversarial examples or evasion attacks [18], these are slightly modified model inputs crafted with the intention of altering the model's behavior for the given input. Consequently, these adversarial examples can be employed to bypass a model's detection and cause misclassification of samples. Fig. 5 illustrates a simple adversarial example. Among various security research subjects, adversarial examples stand out as the most extensively studied class of attacks, with numerous papers, amounting to several thousand, 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, during training, these features can be exploited to solve the specific training task effectively [25].

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 [46] or identifying deepfakes [23]. 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 [18, 33] 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 [34, 10] or only the predicted label [2, 9], 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 has clearly disadvantages, there are also upsides to make models publicly available for everyone. In the following, we provide a discussion on both perspectives. Open-source machine learning models continue to be an important resource for the AI community despite these difficulties. By implementing best practices for model usage, performing security audits, and encouraging community cooperation to proactively solve security and privacy issues, risks can be reduced. Additionally, promoting responsible vulnerability disclosure can assist in preserving the security and dependability of open-source projects.

⛔ **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 inadvertently memorize or encode this information into its 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 aimed at manipulating or compromising 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. It is also an open question to which extent generative models can commit copyright infringement. Since parts of the training data may underlay copyright regulations, the generated data might also incorporate parts of it and, therefore, 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 of any kind. 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 the publishing companies such as OpenAI, Meta, or Google. In the case of open-source models, this process of finding and disclosing vulnerabilities can be done by the community 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 and allow 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.

5 Conclusion

In conclusion, we have highlighted and discussed the vulnerabilities of open-source models concerning security and privacy attacks, which are expected to pose a greater risk compared to closed-source models. The 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 not only impact the published model itself, but also 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.

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 Ludewig [31], Lucan [30] and Heinrich-Böll-Stiftung [20].

References

[1] Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *Symposium on Security and Privacy (S&P)*, pages 141–159, 2021.

[2] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations (ICLR)*, 2018.

[3] 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. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[4] Nicholas Carlini and Andreas Terzis. Poisoning and backdooring contrastive learning. In *International Conference on Learning Representations (ICLR)*, 2022.

[5] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security Symposium*, pages 267–284, 2019.

[6] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *USENIX Security Symposium*, pages 2633–2650, 2021.

[7] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *Symposium on Security and Privacy (S&P)*, pages 1897–1914, 2022.

[8] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. *arXiv preprint*, arXiv:2301.13188, 2023.

[9] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *Symposium on Security and Privacy (S&P)*, pages 1277–1294, 2020.

[10] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 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)*, pages 15–26, 2017.

[11] Si Chen, Mostafa Kahla, Ruoxi Jia, and Guo-Jun Qi. Knowledge-enriched distributional model inversion attacks. In *International Conference on Computer Vision (ICCV)*, pages 16158–16167, 2021.

[12] Christopher A. Choquette-Choo, Florian Tramèr, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International Conference on Machine Learning (ICML)*, pages 1964–1974, 2021.

[13] Emilia David. 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.

[14] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, 2015.

[15] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. *arXiv preprint*, arXiv:2303.07345, 2023.

[16] Xinyang Geng and Hao Liu. Openllama: An open reproduction of llama, 2023. URL `https://github.com/openlm-research/open_llama`.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[18] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[19] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint*, arXiv:1708.06733, 2017.

[20] Heinrich-Böll-Stiftung. https://commons.wikimedia.org/wiki/File:Olaf_Scholz_(14271189360).jpg, Licensed as CC BY-SA 2.0, accessed: 24.07.2023.

[21] Dominik Hintersdorf, Lukas Struppek, and Kristian Kersting. To trust or not to trust prediction scores for membership inference attacks. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3043–3049, 2022.

[22] Dominik Hintersdorf, Lukas Struppek, and Kristian Kersting. Clipping privacy: Identity inference attacks on multi-modal machine learning models. *arXiv preprint*, arXiv:2209.07341, 2022.

[23] Shehzeen Hussain, Paarth Neekhara, Malhar Jere, Farinaz Koushanfar, and Julian McAuley. Adversarial deepfakes: Evaluating vulnerability of deepfake detectors to adversarial examples. In *Winter Conference on Applications of Computer Vision (WACV)*, pages 3348–3357, 2021.

[24] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021.

[25] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 125–136, 2019.

[26] Getty Images. Getty images statement. https://newsroom.gettyimages.com/en/getty-images/getty-images-statement, 2023. Online; accessed 24-July-2023.

[27] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410, 2019.

[28] Guoyao Li, Shahbaz Rezaei, and Xin Liu. User-level membership inference attack against metric embedding learning. *arXiv preprint*, arXiv:2203.02077, 2022.

[29] Zheng Li and Yang Zhang. Membership leakage in label-only exposures. In *Conference on Computer and Communications Security (CCS)*, pages 880–895, 2021.

[30] Michael Lucan. https://commons.wikimedia.org/wiki/File:2021-08-21_Olaf_Scholz_0433.JPG, Licensed as CC BY-SA 3.0, accessed: 24.07.2023.

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

[32] Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Analyzing leakage of personally identifiable information in language models. In *Symposium on Security and Privacy (S&P)*, pages 346–363, 2023.

[33] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[34] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Asia Conference on Computer and Communications Security (ASIA CCS)*, page 506–519, 2017.

[35] Rahil Parikh, Christophe Dupuy, and Rahul Gupta. Canary extraction in natural language understanding models. In *Annual Meeting of the Association for Computational Linguistics (ACL) - Short Paper*, pages 552–560, 2022.

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021.

[37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2022.

[38] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *Conference on Artificial Intelligence (AAAI)*, pages 11957–11965, 2020.

[39] Aniruddha Saha, Ajinkya Tejankar, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash. Backdoor attacks on self-supervised learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13337–13346, 2022.

[40] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 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.

[41] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. BLOOM: A 176b-parameter open-access multilingual language model. *arXiv preprint*, arXiv:2211.05100, 2022.

[42] Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In *Symposium on Security and Privacy (S&P)*, pages 1354–1371, 2022.

[43] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Symposium on Security and Privacy (S&P)*, pages 3–18, 2017.

[44] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. *arXiv preprint*, arXiv:2305.20086, 2023.

[45] Lukas Struppek, Dominik Hintersdorf, Antonio De Almeida Correia, Antonia Adler, and Kristian Kersting. Plug & play attacks: Towards robust and flexible model inversion attacks. In *International Conference on Machine Learning (ICML)*, pages 20522–20545, 2022.

[46] Lukas Struppek, Dominik Hintersdorf, Daniel Neider, and Kristian Kersting. Learning to break deep perceptual hashing: The use case neuralhash. In *Conference on Fairness, Accountability, and Transparency (FAccT)*, page 58–69, 2022.

[47] Lukas Struppek, Dominik Hintersdorf, Felix Friedrich, Manuel Brack, Patrick Schramowski, and Kristian Kersting. Image classifiers leak sensitive attributes about their classes. *arXiv preprint*, arXiv:2303.09289, 2023.

[48] Lukas Struppek, Dominik Hintersdorf, and Kristian Kersting. Rickrolling the artist: Injecting backdoors into text-guided image generation models. In *International Conference on Computer Vision (ICCV)*, 2023.

[49] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Agüera y Arcas, Claire Cui, Marian Croak, Ed H. Chi, and Quoc Le. Lamda: Language models for dialog applications. *arXiv preprint*, arXiv:2201.08239, 2022.

[50] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv preprint*, arXiv:2302.13971, 2023.

[51] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti 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, Madian 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, Rashi 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. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint*, arXiv:2307.09288, 2023.

[52] Gerrit J. J. van den Burg and Chris Williams. On memorization in probabilistic deep generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 27916–27928, 2021.

[53] James Vincent. 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.

[54] Kuan-Chieh Wang, Yan Fu, Ke Li, Ashish Khisti, Richard S. Zemel, and Alireza Makhzani. Variational model inversion attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9706–9719, 2021.

[55] Jing Xu, Minhui Xue, and Stjepan Picek. Explainability-based backdoor attacks against graph neural networks. In *ACM Workshop on Wireless Security and Machine Learning*, pages 31–36, 2021.

[56] Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y. Zhao. Latent backdoor attacks on deep neural networks. In *Conference on Computer and Communications Security (CCS)*, pages 2041–2055, 2019.

[57] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *Computer Security Foundations Symposium (CSF)*, pages 268–282, 2018.

[58] Eric Zhang, Kai Wang, Xingqian Xu, Zhangyang Wang, and Humphrey Shi. Forget-me-not: Learning to forget in text-to-image diffusion models. *arXiv preprint*, arXiv:2303.17591, 2023.

[59] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 250–258, 2020.

[60] Zaixi Zhang, Jinyuan Jia, Binghui Wang, and Neil Zhenqiang Gong. Backdoor attacks to graph neural networks. In *ACM Symposium on Access Control Models and Technologies (SACMAT)*, pages 15–26, 2021.

[61] Zhengming Zhang, Ashwinee Panda, Linyue Song, Yaoqing Yang, Michael W. Mahoney, Prateek Mittal, Kannan Ramchandran, and Joseph Gonzalez. Neurotoxin: Durable backdoors in federated learning. In *International Conference on Machine Learning (ICML)*, volume 162, pages 26429–26446, 2022.