Property Inference Attacks on Convolutional Neural Networks: Influence and Implications of Target Model's Complexity

Mathias Parisot[1][a], Balázs Pejó[2][b] and Dayana Spagnuelo[3][c]
[1]*University of Amsterdam, Amsterdam, The Netherlands*
[2]*CrySyS Lab, Dept. of Networked Systems and Services, Budapest Univ. of Technology and Economics, Budapest, Hungary*
[3]*Vrije Universiteit Amsterdam, Amsterdam, The Netherlands*

Keywords: Property Inference Attacks, Convolutional Neural Networks, Model Complexity.

Abstract: Machine learning models' goal is to make correct predictions for specific tasks by learning important properties and patterns from data. By doing so, there is a chance that the model learns properties that are unrelated to its primary task. Property Inference Attacks exploit this and aim to infer from a given model (*i.e.*, the target model) properties about the training dataset seemingly unrelated to the model's primary goal. If the training data is sensitive, such an attack could lead to privacy leakage.
In this paper, we investigate the influence of the target model's complexity on the accuracy of this type of attack, focusing on convolutional neural network classifiers. We perform attacks on models that are trained on facial images to predict whether someone's mouth is open. Our attacks' goal is to infer whether the training dataset is balanced gender-wise. Our findings reveal that the risk of a privacy breach is present independently of the target model's complexity: for all studied architectures, the attack's accuracy is clearly over the baseline.

1 INTRODUCTION

Machine Learning (ML) applications received much attention over the last decade, mostly due to their vast application range. It is generally accepted that data plays a vital role in ML models' performance, and that more elaborate models can solve difficult tasks more accurately as they can learn complex patterns from data. Notwithstanding, besides improving the performance, such ML models introduce privacy issues for the underlying datasets (He et al. (2019)).

Training ML models typically requires significant amounts of data, potentially private and sensitive data, and the risk of privacy leakage is not negligible. Suppose a classification model that, once trained, can recognize the appropriate class of a data instance by learning mapping patterns between the training dataset and its original set of classes. This mapping is contained within the model parameters, for instance, in a neural network it is encoded in each neuron's weights. An attacker knowing the trained model parameters could also gain some information about the data it was trained on. This is the rationale of Property Inference Attacks (PIAs) (Ganju et al. (2018); Melis et al. (2019)), which aim at uncovering properties of the dataset in which a given model was trained by analyzing the parameters of the model only.

Given the current popularity and the increase in performance of ML applications, it is reasonable to question to which extent the training dataset is vulnerable to privacy attacks. In particular if improving a model's performance is realized by increasing the complexity of the model. As more complex models have more parameters and can retain more information about the training dataset, intuitively one may think that due to this information retention, more complex models could be more sensitive to PIAs as well. In this paper, we study this phenomenon, which — if the model is trained on a dataset containing personal data — could lead to potential privacy leakage.

Contribution. In this work, we test the influence of a model's complexity on its vulnerability to PIAs. Our setting is processing facial images; thus, we focus on Convolutional Neural Networks (CNNs), the most common model of choice for computer vision tasks. Given that we measure complexity in terms of

[a] https://orcid.org/0000-0003-1521-918x
[b] https://orcid.org/0000-0002-1825-9251
[c] https://orcid.org/0000-0001-6882-6480

Parisot, M., Pejó, B. and Spagnuelo, D.
Property Inference Attacks on Convolutional Neural Networks: Influence and Implications of Target Model's Complexity.
DOI: 10.5220/0010555607150721
In *Proceedings of the 18th International Conference on Security and Cryptography (SECRYPT 2021)*, pages 715-721
ISBN: 978-989-758-524-1

the number of layers and weights of the model's architecture, *we hypothesize that more complex models are also more sensitive to PIAs.* To uncover whether and how the risk of privacy leaks is influenced by the architecture's complexity of the model, we experiment on nine different CNN architectures. For each, we conduct 30 attacks based on 1500 shadow models.

Organization. In Section 2 we present works related to ours, then we describe the methodology we follow to compose our attack (Section 3). In Section 4 we describe the implementation of our experimental setup, present the results of our attack and discuss their meaning. Finally, in Section 5, we present final remarks and the future directions for our work.

2 RELATED WORKS

Several security threats are studied regarding machine learning focusing information security. For instance (Shumailov et al. (2020)) present an ML attack targeting the model's availability. According to (He et al. (2019)) the main attack categories for integrity are adversarial and poisoning attacks, while for confidentiality, these are model extraction and inversion.

Adversarial attacks (Szegedy et al. (2013)) aim to take advantage of the weaknesses of the target model's classification boundary to craft data instances that are wrongly classified. Poisoning attacks (Mei and Zhu (2015)) is similar to adversarial attacks as their goal is to influence the prediction of the target model. They do that by polluting the training set with malicious samples. While those are realistic threats to ML models' integrity, they do not pose immediate risk to data privacy. Instead, our work focuses on confidentiality attacks.

Concerning confidentiality (and privacy), the recent survey (Rigaki and Garcia (2020)) gives a comprehensive overview of the subject. Here we only briefly comment on the most relevant works for our purposes, we refer the reader to that work for more details. Model extraction (Papernot et al. (2017); Wang and Gong (2018)) attacks aim at inferring the behavior of the target model to create a substitute model. Model inversion attacks (Fredrikson et al. (2015); Mehnaz et al. (2020)) aim at inferring information about the training data, for example, by reconstructing a representative of a particular class of the training set. In such a case (Mehnaz et al. (2020)) showed that some subgroups are more vulnerable to this type of attack than others.

Depending on the goal of the attacker, we can classify three more attacks under the category of model inversion: membership inference, reconstruction attack, and property inference attack (PIA). Membership inference attacks (Truex et al. (2018); Murakonda et al. (2019)) aim at determining whether a particular data instance was used for training. This raises privacy concerns when the instance directly maps to an identifiable individual, for instance, a medical records dataset. In contrast, as its name suggests, reconstruction attacks (Zhu and Han (2020); Li et al. (2019)) take this one step further and aim at recovering both the training inputs and the corresponding labels. The last category is the one of PIAs, the subject of this study. It aims at inferring hidden properties of the dataset that are independent of any class characteristics, and are therefore not necessarily related to the main classification task. Such properties can be general statistics about the dataset or can reflect biases in the training data.

2.1 Property Inference Attacks

According to (He et al. (2019)), only four papers were published on model inversion attacks, and researchers have not yet entirely determined the vulnerability of neural network architectures to privacy attacks such as PIA. The works we discuss next perform PIAs in a federated learning setting, which allows multiple participants (also called clients by some works) to train a standard model without the need to share data. After each round of training, only the weights and the gradients are exchanged, while data remains protected on the participants' premises.

The work presented in (Melis et al. (2019)) manages to infer properties that hold for a subset of the training data and that are independent of the property the target model aims to predict. Since the attack is performed during the training phase, it requires the model updates that are exchanged between participants. In contrast, the attack we focus on does not require the gradient values after each training round. We also target properties that are true for the whole dataset and not only for a subset.

In (Wang et al. (2019a)) three kinds of PIAs are proposed: class sniffing, quantity inference, and whole determination. Class sniffing detects whether a training label is present within a training round. Quantity inference determines how many clients have a given training label in their dataset. The whole determination infers the global proportion of a specific label. Those attacks extract properties related to classification labels, and therefore to the main classification task. We focus on properties that are, in theory, unrelated to the task of the target model.

Attempts to explore user-level privacy leakage

in a federated learning scenario is also subject of recent works (Wang et al. (2019b); Pejó (2020)). They define client-dependent properties to precisely characterize the clients and distinguish them from each other, *e.g.*, (Pejó (2020)) assume an honest-but-curious setting and recover the participants' quality information via a differential attack without extra computational needs or access to the individual gradient updates.

2.2 Attacks Concerning Model Complexity

A model inversion attack is presented in (Zhang et al. (2020)). The authors study and theoretically prove the attack's relation with the model predictive power: more complex models, which should have greater predictive power, should also be more sensitive to model inversion attacks. However, the result of (Zhang et al. (2020)) was not proven for PIAs, our focus.

Model inversion attacks are also studied in recent works (Geiping et al. (2020)). This work analyzes the effects of the target model's architecture on the difficulty of reconstructing input images. The authors investigate attacks on networks with various widths and depths and found that the width has the most significant influence on the reconstruction's quality. Contrary to ours, their study does not consider PIAs and is restricted to federated learning as their attack utilizes the gradient values.

In (Ateniese et al. (2015)) the first PIA attack using meta-classifiers is described, this is the methodology of the attack we use in our paper. However, their research does not focus on the privacy leakage caused by such an attack (our goal), but instead on the impact of the training set properties on the model performance. They also attack models implemented via Support Vector Machines and Hidden Markov Models using a binary tree meta-classifier, while we experiment with deep neural network models.

Finally, an extension of the previous work is presented by (Ganju et al. (2018)). The authors focus on neural networks and notice that a limitation of PIA performance is due to a property of fully connected networks called invariance. They propose two successful strategies to reduce this and used a pre-trained network to generate an embedding, which they feed as input to their target neural network. However, they do not study the influence of the type of layers and the model's architecture on the attack performance, which is the goal of our work.

Table 1: Notations used in the rest of the paper.

Notation	Meaning
M_t	Target model (CNN)
D_t	Training dataset of the target model
P	Property to be inferred
$M_{s_1}, \ldots, M_{s_k}$	Shadow models, mimicking M_t
$W_{s_1}, \ldots, W_{s_k}$	Weights of the shadow models
$D_{s_1}, \ldots, D_{s_k}$	Training dataset of the shadow models
D	The dataset from which D_{s_i} is created
M_a	Attack model to predict P about D_t
D_a	Training dataset of the attack model composed by $W_{s_1}, \ldots, W_{s_k}$

3 METHODOLOGY

Threat Model. Our target model is a CNN classifier. We assume a training dataset for the classifier, which contains sensitive data, *e.g.*, data revealing racial or ethnic origin, religious beliefs, or biometric data. We assume an attacker whose goal is to infer general information about the training dataset, such as the proportion of the training data having a property P. This property is unrelated to the main classification task of the model. We assume the attacker can fabricate datasets similar to the original training dataset, *e.g.*, (s)he knows from which distribution the original data was created. Moreover, the attacker can manipulate these datasets so that they either contain or not property P. We also assume the attacker has access to the target model and can train a large number of neural networks.

We focus on the white-box setting, where the attacker has access to the target model's full architecture and parameter values. Such information could be obtained in many ways: for instance, it could be shared explicitly as in Federated Learning. Alternatively, when this information is not shared for neural networks, it could still be obtained by creating a substitute model with a similar decision boundary as the target model using a model extraction attack (Papernot et al. (2016)).

To improve readability of the next paragraphs, we summarize the paper's notations in Table 1.

Attack. We focus on PIAs whose goal is to extract information about the target model's training dataset. This information is named property P, which can be true or false. For instance, if the used dataset contains images of faces, P can be defined as *more than 20% of the images within the dataset depict non-white people*. In this sense, PIA is transformed into a classification problem: to determine whether a given model was trained on a dataset with property P. This at-

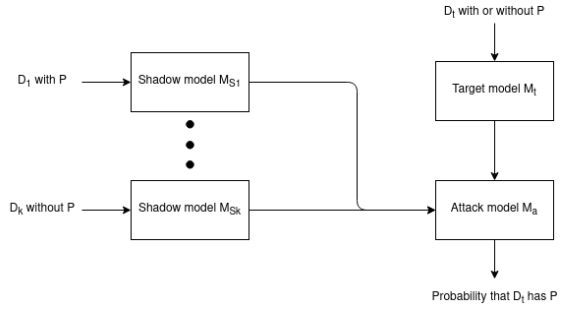


Figure 1: Property Inference Attack using a meta-classifier M_a and datasets of shadow models $\{M_{s_1}, \ldots, M_{s_k}\}$.

tack model can be understood as a meta-classifier as the dataset on which it is trained composed of shadow models, which are themselves classifiers.

The attack consists of training k shadow models $(M_{s_1}, \ldots, M_{s_k})$ on k datasets $(D_{s_1}, \ldots, D_{s_k})$ specifically crafted to contain or not the target property P. The training set (D_a) for the attack model (M_a) is composed by the weights $(W_{S_1}, \ldots, W_{S_k})$ of shadow models $(M_{s_1}, \ldots, M_{s_k})$ fabricated with the same architecture as the target model M_t.

The general overview of our attack is described in Figure 1: we train an attack model that takes as input the weights of the target model and outputs the probability the dataset used for training the target model has property P or not. This is based on the baseline PIA presented in (Ganju et al. (2018)), which serves our purpose as we do not focus on the PIA itself but rather on the PIA's behavior performed on models with different complexities.

4 EXPERIMENTAL SETUP

The experiments were performed on a laptop with an Intel i7-8750H (2.20GHz), 8GB RAM, an Nvidia Quadro P600 GPU, and operating system Ubuntu 20.04. The shadow models' training and the attack models were both done using Pytorch and are available on a public repository[1].

Datasets. To train the shadow models for our experiments, we have selected CelebFaces Attributes (CelebA) (Liu et al. (2015)) dataset, which contains personal and sensitive information. This is a face attributes dataset containing more than 200.000 face-centered images of 64 by 64 pixels of more than 10.000 celebrities. The images are labeled using 40 physical attributes such as hair color, smiling, and wearing a hat. Our shadow models and the target model are trained to detect whether the person appears with their mouth open in a given photo.

Although this might seem like an irrelevant classification task, at the time of execution of this work, the world is undergoing a pandemic, and mouth covering masks is one of humanity's currently available weapons against the SARS-CoV-2 virus (Eikenberry et al. (2020)). This task can be related to automated mask detection[2], especially since, in many places, their use is compulsory.

Shadow Models. The shadow models $\{M_{s_1}, \ldots, M_{s_k}\}$ are trained to mimic the target model M_t, *i.e.*, to differentiate between images of persons with and without their mouth open. However, the attacker's goal is to infer whether the training set of a given model is composed of a seriously unbalanced number of images of males. We would like to detect disproportions above 40% which is double of the rarely occurring 20% unbalance in the real world (Hesketh and Min (2012)): *P is true when the model's training set is composed of 70% or more images containing males*. It is important to note that P is not related to the model's classification task and that the target model does not use, at any time during training, the gender attribute.

The shadow models $\{M_{s_1}, \ldots, M_{s_k}\}$ have the same architecture as the targeted model and are trained to a reasonable level of accuracy to mimic the target model's behavior: the shadow models are at least 85% accurate (on the mouth open classification task) when the underlying distribution of the dataset is 51.7%.

Many shadow models are trained, as the attack model's inputs are the weights of the shadow models. For a specific target model architecture, we train 1800 shadow models. Since the computational cost of training many shadow models is significant, we decide not to use all images of *CelebA*. Rather, for each shadow dataset $\{D_{s_1}, \ldots, D_{s_k}\}$, we use only 2000 randomly selected images. For half of the shadow models (i.e., 900 times) these 2000 images were selected to have property P, while the remaining does not. The exact proportion of males for each dataset was randomly taken from a uniform distribution either above or below 70%, respectively. It is important to note that while each shadow model is trained using only 2000 images, they perform with at least 85% accuracy on the whole test set of CelebA; therefore, they do not overfit to their smaller training set.

We experimented on target model's (and consequently the shadow models') architectures composed of up to 9 layers, each of three kinds: convolution

[1] https://github.com/MatPrst/PIA-on-CNN

[2] http://tinyurl.com/2a8ewzvl

layers, pooling layers, and fully connected layers. We trained 9 architectures ($A_1, \ldots, A_9$) which are presented in Table 2, while the description of each layer is presented in Table 3. The models take as input 64×64 RGB face images and output each picture's probability of representing a person with mouth open. Every architecture comprises 1-3 convolution layers, each followed by a max-pooling layer with a ReLU activation and 1-3 fully connected layers with a ReLU activation. The shadow models are trained for 50 epochs using the Mean Squared Error loss and the Adam optimizer with a learning rate of 0.001 and without decay or regularization.

Attack Model and Evaluation. The attack model classifies shadow models on whether they were trained on a dataset with the property P. The dataset used for the attack is composed of the 1800 shadow models and is split into training (1500 models), validation (100 models), and test sets (200 models). The training algorithm is presented in Algorithm 1. The attack model is a simple multi-layer perceptron tuned using the validation set and evaluated on the test set.

Algorithm 1: Attack model training.

procedure TRAIN_ATTACK(D, k)
  let: D be the dataset of images, k be the number of shadow models to train, D_{s_i} be a subset of D, P_{s_i} be a boolean value determining whether P is True on D_{s_i}, D_a be the dataset used to train the attack model
  $D_a \leftarrow \{\}$
  for $i \leftarrow 1, k$ **do**
    $D_{s_i} \leftarrow$ sample subset of D
    $P_{s_i} \leftarrow$ evaluate P on D_{s_i}
    $M_{s_i} \leftarrow train(D_{s_i})$
    $W_{s_i} \leftarrow getWeights(M_{s_i})$
    $D_a \leftarrow D_a \cup \{(W_{s_i}, P_{s_i})\}$
  end for
  $M_a \leftarrow train(D_a)$
  return M_a
end procedure

We tuned the attack model by performing a grid search over the following hyper-parameters: learning rate, loss function, batch size, optimizer, and the activation function of the first layer of the attack model. The hyper-parameters values we used are presented in Table 4. We trained six attack models for the 9 model architectures and each combinations of parameters during ten epochs. We selected the best parameters by combining the largest median accuracy over the 9 model architectures.

The attack model's inputs are the flattened weights of the model it is trying to classify as having or not the property P. Therefore, a target model architecture with a larger number of parameters induces a more comprehensive input layer for the attack model. The attack model comprises two fully connected layers (the first with 10 neurons followed by a ReLU activation and the second, the output layer, with one neuron) which were trained 30-fold for 20 epochs for each shadow model architecture. The average performances (Mean Squared Error) is presented when the Adam optimizer was used with learning rate 0.005 and without any regularization.

5 DISCUSSIONS AND CONCLUSION

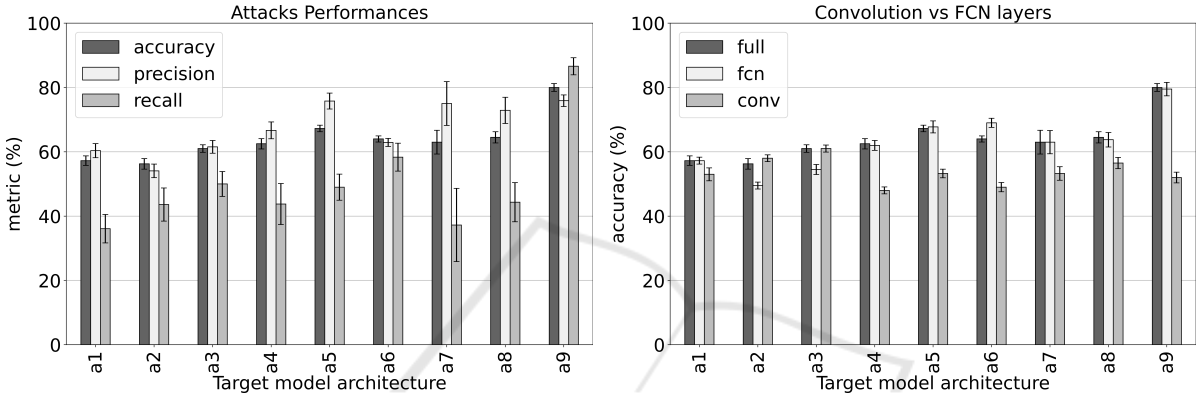
Figure 2 summarizes the performance of our attack. In detail, Figure 2a presents the accuracy of the attacks on each target model architecture, which varies between $56\% - 80\%$ depending on the architecture. These results confirm the findings of (Ganju et al. (2018)) that the target models do learn information unrelated to the task they were trained to learn. We create as many shadow models presenting the property P as ones not presenting it in our setup. Therefore, the expected baseline is 50% accuracy. Our attack's accuracy are always above this baseline independently of the underlying architecture.

We performed PIAs on distinctive neural network architectures with different amounts and types of layers. As convolution layers and fully connected ones play different roles in a CNN, we also tested whether the type of used layers impacts the attack's accuracy. Thus we conducted three additional PIAs on each of the architectures presented in Table 2: 1) using all the weights of the shadow model; 2) using only the weights of the convolution layers, and 3) using only the weights of the fully connected layers. Figure 2b presents the accuracy of the three attacks. For most target model architectures, the PIA using only the fully connected weights performs as well, and sometimes better, as the PIA using the weights from both types of layers. Consequently, the information leaked by a CNN seems to be contained in the fully connected part of the network. Moreover, the leakage does seems to be related to the attacked model's complexity (which we define by the number of parameters) as shown in Figure 3.

Conclusion. This work presents an attack that tries to determine if a given dataset (of faces) used to train a CNN model (to determine if a mouth is open in a picture) has some property P, in our case, whether the dataset was unbalanced gender-wise. We conducted several experiments to uncover the relationship be-

Table 2: Layer-level description of target and shadow models' architectures. Parameters in each layer are shown in Table 3.

	Conv. 1	Max-pool	Conv. 2	Max-pool	Conv. 3	Max-pool	FC 1	FC 2	FC 3
A_1	✓	✓	✓	✓	✓	✓	✓	✓	✓
A_2	✓	✓	✓	✓	✓	✓	✓		✓
A_3	✓	✓	✓	✓	✓	✓			✓
A_4	✓	✓	✓	✓			✓	✓	✓
A_5	✓	✓	✓	✓			✓		✓
A_6	✓	✓	✓	✓					✓
A_7	✓	✓					✓	✓	✓
A_8	✓	✓					✓		✓
A_9	✓	✓							✓



(a) Accuracy, precision, and recall using weights from entire target model.

(b) Accuracy using weights from: entire target model (full), only fully connected layers (fcn), and only convolution layers (conv).

Figure 2: Attack's performance on each architecture (bars = median of 30 attacks; error bars = ± standard deviation).

Table 3: Layers in target and shadow models' architecture.

Layer	Description
Convolution 1	6 filters 5x5
Max-pool	2x2, ReLU
Convolution 2	16 filters 5x5
Max-pool	2x2, ReLU
Convolution 3	32 filters 5x5
Max-pool	2x2, ReLU
Fully-Connected 1	120 neurons, ReLU
Fully-Connected 2	84 neurons, ReLU
Fully-Connected 3	1 neuron

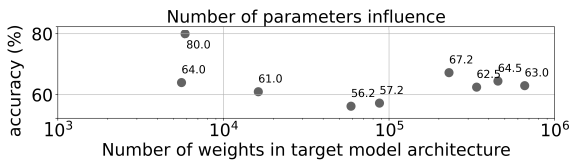


Figure 3: Influence of the complexity of the target model (express as the number of parameters) on the accuracy of the attacks on each architecture (dot = median of 30 attacks).

Table 4: Hyper-parameter tuning for attack model with optimal ones marked in bold.

Parameter	Values
Learning rate	**0.005**; 0.001; 0.0005
Loss function	**MSE**; L1-loss
Batch size	16; **32**; 64
Optimizer	SGD; **Adam**
Input layer act. func.	sigmoid; **ReLU**; tanh

tween target model complexity and privacy leakage (aka PIA's accuracy), and we find no significant correlation between the two. Although our findings do not support our initial hypothesis, they reveal a surplus of personal data used in the training stage of CNN models. Intuitive solutions for the surplus of data, such as cropping images to contain only the area relevant for the classification task, could prove insufficient. As empirically demonstrated[3], gender traits can be encoded in many areas of a facial image. Other alternatives (*e.g.*, feature anonymization (Kim and Yang (2020)) or gender obfuscation through morphing (Wang (2020))) could be tested for their impact on PIAs. We leave this task for future work. Due

[3]https://www.pewresearch.org/interactives/how-does-a-computer-see-gender/

to the nature of PIAs, our work has an explicit limitation: it is tailored to one specific property of the dataset. Our attack can be adapted to other properties P as well, as long as the attacker can fabricate datasets containing or not the given property.

REFERENCES

Ateniese, G., Mancini, L. V., Spognardi, A., Villani, A., Vitali, D., and Felici, G. (2015). Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks*.

Eikenberry, S. E., Mancuso, M., Iboi, E., Phan, T., Eikenberry, K., Kuang, Y., Kostelich, E., and Gumel, A. B. (2020). To mask or not to mask: Modeling the potential for face mask use by the general public to curtail the covid-19 pandemic. *Infectious Disease Modelling*.

Fredrikson, M., Jha, S., and Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*.

Ganju, K., Wang, Q., Yang, W., Gunter, C. A., and Borisov, N. (2018). Property inference attacks on fully connected neural networks using permutation invariant representations.

Geiping, J., Bauermeister, H., Dröge, H., and Moeller, M. (2020). Inverting gradients–how easy is it to break privacy in federated learning? *arXiv preprint arXiv:2003.14053*.

He, Y., Meng, G., Chen, K., Hu, X., and He, J. (2019). Towards privacy and security of deep learning systems: a survey. *arXiv preprint arXiv:1911.12562*.

Hesketh, T. and Min, J. M. (2012). The effects of artificial gender imbalance: Science & society series on sex and science. *EMBO reports*.

Kim, T. and Yang, J. (2020). Selective feature anonymization for privacy-preserving image data publishing. *Electronics*.

Li, Z., Huang, Z., Chen, C., and Hong, C. (2019). Quantification of the leakage in federated learning. *arXiv preprint arXiv:1910.05467*.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Mehnaz, S., Li, N., and Bertino, E. (2020). Black-box model inversion attribute inference attacks on classification models. *arXiv preprint arXiv:2012.03404*.

Mei, S. and Zhu, X. (2015). Using machine teaching to identify optimal training-set attacks on machine learners. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Melis, L., Song, C., De Cristofaro, E., and Shmatikov, V. (2019). Exploiting unintended feature leakage in collaborative learning.

Murakonda, S. K., Shokri, R., and Theodorakopoulos, G. (2019). Ultimate power of inference attacks: Privacy risks of learning high-dimensional graphical models. *arXiv preprint arXiv:1905.12774*.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. (2016). Practical black-box attacks against deep learning systems using adversarial examples. *arXiv preprint arXiv:1602.02697*.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. (2017). Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*.

Pejó, B. (2020). The good, the bad, and the ugly: Quality inference in federated learning. *arXiv preprint arXiv:2007.06236*.

Rigaki, M. and Garcia, S. (2020). A survey of privacy attacks in machine learning. *arXiv preprint arXiv:2007.07646*.

Shumailov, I., Zhao, Y., Bates, D., Papernot, N., Mullins, R., and Anderson, R. (2020). Sponge examples: Energy-latency attacks on neural networks. *arXiv preprint arXiv:2006.03463*.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Truex, S., Liu, L., Gursoy, M. E., Yu, L., and Wei, W. (2018). Towards demystifying membership inference attacks. *arXiv preprint arXiv:1807.09173*.

Wang, B. and Gong, N. Z. (2018). Stealing hyperparameters in machine learning. In *2018 IEEE Symposium on Security and Privacy (SP)*.

Wang, L., Xu, S., Wang, X., and Zhu, Q. (2019a). Eavesdrop the composition proportion of training labels in federated learning. *arXiv preprint arXiv:1910.06044*.

Wang, S. (2020). Gender obfuscation through face morphing. Master's thesis, University of Twente.

Wang, Z., Song, M., Zhang, Z., Song, Y., Wang, Q., and Qi, H. (2019b). Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*.

Zhang, Y., Jia, R., Pei, H., Wang, W., Li, B., and Song, D. (2020). The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Zhu, L. and Han, S. (2020). Deep leakage from gradients. In *Federated Learning*.