# Privacy Not Guaranteed

The exponential growth of AI has often been attributed to increases in the availability of data, computing power, and algorithmic innovation. Unsurprisingly, privacy concerns in AI have also been top of mind lately, exasperated by these same factors. This article summarizes the privacy and security threats in the existing landscape with a focus on Machine Learning (ML) specific threats and offer some mitigating techniques specific to the ML space as well as a path forward to a more generalized solution.

## Introduction

Privacy and security in machine learning draw similarities to web application security. When building an application, many developers often neglect the protection of sensitive data and opt for a workable application as the top priority. Similarly, in Machine Learning, model accuracy and predictive power is often the top priority with data protection as an afterthought, if at all.

Data privacy risks can be categorized as direct or indirect information exposure. Direct exposure refers to data leak or breach either unintentionally or by an adversary. Indirect exposure refers to having some aspect about the private data, either attribute, membership, or model parameters can be inferred by an attacker at inference time. With the rise of algorithmic innovation, this is especially troubling as ML algorithms are increasingly being used for these attacks. These threats can be further categorized as being perpetrated by an internal or external entity, having various levels of access to the infrastructure or the ML model itself.

## Direct Exposure

Direct exposure is easy to understand as it’s not limited to the ML domain. These include direct compromise of an infrastructure, unsecure communication channels, or an insider with elevated access. Mitigation techniques has been thoroughly studied and include having the proper processes and procedures in place. Most commonly these days, SOC 2 compliance is required for service providers to ensure customers that they follow proper processes and procedures for handling sensitive data.

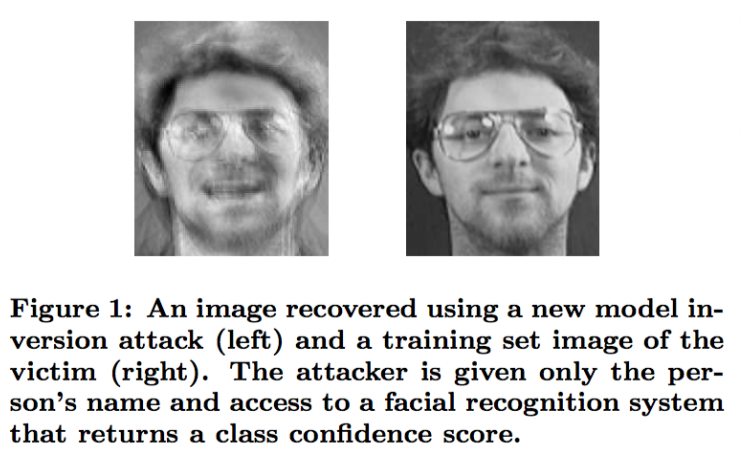


In the ML domain, there are specific techniques aimed at separating the need for having all the training data at one place as was traditionally required – using federated learning or split learning techniques. For securing the process/execution at inference, techniques such as homomorphic encryption, secure multi-party computation, and trusted execution environments can be used.

## Indirect Exposure

Indirect information exposure, as its applied to the ML domain, refers to the ability for an attacker to infer private information about the training data. These include Membership Inference, detecting whether a specific instance of data was used in the training of the model, Model Inversion, deriving sensitive attributes about a specific instance of data, and Model Stealing, inferring the parameter and hyperparameters of the model itself.

One of the most cited examples shows an image recovered using a model inversion technique.



Mitigation techniques for indirect information exposure centers around aggregating and anonymizing the training data with the aim to preserve the privacy of the data contributors while still being useful for training a model. It can range from as simple as removing personally identifiable attributes from the dataset, to context aware techniques like Information-Theoretic approach. One example uses a generative adversarial network (GANS) to generate privatized datasets.

## Conclusion

As we can see, the risk to privacy in ML is abundant and at this point feels a lot like the wild wild west of web application security in the 1990s. As the growth in ML applications explode exponentially, so too are the risks to personal privacies. Many ML service providers already go through a SOC 2 certification process to certify their data handling practices, which help mitigates direct exposures described above. Perhaps a similar certification process can be created to categorize and formalize the indirect exposure risk mitigation techniques.

