Attacking and Protecting Data Privacy in Edge-Cloud Collaborative Inference Systems

Kandagatla Raghuvaran[1], Ganji Deepak[2], Mohammed Shoaib[3], Dr.P.Srinivas Rao[4]

[1,2,3]Scholar, Department of CSE in Jayamukhi institute of technological sciences, Narsampet, Warangal, Telangana, India

[4]Assistant professor, Department of CSE in Jayamukhi institute of technological sciences, Narsampet, Warangal, Telangana, India

ABSTRACT

IoT devices and systems are getting more intelligence and number of co as a direct result of the progress that has been made in Deep Learning technology. It is anticipated that they would carry out a variety of Deep Learning interpretation jobs with a high level of efficiency as well as performance. The disparity between the expected and actual results is a problem for this criterion. the limited computational power of edge devices and the large-scale nature of the Deep Neural Network - based. Collaboration e2 are then what you have. established in order to reduce the severity of this conflict and to help source of energy devices connected to the internet of things to host arbitrary applications for deep learning. On the other hand, the advent of clouds operated by third parties may bring concerns relating to privacy might be raised by edge computing. In this article, we will discuss systematically investigate the many possibilities for assaulting the target as well as the protection of users' personal information in collaborative edge-cloud environments. Our contributions may be broken down into two categories: (1) initially, we come up with a new set of assaults aimed against a cloud that cannot be trusted to recover arbitrary inputs pushed into the system itself, even in the event that the attacker does not possess access to the edge device's data, calculations, or rights to query this system are all examples of what may be requested We experimentally show that strategies that increase noise fail to Defeat the assaults that we have offered, and then provide two more effective alternatives. defence methods. This offers insights as well as guidance for the development process. a greater number of collaborative systems and algorithms that protect users' privacy.

I INTRODUCTION

The terms "Deep Learning" (DL) and "Internet of Things" (IoT) refer to two technologies that have seen explosive growth in recent years. The Internet of Things devices are becoming desirable targets for deep learning applications. They gather data and information from environmental settings using a variety of sensors (such as cameras, microphones, and gyroscopes), run the DL programmes to interpret the sensory input, and make control choices based on that interpretation. The combination of artificial intelligence with the internet of things ushers in the age of artificial intelligence of things (AIoT), which has brought about substantial shifts in our day-to-day lives. Small-scaled AIoT systems are implemented to build smart homes and increase the patients ' quality of life; medium-scale AIoT applications are equipped in large warehouses for increased performance and automation; and large-scale AIoT systems can contribute to the establishment of smart cities. [Citation needed].There are several obstacles to overcome when attempting to deploy deep learning inference applications on

commodity edge devices. On the one hand, an Internet of Things device is able to capture streaming data at a very rapid pace (e.g. vehicle detection [2], remote monitoring [3], scene analysis [4] and application trace analysis [5]). In order for this to be possible, the machine must quickly process the DL models and carefully examine the data. The state-of-the-art deep learning models, on the other hand, are becoming more complicated as their sizes increase, which makes it impossible for resource-constrained Connected systems to meet the performance requirements. The limited computation reserves of the device can result in significant latency; the limited data storage makes it difficult to store a large DNN model; and the limited battery capacity results in a critical power consumption restriction.

Offloading the complete deep learning model and inference processing to the cloud is one technique that might be used to successfully overcome this obstacle. The input data are sent to the cloud via the edge device, which then gets the output data. In spite of the fact that this has the potential to overcome the aforementioned constraints of edge devices, it does so at the expense of considerable increases in the communication expenses incurred when transferring a big amount of raw data. In addition, there is a risk of a breach in the integrity of the model [7] and a breach in the privacy of the inference data [6] if the cloud cannot be relied upon. This risk is particularly significant if the input data are very sensitive, such as records of patients.

The use of collaborative inference between edge devices and the cloud is a method that, when optimised, will provide the best results [8, 9], [10], [11], [12]. The DL model may be broken up into two distinct components. The local edge device is responsible for storing the network's first few layers, while the remaining layers are offloaded to a cloud located farther away. When an input is provided, the edge device does the calculations necessary to determine the output of the first layers, then it uploads those calculations to the cloud and obtains the final results. This strategy has the potential to cut down on communication expenses since the intermediate output may be made to be much more compact than the raw input. A data transport bandwidth that low also results in a lower overall latency and a lower overall energy usage. Collaborative inference makes it possible to deploy large-scale intelligent workloads on the edge platforms of today in a way that is both viable and efficient.

The purpose of this work is to report the findings of a study into the inference data privacy in edge-cloud collaborative systems, from the points of view of both assaults and countermeasures. Previous studies ignored possible safety concerns in favour of optimising the performance and efficiency of such systems, but all of those efforts had the same goal. We believe that we are the first people to show that it is possible to launch input data privacy assaults against cloud-edge collaboration directly or indirectly affected.

The confidentiality of the raw inputs is the aspect of data privacy that is taken into consideration in this study. We demonstrate that a cloud that cannot be trusted can nonetheless readily and correctly retrieve sensitive data from intermediate values even when it does not have access to the edge side model.

In order to accomplish this objective in a variety of contexts, we devise a collection of innovative offensive strategies. First, in order to retrieve the samples from the

model parameters and intermediate values for a white-box attacker, we suggest adopting a method called Regularized Maximum Likelihood Estimation. Second, we suggest the Inverse Network attack as a method for a black-box attacker to use in order to determine the reverse mapping from the intermediate outputs to the inputs without having prior knowledge of the information about the model. Third, we take into consideration the most restricted antagonistic capability, which involves the cloud not having any prior knowledge of the target model and being prohibited from interrogating the model.

It is highly challenging to carry out privacy assaults in this environment, and this threat model has been examined in previous research only very seldom. In order to accomplish these query-free assaults, we provide a novel approach to Shadow Modelling Reconstruction as the means by which to do so.

In this work, we answer the second concern that was posed to us, which is: how can the edge devices prevent privacy leakage from the untrusted cloud? In previous work, differential privacy was used to ensure the safety of the inference data [13]. We demonstrate that this strategy cannot be used against the assaults that we have provided since it results in an unacceptable reduction in the DL models' levels of performance. Instead, we offer two innovative solutions that may better defend against the privacy threats while still keeping the model's performance at a satisfactory level. The first strategy is known as the dropout defence, and it involves killing random neurons while the adversary is making an inference. This prevents the adversary from accurately generating the initial pictures from the intermediate values.

Our second line of defence is a privacy-aware DNN partitioning strategy. In this strategy, we do an exhaustive analysis of the many aspects that might influence the outcome of an attack, and we provide some suggestions for how to divide the deep learning models in order to improve privacy. We have high hopes that our results will be able to direct academics and practitioners of machine learning toward the development of more trustworthy collaborative inference systems.

• A comprehensive examination of assaults and responses for inference data privacy in angle collaboration machine learning systems. These are the primary contributions of this work.

• Three alternative methods of attack to extract inference data under a variety of different conditions

• Two innovative defensive strategies to stop data from drawing inferences that are sent to an untrusted cloud.

II RELATED WORK

Machine Learning Privacy Attacks

Attacks on the privacy of data training. There are many sorts of privacy threats against the training data. The first kind of attack is called a property inference attack, and its goal is to determine certain aspects of the training data by analysing the model's parameters. Traditional machine learning classifiers [25] and fully-connected neural networks [26] have both been subjected to attack demonstrations.

Membership inference attacks are a kind of property inference attacks that determine whether or not an individual sample is a part of the training set by inferring whether or not it is included. This defence was described for the first time in [27]. The subsequent research investigated whether or whether it was possible to launch

assaults using a variety of adversary capabilities [28], model characteristics [29], [30], in Generative Adversarial Networks [31], [32], and collaborative training systems [33]. the softmax layer in the black-box situation. Even in a more stringent query-free situation, we demonstrate that the adversary may effectively invert the model from multiple levels of the architecture. In addition, we provide defensive tactics that are not included in the work in question.

Wei et al. [40] employed a power side channel to retrieve inference data. However, in order to carry out this attack, the adversary needed to first get access to the victim device in order to acquire side-channel information. Additionally, the attack could only retrieve basic pictures (single pixels). Our technique is capable of recovering any arbitrary or sophisticated data without requiring access to the victim's equipment or knowledge of the victim's computation.

Privacy Solutions Based on Machine Learning

Improving the underlying algorithms. Because various participants might utilise their own data for model training, distributed training was developed as a means of ensuring the confidentiality of the training data [41], [42]. Protecting the training tasks from privileged attackers was accomplished by using the SGX security enclaves that are available in Intel CPUs [43], [44]. Cao et al. [45] came up with a way to get rid of the influence that sensitive training samples have on the models. Abadi et al. [46] used differential privacy to include noise into the stochastic gradient descent process in order to get rid of the parameters' dependence on the training data.

improving the dataset used for training. Bost et al. [47] proposed \sto encrypt the data before feeding it into the training algorithm. They devised machine learning operators that are capable of working on encrypted data. Zhang et al. [48] demonstrated that the addition of noise to the training dataset is an excellent method for maintaining the confidentiality of training data. Generating generated data [49], [50], [51] \shas been suggested for training DNN models while removing \ssensitive information from the real data.

Obfruscating the inference input. Differential privacy, which involves the addition of random noise to the input, has been suggested as a method for protecting model inference [22, 23]. We demonstrate that it is not enough to simply add noise in order to defend against our assaults, and as a result, we present two responses that could be more applicable to our attacks in this work. Recent study [6] suggested to include specially \sdesigned noise and gave a theoretical analysis on the input \sdata privacy leakage. On the other hand, it did not take into account the model inversion assaults that we present, and therefore requires further training for the noise generator.

Homomorphic encryption. This enables the inference application running on the untrusted participant to directly execute DNN calculations on encrypted input [52], [53]. As a result, the sensitive information will not be disclosed to unauthorised parties. The homomorphic encryption method has a number of flaws, the most significant of which is that it is very inefficient and cannot be used for all DNN operations.

Model inversion attacks [34] are the second form of attack against the privacy of the training data. Given a machine learning model and some of the

characteristics of the training samples, an adversary may retrieve the remainder of the training samples' features. In single-party systems [35] and collaborative learning systems [36], advanced model inversion attacks were proposed to extract pictures from deep neural networks. [35]

The third kind of attack is called a model encoding attack [37], and it occurs when an adversary who has direct access to the training data attempts to encode sensitive material into a model such that it may be retrieved by another entity.

Model privacy threats. The attacker makes an effort to steal the model's parameters [18], hyperparameters [20], or structures [19], [38] by using prediction APIs, memory side channels, and other such mechanisms.

Attacks on the privacy of inferenced data The research described in [39], which trains an inverse network on the output probability distribution in order to get the inversed inference data, is conceptually closer to our own study.

Nonetheless, the only model inverting attack they take into consideration is from

III PROPOSED SYSTEM

In light of the severity of privacy inference assaults that may occur in edge-cloud collaborative systems, the purpose of this section is to investigate possible defences. First, we perform an empirical evaluation of one common method that has been proposed in previous work. This method is known as noise obfuscation, and it did not specifically target the edge-cloud privacy attacks. We demonstrate that it is not successful in protecting inference data privacy against our assaults. Then, we provide two innovative solutions that have the potential to better avoid the privacy leaks while having a modest influence on the performance and capabilities of the system.

A. Disguise via the Use of Random Noise

Differential privacy, which involves the addition of random noise to the input, has been suggested as a method for protecting model inference [22, 23].

In the edge-cloud scenario, we have the option of either adding noise to the initial input, which would look like this: v = f1 (x +), or adding noise straight to the intermediate value, which would look like this: v = f1 (x) +. This would be done before transmitting the value to the untrusted cloud C. There is a trade-off that must be made between usefulness and privacy: the accuracy of the model will decrease when a larger degree of noise is included. In order for this strategy to be successful, it is essential to determine whether or not this compromise can be struck. In the following, we test the consequences of the assault as well as the accuracy of the model by employing noise obfuscation. In our experiments, we take into consideration both Gaussian and Laplacian noise. Figures 6a, 6b (Gaussian) and 7a, 7b (Laplacian) provide a visual representation of the recovered images on the MNIST and CIFAR10 datasets after varying amounts of noise were added to either the input (the first two rows in each figure) or the intermediate layer output. These figures show the recovered images on the MNIST and CIFAR10 datasets (last two rows). We have shown that increasing the amount of noise to an appropriate level may, in fact, improve privacy while simultaneously lowering the quality of photos that are recovered. In addition, noise introduced at the primary input has a greater impact than noise introduced at an intermediary layer.

In Figures 9 and 10, we present a quantitative analysis of model accuracy (usability, shown on the y-axis) and inversed image quality (privacy, shown on the x-axis) using the MNIST dataset and the CIFAR10 dataset, respectively. The Gaussian noise is represented by the blue lines, while the Laplacian noise is represented by the orange lines. The process of adding noise to the input layer and the intermediate layer is shown by solid lines and dotted lines, respectively. The optimal area of the graph may be found in the upper left corner. If the noise is introduced to the input (the blue and orange solid lines), then the model accuracy will decline more than it would if the noise is added to the intermediate layer. This is because restoring the recovered image quality (SSIM or PSNR) requires repairing the picture quality (blue and orange dotted lines). This corroborates the findings that can be seen in Figures 6a and 7a.

The accuracy of the model does not significantly change depending on the parameters of the noise distribution, such as whether the noise is Gaussian or Laplacian.

On the MNIST dataset (Figure 9), the noise level must be constrained to be less than 0.8 and b must be less than 0.5 in order to keep a decent model accuracy (that is, more than 95 percent). Even at this point, the adversary is able to recover pictures of a high quality (SSIM > 0.4 and PSNR > 8.5 dB). Figure 10 demonstrates that the CIFAR10 dataset also yielded comparable findings. Recent research [13], [6] offered unique strategies for creating noise to safeguard inference data privacy. Despite this, it is still possible that these algorithms may not work for our novel assaults; rather, the noise generator will need additional extra-special training. As a result, we will now suggest some novel protection techniques below that do not rely on the addition of noise and are more practical in the sense that they preserve the privacy of the inference data with a much reduced impact on performance.

In light of the fact that noise obfuscation may not be a safe method, we offer an additional randomization-based technique called dropout as a means of protecting against the presented assaults. By causing the output of random neurons in one layer to become zero, dropout deactivates those neurons. In mathematical terms, the calculation goes as follows: fi dropout(x) = f(x) M (6), where M is a mask in which each element is arbitrarily given a value of 0 with probability r and a value of 1 with probability 1-r. means element-wise multiplication.

Intuitively, dropout makes use of the redundancy property of neural networks [24]. As a result, deleting partial information in the inference does not affect the performance of the model, but it does obscure the input data.

In a manner similar to that of noise obfuscation, dropout may also be applied to either the input or the output of the intermediate layer. In Figure 8a, we provide some instances of the pictures that were recovered from the ReLU2 (MNIST) layer. The impact that dropout has on the input is shown in the top two rows, and the influence that it has on the intermediate output is shown in the bottom two rows. We have found that a rise in the dropout rate, denoted by r, results in a decline in the quality of the inversed pictures. When r hits 0.6, the attacker will be unable to gain any information that is of any value to them. Figure 8b displays pictures that have been flipped horizontally from ReLU22 (CIFAR10).

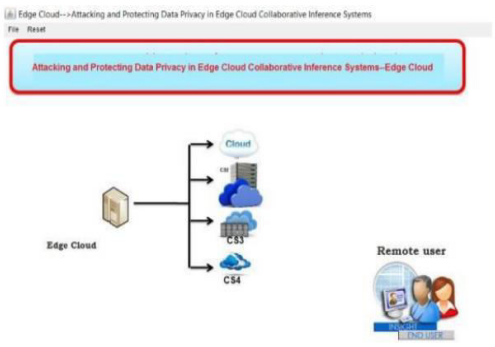
When r hits 0.6, similarly, there is no way to gather any information that is of any value. The usability against privacy trade-off of dropout is further measured, and then compared to the noise obfuscation strategy (Figure 9 on MNIST and Figure 10 on CIFAR10). A higher accuracy indicates improved usefulness, whereas a lower SSIM and PSNR indicate increased levels of privacy protection. The lines that are situated higher up and to the left provide a more favourable trade-off. It has come to our attention that dropout (shown by the green lines) performs noticeably better than any of the noise obfuscation options (blue and orange lines).

This is due to the fact that dropout uses the redundancy of the DNN model to conceal partial information while maintaining model accuracy, but introducing random noise adds obfuscation on all neurons, which reduces model accuracy. In addition, dropout on the intermediate layer (shown by the green dotted line) is a modest improvement over dropout on the input (represented by the green solid line) in that it may completely safeguard the inference data privacy (SSIM less than 0.25) with an accuracy more than 95%. Dropout on the intermediate layer greatly outperforms the other algorithms on the CIFAR10 dataset, completely safeguarding inference data privacy (SSIM0.25) with just a 0.8 percent decrease in accuracy. In order to conduct an exhaustive analysis of the performance of this dropout mechanism, we are contemplating the possibility of breaking the model into many parts throughout the MNIST dataset.

Since dropout on the intermediate layer (also known as split layer) is superior than dropout on the input layer, we do it there instead. The recovered pictures are shown in order from shallow to deep layers. They are as follows: ReLU1, pool1, conv2, ReLU2, pool2. We have shown that as the depth of the split layer increases, a lower dropout rate is adequate to avoid the disclosure of private information. For instance, when the model is divided at the ReLU1 layer (the first row), r may be set to 0.9 to completely obscure the input, but when the modeling is split at the pool2 layer, r can be set to 0.2 instead (last row). This is more clearly shown in the usability-privacy curves in Figure 13, which are as follows: The dropout that occurs on deeper levels has a greater impact (and is thus closer to the top left areas) than that which occurs on shallower layers.

Following is the order of performance for SSIM and PSNR, beginning with the worst: pool1 (blue), then conv2 (green), then ReLU2 (orange), and finally pool2 (grey). The ReLU1 layer is the only one that deviates from the rule; it performs far better than anticipated. It is superior than conv2 in terms of PSNR performance and is the best option for SSIM. The fact that the recovered picture maintains its structure identifiable to the human eye despite a decrease in illumination in the ReLU1 layer, which contributes more heavily to PSNR and SSIM than human recognition, is one potential explanation for this phenomenon.

IV RESULTS



Cloud Server-1

Cloud Server-2

V CONCLUSIONS

In this research, we investigate the potential privacy risks posed by inference data in collaborative edge-cloud environments. We came to the conclusion that an untrusted cloud might readily reconstruct the inference samples using the possible results. We present a new set of attack approaches as a means of breaching the confidentiality of inference data using a range of different assault settings. We show that it is possible for the adversary to successfully and dependably retrieve the inputs with just a small number of requirements. Additionally, we provide a few different ways to safeguard the confidentiality of the inference data for edge computing.

In all of the previous work, the performance, efficiency, and functionality of Artificial Intelligence of Things have been the primary focuses, but privacy concerns have been completely ignored. We have high hopes that our research will help make people aware about how important it is to secure the anonymity of inferences data in edge-cloud systems, and we also hope that it will motivate designers and implementers of such systems to strike a balance between information privacy and accessibility.

REFERENCES

[1] Z. He, T. Zhang, and R. B. Lee, “Model inversion attacks against collaborative inference,” in Proceedings of the 35th Annual Computer Security Applications Conference, 2019, pp. 148–162.

[2] Y. Tang, C. Zhang, R. Gu, P. Li, and B. Yang, “Vehicle detection and recognition for intelligent traffic surveillance system,” Multimedia tools and applications, vol. 76, no. 4, pp. 5817–5832, 2017.

[3] G. Chen, T. X. Han, Z. He, R. Kays, and T. Forrester, “Deep convolutional neural

network based species recognition for wild animal monitoring," in 2014 IEEE International Conference on Image Processing (ICIP). IEEE, 2014, pp. 858–862.

[4] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 833– 841.

[5] L. Xiao, Y. Li, X. Huang, and X. Du, "Cloud-based malware detection game for mobile devices with offloading," IEEE Transactions on Mobile Computing, vol. 16, no. 10, pp. 2742–2750, 2017.

[6] F. Mireshghallah, M. Taram, P. Ramrakhyani, A. Jalali, D. Tullsen, and H. Esmaeilzadeh, "Shredder: Learning noise distributions to protect inference privacy," in Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems, 2020, pp. 3–18.

[7] Z. He, T. Zhang, and R. Lee, "Sensitive-sample fingerprinting of deep neural networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4729–4737.

[8] J. Hauswald, T. Manville, Q. Zheng, R. Dreslinski, C. Chakrabarti, and T. Mudge, "A hybrid approach to offloading mobile image classification," in 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014, pp. 8375–8379.

[9] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, and L. Tang, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," AcmSigplan Notices, vol. 52, no. 4, pp. 615–629, 2017.

[10] S. Teerapittayanon, B. McDanel, and H. Kung, "Distributed deep neural networks over the cloud, the edge and end devices," in IEEE International Conference on Distributed Computing Systems, 2017.

[11] J. H. Ko, T. Na, M. F. Amir, and S. Mukhopadhyay, "Edge-host partitioning of deep neural networks with feature space encoding for resource-constrained internet-of-things platforms," in IEEE International Conference on Advanced Video and Signal Based Surveillance, 2018.

[12] A. E. Eshratifar, M. S. Abrishami, and M. Pedram, "Jointdnn: an efficient training and inference engine for intelligent mobile cloud computing services," arXiv preprint arXiv:1801.08618, 2018.

[13] F. Mireshghallah, M. Taram, A. Jalali, A. T. Elthakeb, D. Tullsen, and H. Esmaeilzadeh, "A principled approach to learning stochastic representations for privacy in deep neural inference," arXiv preprint arXiv:2003.12154, 2020.

[14] https://pytorch.org/docs/0.4.0/torchvision/datasets.html, 2018.

[15] https://en.wikipedia.org/wiki/Peak-signal-to-noise-ratio, 2018.

[16] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli et al., "Image quality assessment: from error visibility to structural similarity," IEEE transactions on image processing, vol. 13, no. 4, pp. 600–612, 2004.

[17] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation-based noise removal algorithms," Physica D: nonlinear phenomena, vol. 60, no. 1-4, pp. 259–268, 1992.

[18] F. Tramer, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing ` machine learning models via prediction apis." in USENIX Security Symposium, 2016.

[19] S. J. Oh, M. Augustin, M. Fritz, and B. Schiele, "Towards

[20] B. Wang and N. Z. Gong, "Stealing hyperparameters in machine learning," in IEEE Symposium on Security and Privacy, 2018.

[21] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward

neural networks," in Proceedings of the thirteenth international conference on artificial intelligence and statistics, 2010, pp. 249–256.

[22] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in Theory of cryptography conference. Springer, 2006, pp. 265–284.

[23] C. Dwork, A. Roth et al., "The algorithmic foundations of differential privacy," Foundations and Trends in Theoretical Computer Science, vol. 9, no. 3–4, pp. 211–407, 2014.

[24] Y. Cheng, F. X. Yu, R. S. Feris, S. Kumar, A. Choudhary, and S.- F. Chang, "An exploration of parameter redundancy in deep networks with circulant projections," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2857–2865.

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

[26] K. Ganju, Q. Wang, W. Yang, C. A. Gunter, and N. Borisov, "Property inference attacks on fully connected neural networks using permutation invariant representations," in ACM conference on computer and communications security, 2018, pp. 619–633.

[27] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in IEEE Symposium on Security and Privacy, 2017.

[28] A. Salem, Y. Zhang, M. Humbert, M. Fritz, and M. Backes, "Mlleaks: Model and data independent membership inference attacks and defenses on machine learning models," in Network and Distributed System Security Symposium, 2018.

[29] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in IEEE Computer Security Foundations Symposium, 2018.

[30] Y. Long, V. Bindschaedler, L. Wang, D. Bu, X. Wang, H. Tang, C. A. Gunter, and K. Chen, "Understanding membership inferences on wellgeneralized learning models," arXiv preprint arXiv:1802.04889, 2018.

[31] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, "Logan: evaluating privacy leakage of generative models using generative adversarial networks," arXiv preprint arXiv:1705.07663, 2017.

[32] K. S. Liu, B. Li, and J. Gao, "Generative model: Membership attack, generalization and diversity," arXiv preprint arXiv:1805.09898, 2018.

[33] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in IEEE Symposium on Security and Privacy, 2019.

[34] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing." in USENIX Security Symposium, 2014.

[35] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in ACM Conference on Computer and Communications Security, 2015.

[36] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the ´ gan: information leakage from collaborative deep learning," in ACM Conference on Computer and Communications Security, 2017.

[37] C. Song, T. Ristenpart, and V. Shmatikov, "Machine learning models that remember too much," in ACM Conference on Computer and Communications Security, 2017.

[38] W. Hua, Z. Zhang, and G. E. Suh, "Reverse engineering convolutional neural networks through side-channel information leaks," in ACM/ESDA/IEEE Design Automation Conference, 2018.

[39] Z. Yang, J. Zhang, E.-C. Chang, and Z. Liang, "Neural network inversion in adversarial setting via background knowledge alignment," in Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 225–240.

[40] L. Wei, B. Luo, Y. Li, Y. Liu, and Q. Xu, "I know what you see: Power side-channel attack on convolutional neural network accelerators," in Annual Computer Security Applications Conference, 2018.

[41] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in ACM conference on computer and communications security, 2015.

[42] J. Hamm, A. C. Champion, G. Chen, M. Belkin, and D. Xuan, "Crowdml: A privacy-preserving learning framework for a crowd of smart devices," in IEEE International Conference on Distributed Computing Systems, 2015.

[43] O. Ohrimenko, F. Schuster, C. Fournet, A. Mehta, S. Nowozin, K. Vaswani, and M. Costa, "Oblivious multi-party machine learning on trusted processors." in USENIX Security Symposium, 2016.

[44] T. Hunt, C. Song, R. Shokri, V. Shmatikov, and E. Witchel, "Chiron: Privacy-preserving machine learning as a service," arXiv preprint arXiv:1803.05961, 2018.

[45] Y. Cao and J. Yang, "Towards making systems forget with machine unlearning," in IEEE Symposium on Security and Privacy, 2015.

[46] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in ACM Conference on Computer and Communications Security, 2016.

[47] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data." in Network and Distributed System Security Symposium, 2015.

[48] T. Zhang, Z. He, and R. B. Lee, "Privacy-preserving machine learning through data obfuscation," arXiv preprint arXiv:1807.01860, 2018.

[49] A. Triastcyn and B. Faltings, "Generating artificial data for private deep learning," arXiv preprint arXiv:1803.03148, 2018.

[50] X. Zhang, S. Ji, and T. Wang, "Differentially private releasing via deep generative model (technical report)," arXiv preprint arXiv:1801.01594, 2018.

[51] H. Yin, P. Molchanov, Z. Li, J. M. Alvarez, A. Mallya, D. Hoiem, N. K. Jha, and J. Kautz, "Dreaming to distill: Data-free knowledge transfer via deepinversion," arXiv preprint arXiv:1912.08795, 2019.

[52] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in International Conference on Machine Learning, 2016, pp. 201–210.

[53] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, "Gazelle: A low latency framework for secure neural network inference," in 27th USENIX Security Symposium, 2018, pp. 1651–1669.