Application for privacy and security in Deep learning

*

Patrick NONKI
Department of Electronics Engineering
Hochschule Hamm-Lippstadt
Lippstadt, Germany
patrick.nonki@stud.hshl.de

***Abstract*—Nowadays, a large amount of data is being generated for usage in big companies like Google, Siemens, Meta and it arose the challenge to deal with them in a context of making them a much as secure and private possible. Since the challenge of keeping them confidential with applications in a very wide range of applications like face recognition, farming, or path recognition, deep learning which is a part of the big family of Machine learning, could be a potential solution since it enables to deal with a lot of data at the same time and in a secure way since most of these companies deal with sensitive data of customers (for example private images, geographic positions, locations and passwords).**

DL's privacy problems have recently come to light, and a number of attacks have been suggested. Model extraction attacks and model inversion attacks are two types of attacks that violate a model's privacy. In model extraction attacks, the adversary seeks to replicate the parameters/hyperparameters of the model that is used to deliver cloud-based ML services, jeopardizing the confidentiality of the ML algorithms and the service provider's intellectual property. Attackers use information already accessible to infer sensitive information in model inversion attacks. [1] Several strategies have been put out to address privacy and security concerns in DL. In terms of DL privacy for privacy-preserving, there are now four primary technologies: differential privacy, homomorphic encryption, secure multi-party computation, and trusted execution environment. By using differential privacy, the adversary is prevented from deducing whether a certain instance was used to train the target model. The privacy of the training and testing data is prioritized by the homomorphic encryption and secure multi-party computation scheme. In order to safeguard training code and sensitive data, the trusted execution environment tries to employ hardware to create a secure and isolated environment. [1]

***Index Terms*—Deep Learning, Defense, attack, training, layers, network**

I. Motivation

Deep Learning has recently demonstrated great performance in a variety of fields, including image recognition, pattern matching, and even cybersecurity. Deep learning has many benefits, including the ability to solve complex problems quickly, a great deal of automation, the greatest use of unstructured data, the ability to produce results of a high caliber, the elimination of high costs, the elimination of the need for data labeling, and the identification of complex interactions. However, it also has drawbacks, including being opaque, computationally intensive, requiring a lot of data, and having more complicated algorithms. Many applications that we use every day to make decisions based on predictions use deep learning models, and if these models were to mispredict the future due to malevolent internal or external factors, it might cause problems in our daily lives. Furthermore, the Deep Learning training models frequently contain sensitive user data, so those models shouldn't be exposed to security and privacy risks. Deep Learning and machine learning algorithms are still susceptible to many dangers and threats to their security. Since it is important to be aware of security threats and related countermeasures approaches for deep learning, the seminar paper undertook a thorough survey of the difficulties and solutions connected to deep learning security and privacy. [2]

II. The types of attacks on security

A. Model Extraction Attack

Model extraction attacks undermine the secrecy of the ML method and the learner's intellectual property since the adversary seeks to take parameters of the target model having black-box access to the target model. [1]

B. Model Inversion Attack

In model inversion attacks, the adversary seeks to reveal the secrecy of private records that were utilized as part of the training set by using model predictions. [1]

C. Adversarial Attack

The goal of adversarial attacks is to create an adversarial example using the target model's knowledge, which causes the target model to make a highly confident erroneous prediction. An adversarial sample is a manipulated image created by the addition of undetectable noise that can lead the classifier to confidently make incorrect predictions. Additionally, transferability refers to the likelihood that an adversarial sample developed for one model may work well for other models. The two types of adversarial attack depend on the opponent's objective. [1]

1) **Non-targeted attack**: The adversary crafts adversarial examples xadv to cause the target model f to misclassify the input with high confidence, but does not require the prediction to be specified class, that is, $f(x_{adv}) \neq y_{true}$, where $f(x_{adv})$ can be any class except the correct class y_{true} . [1]
2) **Targeted attack** : The adversary crafts adversarial examples xadv to cause the target model f to misclassify the input with high confidence into a particular class t specified by the adversary, that is, $f(x_{adv}) = t$, where t is not correct class y_{true} [1]

D. Poisoning Attack

In poisoning attacks, the attacker tries to contaminate the training data such that the learner develops a subpar classifier that will incorrectly categorize harmful samples or actions created by the attacker at the testing stage. The antagonist may introduce harmful samples, change data labels, and damage the training data. The poisoning attack can be divided into three groups based on the objectives of the enemy. [1]

1) **Accuracy drop attack**: The adversary aims to disrupt the training process by injecting malicious samples to reduce the performance of the target model at the testing stage [1]
2) **Target misclassfication attack**: The adversary aims to enforce test samples to be misclassified at the testing stage [1]
3) **Backdoor attack**: The adversary aims to install a backdoor with a specific mark so that the target model has a target output for that particular input [1]

III. DL Algorithms of defence in security [2]

A. Defense on Evasion attacks

Enhancing adversarial instances, detecting adversarial examples, adversarial training, and defensive distillation are the best ways to defend against an evasion attempt. [2]

1) Detecting adversarial attacks: To develop diverse benign and hostile instances as well as to identify adversarial examples in the input, some scholars presented various strategies. The attacker's goal, as we previously discussed, is to increase the amount of noise in order to create powerful adversarial examples. They claim that it is difficult to identify such adaptive attacks and that certain detection methods are successful while others are unsuccessful. The testing example that is used to forecast the adversarial example is highly difficult to handle, and thus makes it difficult to discover hostile examples. As a result, the expert should manually label the test samples. [2]

They suggested a method for validating hostile instances using testing examples and the testing example template. The authors claim that labeling the classifier is not necessary if the adversarial example is demonstrated by testing examples during verification; nevertheless, if not predicted, testing examples must be reformed using the reformer by removing extraneous noise from the testing example. When this operation is finished, the classifier will label the testing example for the deep neural network and will regard it as a legitimate testing example. [2]

2) Adversarial training: Researchers described a method for training a Deep Neural Network using a growing training dataset and a number of hostile examples. They called it adversarial training. The author suggested training benign examples against each training adversarial example to combat the evasion assault. Through the initial benign example and the attack adversarial example, the system learner will use the backpropagation technique to gain knowledge of the Deep Neural Network. The antagonistic training variations were also suggested by the writers listed below. To handle problems involving min-max optimization, the authors employed robust optimization approaches. Accuracy in the benign example is the main problem in adversarial training. [2]

B. Defense against Poisoning Attack

The framework proposed by one researcher adopts the strategy of excluding extreme values that are outside the pertinent group. They look for the midpoints of the positive and negative categories in the binary classification. The points that are not close to the important focal point are then removed by the authors. They employ the defense field, which eliminates points outside the ball's radius, and a slab defense, which complementarily ignores points away from the line, to gather information about these spots. [2]

Instead of eliminating the data points with foreign values, some researchers choose to rename them. Attack flipping label is an exclusive tool for data poisoning that enables an attacker or hacker to manage the appointment of a meager number of training points. The author goes on to describe a method that reclassifies points determined to be detrimental and investigates points outside the scope of the resolution. Every case's label is reset as part of the operation. [2]

Through remote sensing, other researchers also suggest a defense mechanism to lessen the severity of poisoning attacks. With a paltry amount of poison points, the label makes an effort to exert the greatest impact possible over the guardian. Every x in the initial data set has its external result computed as part of the external detection process. Additionally, there are several and distinctive ways to determine the outward result. [2]

IV. The types of attacks on privacy

A. Model Extraction Attack

In poisoning attacks, the attacker tries to contaminate the training data such that the learner develops a subpar classifier that will incorrectly categorize harmful samples or actions created by the attacker at the testing stage. The antagonist may introduce harmful samples, change data labels, and damage the training data. The poisoning attack can be divided into three groups based on the objectives of the enemy. [1]

Additionally, attacks to steal the hyperparameter of the target model are being considered. Given that the objective of the ML algorithm's learning process is to reach minima of the objective function when the gradient of the objective function is close to 0, the adversary can create multiple linear equations by running a large number of queries based on this observation. Finally, linear least squares can be used to estimate the hyperparameters. They used empirical evidence to show that their technique could accurately steal hyperparameters from regression algorithms with less than 10^{-4} error. [1] Watermarks for DNN have been created as a deterrent to potential intellectual property thefts and integrate watermarks into the DL model. It is demonstrated that when the embedded watermark length rises, the watermarking approach raises the standard deviation of the weight distribution. A watermark detection algorithm is suggested based on this discovery, and the watermark can subsequently be removed using the knowledge that is already accessible. However, when embedding into DL models, the watermarking's detachability and overwriting are frequently taken into account. Therefore, two unique evasion strategies have been developed to enable an adversary to use an MLaaS with stolen ML models while avoiding detection by the rightful owners of those ML models, providing that the watermarking may not be deleted. [1]

B. Model Inversion Attack

A Membership Inference Attack (MIA) against ML models has been proposed. In order to discern between the behavior of the target model on the sample for its trained sample and its untrained sample, the adversary trains an attack model. The attack model is a classification model, in other words. The author created a novel method called "shadow training" that creates numerous shadow models to replicate the target model in order to generate such an attack model in a "black-box" environment. They employ the input/output pairs of the shadow model to train an attack model since the target model's data distribution is unknown. The trials showed that an adversary having black-box access to the target model may successfully carry out an MIA. However, it is noted that MIA performs well when the model is significantly overfitted to the training set of data. The overfitted model's prediction for the query on the training sample differs noticeably from other queries' predictions (probabilities). The behavior of a well-generalized model is similar to both training and test sets. As a result, under well-generated models, not all data is susceptible to member inference attacks. A Generalized Membership Inference Attack (GMIA) against a well-generalized model is put forth to address this problem. In order to carry out inference attacks, the goal of this assault is to locate target records that are susceptible to member inference attacks. [1]

Too many assumptions are made in the MIA that is being proposed, including the use of numerous shadow models, knowledge of the target model, and data distribution that is similar to the target model's training data. These presumptions are relaxed, and three low-cost, widely applicable adversaries are suggested. Instead of using many shadow models to replicate the target model, the first adversary uses just one. The training data of the target model and its architecture are not directly accessible to the second adversary. The third enemy has the bare minimum of presumptions. It is more practicable in real-world scenarios not to develop any shadow models and to be familiar with the target model's structure. [1]

Additionally, an MIA is suggested to conduct white-box attacks when used with collaborative DL models. To create a Generative Adversarial Network, they built a generator (GAN). Following training, GAN can produce artificial data that is comparable to the target model's training data. The drawback of this strategy is that because all training data from the same category must have a similar visual appearance, they cannot be discriminated under the same distribution. A MIA against models of collaborative learning was also put forth. Collaborative learning is a learning technique in which two or more participants, each of whom has their own training datasets, cooperatively train a joint model by training locally and frequently communicating model updates. However, these updates may unintentionally reveal information about the participants' training records. The embedding layer is first used to convert the input into a low-dimensional vector representation in various non-numeric data contexts, such as natural language processing, where the update of the provided matrix depends simply on whether the word appears in the batch. Therefore, this character can be utilized to create a property inference attack because it immediately discloses if a word appears in the training batches. [1]

V. DL Algorithms of defence in privacy

A. Differential Privacy

Depending on where the noise is added, DP approaches can be classified into three categories: gradient perturbation, objective perturbation, and label perturbation [1]

1) Gradient perturbation: The gradient perturbation is done by injecting noise to the gradients of the parameters during the training stage. [1]

To train DNN with non-convex objectives, the Differentially Private Stochastic Gradient Descent (DPSGD) technique is suggested. At each stage of the stochastic gradient descent process, noise is primarily introduced into the gradient. In addition, a more robust accounting technique known as the moment accountant was created to obtain the tail bound. The moment accountant tracks the cumulative privacy loss using the moments bound along with the Markov inequality, which offers a more accurate accounting of privacy loss than composition theorems. A Differentially Private Generative Adversarial Network (DPGAN) is described that ensures (ϵ,θ) -DP by causing the discriminator's gradient to change during the training process. The output of the differentially private discriminator will not violate privacy, in accordance with the post-processing theory [58], proving that the generator is similarly differentially private. A unique Differentially Private Generative Model (DPGM) has been developed that relies on a combination of k generative neural networks, including variational autoencoders and limited Boltzmann machines [60]. The differential private kernel k-means algorithm is used to cluster the dataset, and then k generative neural networks are allocated to each cluster. k generative neural networks are trained using the DP-SGD, and they are able to create artificial high-dimensional data while maintaining proven privacy. [1]

2) Objective perturbation: The objective perturbation is done by injecting noise to the objective function and solving a precise solution to the new problem [1]
A Deep Private Autoencoder (DPA) has been developed to enforce ϵ-DP by interfering with the classic deep autoencoder's objective functions. By utilizing the Chebyshev expansion in convolutional deep belief networks, it is suggested that the Private Convolutional Deep Belief Network (PCDBN) be used to enforce -DP perturbing the polynomial forms that approximate the non-linear objective function. Additionally, the Adaptive Laplace Approach (AdLM), a cutting-edge mechanism to ensure DP in DNN, has been proposed. This method is easily extended to various differential DNNs and adds noise from the input features to the model output in an adaptive manner. The privacy loss caused by objective perturbation, as opposed to gradient perturbation, is decided at the time the objective function is created and is independent of epoch. Gradient perturbation accumulates privacy loss as training advances. [1]

3) Label perturbation: The information of an ensemble of ”teacher” models is transferred to a ”student” model using the privacy-preserving method known as Private Aggregation of Teacher Ensembles (PATE). In order to train a student model using publicly available data that has been tagged using the ensemble of teacher model, a group of teachers first learns on diverse subsets of the sensitive data. Due to the student model's inability to access sensitive data directly and the injection of differential private noise into the aggregation process, sensitive data privacy is preserved. As part of the learning process, an accountant is also introduced at this point to track the totaled privacy budget. However, PATE's effectiveness is assessed using straightforward categorization tasks. Later, a new aggregation approach is also put forth that effectively extends PATE to the large-scale learning job. Additionally, it has been empirically demonstrated that the modified PATE has higher utility than the original PATE and assures a tighter DP. Furthermore, the differential private GAN framework was built using The PATE. Differential privacy serves as both the discriminator and the discriminator-trained generator. This method's drawback is that it necessitates further teacher model training in order to instruct the student model. [1]

B. Homomorphic Encryption

The HE is mostly utilized in DL to train neural network models and safeguard testing inputs and outcomes. The decline in efficiency, the high ciphertext computation cost, and the significant increase in data volume after encryption are the main drawbacks of using HE. To achieve strictly linear complexity in the depth of the neural network, FHE-DiNN uses the bootstrapping technique. It was noticed that the suggested FHE approach only permits operations on binary data, allowing for the computation of all operations for the binary neural network (BNN). To speed up binary processing in BNN, several people created Tricks to Accelerate (encrypted) Prediction As a Service (TAPAS). The results of the studies indicated that TAPAS significantly reduces the length of the evaluation stage. [1]

C. Secure Multi-Party Computation

SMC in DL has two primary application scenarios. In the first situation, the data owner does not want to provide the server access to all of the training data in order to develop or infer the DL model. In order to train/infer the DL model jointly, he/she intends to distribute the training/testing data across a number of servers. The training/testing data of other servers won't be understood by any one server. The second case is where various data owners desire to collaboratively train a shared DL model on aggregate training data while maintaining the privacy of their individual training data. [1]
A viable system for collaborative deep learning was developed by some, allowing several users to jointly construct neural network models based on their inputs without sharing those inputs and reducing model accuracy. After each round of local training, the participants asynchronously exchange the gradients they computed for a subset of the parameters they used to train their local model. The use of even a small percentage of the gradients kept on cloud servers to retrieve local data was noted by other studies. Due to the increased communication costs between the learning participants and the cloud server, these researchers adopted the additive HE method to ensure privacy against an honest but curious parameter server. [1]

VI. Conclusion

Deep learning is becoming a common practice, and whenever new technology is used, security and privacy concerns undoubtedly arise. There has been a lot of study done recently on the training and interface modules for deep learning and deep neural networks that preserve security and privacy. Security and privacy considerations so become extremely crucial and significant, just like with other technologies, and cannot be disregarded. [3]
The security and privacy frameworks that were employed in this paper have several features and cryptographic primitives. It should be emphasized that private interference frameworks do not fully support the security and privacy needs of DNNs. We describe the specifics of many Deep Learning security attacks.

Model inversion, model extraction, and membership inference are just a few of the attacks used to take advantage of the Deep Learning findings and obtain model information or knowledge about the training data. The aforementioned attacks compile training data and produce desired outcomes. Compared to the interface, Deep Learning's private training section has a higher computational overhead. In order to create a more effective solution for the data privacy preservation while preserving models, more focus and study are needed in this regard. [3]

Due to numerous Deep Neural Network properties, which rely on a significant amount of input training data, privacy risks constantly exist. We also covered potential privacy risks to the private and sensitive data used in deep learning models in this chapter. Numerous experiments employing Deep Learning have been done on privacy-preserving assaults. [3]

It is crucial for future work that the researchers thoroughly examine various cryptographic primitives' solutions for DNNs. A mixed protocol approach can lessen the computational burden on systems that provide privacy and security. It's also a fascinating and open research topic to create a workable method to customize the privacy and security protocols for DNNs. Additionally, the authors plan to conduct research on the use of deep learning, particularly in the fields of astrophysics, plasma physics, atomic physics, thermodynamics, electromagnetics, machines, nanotechnology, fluid mechanics, electro hydrodynamics, signal processing, power, energy, bioinformatics, economy, and finance. [3]

REFERENCES

[1] Ximeng Liu; Lehui Xie; Yaopeng Wang; Jian Zou; Jinbo Xiong; Zuobin Ying; Athanasios V. Vasilakos, Privacy and Security Issues in Deep Learning: A Survey, 15th December 2020

[2] Muhammad Imran Tariq ,Nisar Ahmed Memon, Shakeel Ahmed, Shahza di Tayyaba, Muhammad Tahir Mushtaq, Natash Ali Mian,5Muhammad Imran, and Muhammad W. Ashraf; A Review of Deep Learning Security and Privacy Defensive Techniques; 07 Apr 2020.

[3] Georgios A. Kaissis, Marcus R. Makowski, Daniel Rückert Rickmer F. Braren, Secure, privacy-preserving and federated machine learning in medical imaging; 08 june 2020.