Analyzing the Vulnerability of Machine Learning Models against Membership Inference Attacks

Chirag Daryani
cdaryani@ualberta.ca

Karan Chadha
kchadha1@ualberta.ca

Abstract

With the rapid adoption of machine learning in real-world applications,we are witnessing an equal amount of concern regarding the confidentiality of user information that these models utilize for their training. More and more organizations are using machine learning as a service platforms for creating machine learning models, but these platforms do not provide any transparency about whether they are using any privacy mechanism in their model training process. The models built on these platforms are found to be prone to multiple types of attacks. One such attack is the membership inference attack where an adversary determines whether a particular user's data was used as the training data for the model. These attacks can have serious consequences with respect to the privacy of the end-user. In this regard, our current work focuses on the implementation of such attacks on our own ML models. We demonstrate that our attack achieves a considerably high level of effectiveness. We also perform an analysis of what factors affect the vulnerability of models to leak information about their data.

1 Introduction

In this work, we discuss and analyze how machine learning models are prone to leaking private data which were fed to them during their training. Our work primarily revolves around the membership inference attacks in a black-box setting. It means that the adversary only has access to the model API which can be used to query the models and get the predictions on the particular data record. The adversary has no information about the model structure, parameters, algorithm, and the data used for training that particular machine learning model. With only this limited information, the attacker has to determine whether the given data record was a part of the training dataset for the model or not. If the adversary is able to correctly predict this membership, we say the attack is successful.

In general, an ideal machine learning model is one that not only is able to correctly predict something on the data on which it was trained but also performs well on the unseen data. Such a generalized model can be accomplished when appropriate hyperparameter tuning is performed along with sufficient training of the model on the inputs. However, it is very common to observe that the model performs well on data it had seen but the confidence level of its predictions on unseen data is comparatively less. Membership inference attacks exploit this property that the model has a different output behavior on training samples and the samples that it did encounter during training.

Towards the goal of exposing this vulnerability of machine learning models, the focus of our current work is implementing these membership inference attacks first described in the paper by Shokri et al. [2017]. We will be implementing these attacks on a custom-designed neural network model built on a real-world image dataset that was not used in the paper. We will then do an analysis on what factors affect this vulnerability of machine learning models, and how these attacks can be mitigated. We also correlate our observations with the claims presented by the authors of the above paper.

2 Motivation

In today's world, machine learning services have become very popular. These can be standalone applications or can be a small module of an existing web or mobile application. There is a rapid invention of use cases where machine learning can be applied, providing better results and saving large amounts of manual effort for the companies or end-users. Due to this, more and more companies are trying to integrate modules of ML into their applications. These services can be recommendation systems where a particular product is recommended to the users by analyzing their preferences, or it could be some image or speech recognition service. It can also be used for analyzing the health conditions of an individual and in predicting the onset of future diseases or prescribing treatment. But not all small companies have the resources to build complete ML systems from scratch. These companies then have to rely on Machine learning as a service (MLaaS) providers like Amazon's AWS, Microsoft Azure, Google Cloud Machine Learning Engine, BigML, etc. to get the ML services for their applications. The end users only have to upload their data to these platforms and then these providers perform the training of models on this data and provide a simple black-box API that can be used to query the models and get the predictions. These platforms present no control to the end-user and don't provide any transparency about the privacy mechanisms that were used while building these models. It is this training of models done by these service providers that makes the models prone to privacy attacks. One such attack is the membership inference attack.

Membership Inference Attacks are carried out on a target machine learning model by an adversary and the main goal is to ascertain whether a given record was a part of the training dataset used to train that model. Such membership inference attacks can have serious implications related to the breach of confidential user information. For instance, if the attacker is able to determine the fact that a particular person's medical record was used to train a model that prescribes treatment for a disease, then the attacker is able to make the conclusion that the person is probably suffering from that same disease. Even with only limited query access to the target model, the attacker can infer a relationship between the input data and the output predictions of the model. In real-world situations, the attacker might also be aware of the background information of the population whose data was used as a part of the training dataset. They might also know the statistical information about the features of the dataset. There can also be situations where the attacker has a dataset very similar to the dataset used in training the model by the service provider, which would make the application of these attacks even broader. These possibilities aggravate the privacy concerns related to the leak of personal data using such membership attacks. Hence, we believe this work is very relevant in today's world where preserving privacy is an integral aspect of consideration for corporations that design artificial intelligence-based products and for consumers whose information is at risk. This is our main motivation to pursue this project, where we try to implement these attacks from scratch and perform a thorough analysis on the same.

3 Related Work

We reviewed several papers that would aid us in the implementation of the project. The paper "Membership Inference Attacks Against Machine Learning Models" by Shokri et al. [2017], was the first work that introduced and raised the concern of membership inference attacks on machine learning models. In their work, they proposed the novel technique of Shadow Model Training in order to carry out these attacks against the models. This is the main focus of our project as we try to primarily demonstrate the implementation of membership attacks through this technique implemented in this paper. The paper "ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models" by Salem et al. [2018], relaxes some of the assumptions of these attacks and discusses that these attacks have a broader application than expected. Their results depict that membership inference attacks can be slightly modified and performed in a simpler and more efficient manner, which implies the severity of risk that is posed to the ML models. They have also suggested defense mechanisms like model stacking, dropout against the membership inference attacks. Yeom et al. [2018] in their paper, "Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting" (2018) discusses in detail the effect of overfitting and the influence on membership inference attacks. Through their study, they concluded that overfitting was sufficient to allow an attacker to perform membership inference. We are utilizing some aspects of these techniques as we try to analyze the effect of overfitting, and regularization techniques on the performance of the attacks we implementing.

4 Methodology

The main intuition behind the membership inference attacks is that given only access to a data record and the prediction for that record from the model, if the attacker is able to correctly determine whether the particular data record was a part of the training set, then we consider the attack to be successful. If not, we say the attack failed. For the implementation of this attack, we are following the Shadow Model Training technique proposed by Shokri et al. [2017]. The basic setup we have for implementing this attack is as follows. The model that we want to attack is called the Target Model. It can be any standard machine learning-based model. Our ultimate goal is to train another machine learning model called the Attack Model whose main purpose would be to classify whether a particular training data record was used in the training set of the target model or not. So essentially we are turning machine learning against itself and training the attack model in such a way it learns to recognize the differences in the target model's output on the inputs the target model had seen during the training and the inputs it did not see. And to train the attack model without knowing anything about the structure or parameters of the target model, we need to create multiple Shadow Models that will help the attack model learn this behavior of the target model. Now let's discuss the basic experimental setup we have for implementation of this Shadow Model Training technique.

4.1 Dataset

For creating the target model on which we will perform the membership attack, we are using the Fashion-MNIST Image dataset provided by Xiao et al. [2017]. It is a dataset consisting of a training set of 60,000 images and a test set of 10,000 examples. Each image in the dataset is a 28x28 grayscale image, associated with a label ranging from 0 to 9, totaling 10 classes. We selected this dataset for creating our target model due to a couple of reasons. First, this dataset is a more challenging replacement for the MNIST Image dataset that is used in the paper we are basing our work on. By using a different image dataset, we can compare our observations with the results presented in that paper. Since the dataset has 10 output labels, the target model we build will be a multi-class classification model. Also, since the number of classes is more, we can sufficiently vary the number of classes and analyze the effect of changing the number of output classes on the attack accuracy. Since we have a sufficiently large number of data records, we can vary the number of data points, and analyze the effect of training data size on Attack Accuracy. Also, an attack on this dataset will expose the vulnerability of models to leak information about personal image data of users which can be detrimental in the real world.

4.2 Target Model

As described above, the target model can be any standard machine learning-based model. Since we are working on image classification in a multi-class setting, we decided to use a simple neural network model with a softmax activation function in the last layer to give us probabilities for each of the 10 classes. We performed some preprocessing of the image data like the normalization of pixels, one-hot encoding the target labels, and flattening the image vectors into appropriate dimensions before feeding them into the neural network. We then trained the neural network model on the training set. Next, we move on to creating the shadow models.

4.3 Shadow Models

For creating multiple shadow models that would each imitate the target model, we must ensure that the training of each of these individual models happens in a similar fashion as the target model. So if we are attacking a target model built on one particular cloud platform, we would have to utilize the same cloud platform for the training of each of the shadow models. In our experimental setup, because we are attacking the neural network-based target model, we, therefore, use the same neural network architecture to build each of the shadow models. The dataset for each of the shadow models is sampled from the same dataset space as the target model, but we are ensuring that there is no overlap between the datasets of the target model and each of the shadow models. Each dataset used for training the shadow model is disjoint from the training dataset of the target model. We also ensure that the dataset for each shadow model is different from the dataset of other shadow models. We also experimented with various numbers of shadow models increasing the number from 2 to 9 and then did the analysis of our final attack accuracy which will be discussed in the next section.

Now, because we know which record we used in the training for each of these shadow models, and which record we kept aside for testing the shadow models, we, therefore, know the ground truth about membership in these datasets of the shadow models. So, we can generate the labeled outputs for each prediction we get from these trained shadow models. Each prediction would be either given the label "in" (1) which means the record was present in the training dataset of the shadow model or we could give it a label "out" (0) which means the record was present in the test dataset of the shadow model. Next, we would train our attack model on these labeled outputs of the shadow models.

4.4 Attack Model

For the attack model, we have a dataset of records, the corresponding outputs of each shadow model on these records, and the in/out labels which represent the membership of these records for the shadow models. Now we will train the attack model on these labeled outputs and it will learn how to distinguish between outputs that are for member data records and outputs which are for non-member data records. Therefore, the attack model would be a binary classifier with two output classes "in" (1) and "out" (0).

We tried to experiment with different model architectures for the attack model. We tried a logistic regression model, a decision tree model, a linear SVM model, and also a Radial basis function kernel-based SVM model. The comparison of the attack accuracy for each of the model architectures would be discussed in the next section.

In summary, we train the attack model on synthetic data generated by the shadow models for which we know the membership status for each data record, and in this way, we teach the attack model how to differentiate between the members and the non-members.

4.5 Evaluation Metrics

After training the attack model, we now perform the membership inference attack on the target model. We query the target model with a data record, obtain the prediction and pass it to the attack model. If the attack model is able to correctly determine the presence of the given data in the training set of the target model, the attack is deemed successful else we consider it as a wrong prediction. Since the membership inference problem is converted to a binary classification task, we would be using the standard metrics of classification as our evaluation metrics for measuring the attack performance.

First would be the attack accuracy which represents the proportion of the total number of predictions that were correct. Next would be the precision which represents what fraction of records predicted as members (positive case) are indeed members of the training dataset for the target model. Lastly, we would look at recall which would be the fraction of the target model's training dataset's members that are correctly predicted as members by our attack model.

5 Results

In this section, we present the results of our membership inference attacks on the target neural network model trained on the Fashion MNIST dataset. The target model classifies each of the images into one of the 10 categories of clothing. We will present the overall attack accuracy, precision, and recall along with our analysis for various factors that affect the membership attack performance.

5.1 Analysis of Best Performing Attack Model Architecture

We tried to experiment with different model architectures for the attack model. We tried a logistic regression model, a decision tree model, a linear SVM model, and also a Radial basis function kernel-based SVM model as our attack model that would train on top of the shadow models' predictions. We varied the number of shadow models from 4 to 7 and noted the attack accuracy for each combination of attack model and shadow model count. Here are the results we got for all the attack models.

We observed that while both the RBF kernel-based SVM and the Linear SVM gave comparable performance, **the maximum attack accuracy was achieved when we used 6 shadow models in combination with the Linear SVM as the attack model**. The final attack accuracy, precision, and recall for the best performing membership attack we could achieve are shown in the table below.

Attack Model	No of Shadow Models	Attack Accuracy
Logistic Regression	4	0.6451
Logistic Regression	5	0.6021
Logistic Regression	6	0.6102
Logistic Regression	7	0.6451
Decision Tree	4	0.6397
Decision Tree	5	0.6182
Decision Tree	6	0.5860
Decision Tree	7	0.6370
RBF Kernel SVM	4	0.6263
RBF Kernel SVM	5	0.6102
RBF Kernel SVM	6	0.6720
RBF Kernel SVM	7	0.6290
Linear SVM	4	0.6021
Linear SVM	5	0.5913
Linear SVM	6	0.6962
Linear SVM	7	0.6129

Table 1: Accuracy for each Attack Model Architecture

Performance Metric	Value
Accuracy	0.6962
Precision	0.6237
Recall	0.9892

Table 2: Performance Metrics for the Final Attack

Slightly lower precision of our Attack indicates that there are some false positives which means that some non-members are incorrectly predicted as members by our attack model. Despite this, overall attack accuracy and precision are decent. We observe a high recall for our attack which means we are having very few false negatives. It means members are not incorrectly predicted as non-members by our attack model. We believe this metric is important for our task of inferring membership inference when we consider privacy with respect to a particular user whose information is at risk of leakage.

5.2 Effect of changing the Number of Shadow Models

The graph below shows how the accuracy of our attack model varies as we change the number of shadow models used for training the attack model. For this experiment, we used the Linear SVM model as the attack model that would give the final membership prediction. We took the training size for each shadow model as 3000 records and the test size as 2000 records. We then varied the number of shadow models used for training the attack model keeping all other things the same and then noted the attack accuracy in each case.

We found that **overall, increasing the number of shadow models results in some increase in attack accuracy** but the trend is not consistent for each successive value.

The explanation behind this trend of increasing attack accuracy by using more shadow models can be explained as follows. We train the attack model to learn to recognize the differences in the target model's behavior on member inputs and non-member inputs. The attack model trains on the labeled outputs of the shadow models which imitate the target model. So more the number of shadow models, the more the opportunity for the attack model to learn about this behavior of the target model. Hence, the predictive power of the attack model would increase, and thus we see improvement in the attack accuracy.

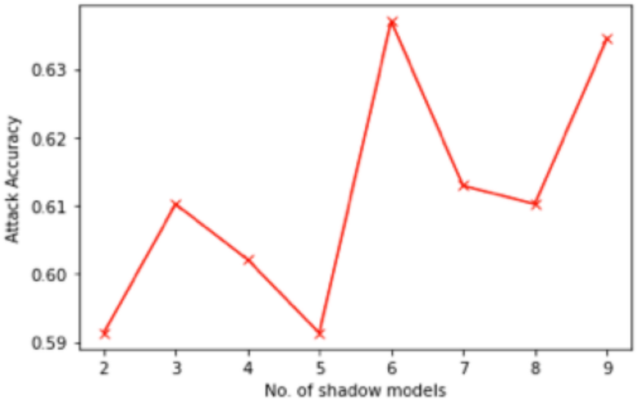

Figure 1: Effect of changing the Number of Shadow Models

5.3 Effect of changing the training size for the target model

To analyze the effect of the target model training data size on attack accuracy, we try two sets of training data. First, we set the training data as 3000 records for the target model and then observe the attack accuracy for each combination of shadow model count. We then increase the training data to 6000 records and then take similar observations.

No of Shadow Models	Accuracy with 3000 samples Training Data	Accuracy with 6000 samples Training Data
4	0.6021	0.5054
5	0.5913	0.5869
6	0.6370	0.6331
7	0.6129	0.6114
8	0.6102	0.5054

Table 3: Attack Accuracy with change in Training data size

We observe that the more training data is used for the target model, the lesser is our attack accuracy. *We were expecting this trend because by using more training samples for the target model, we are making the target model more robust. The target model would generalize better and hence would leak less information about the inputs on which it was trained. Hence, attack accuracy decreases.*

5.4 Effect of changing the Number of Classes

The graph below shows how the accuracy of our attack model varies as we change the number of classes in the target model. First, we performed our experiment with the original number of 10 output classes for the target model. We changed the number of shadow models and noted the attack accuracy for each combination, keeping all other things constant. We observed that the attack accuracy when attacking a target model with 10 output classes ranged from 0.5913 to 0.6370. We then reduced the number of classes from 10 to 5 and noted the attack accuracy for each combination of shadow model numbers. **We observed that the attack accuracy reduced considerably when we were attacking the target model with only 5 output classes.** The attack accuracy now ranged from 0.5160 to 0.5506.

One possible explanation why we observe this trend is that the lesser the number of output classes in the target model, the lesser the target model needs to remember about their training data, and hence lesser will be the amount of information that the model will leak. In other words, lesser the number of classes, then lesser information about the internal parameters that are learned by the target model

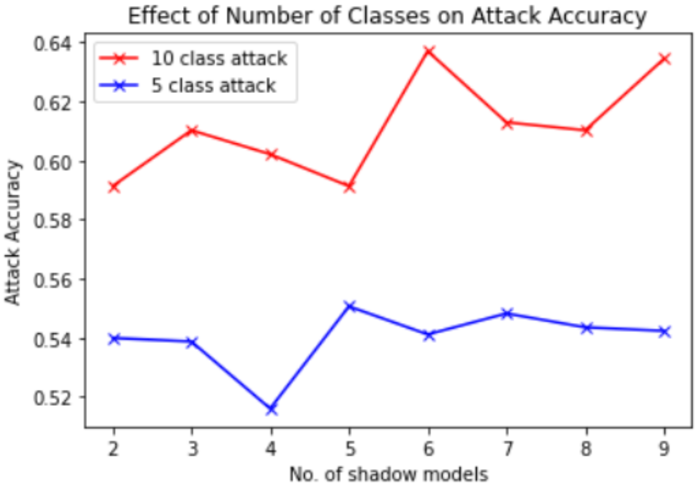

Figure 2: Effect of changing the Number of Classes

would be available for the attack model to exploit and base its decision about membership on. Hence, the attack accuracy would decrease.

5.5 Analysis of Overfitting and Dropout

Next, we move to the analysis of how overfitting can affect attack accuracy and what effect does the mitigation techniques such as dropout has on the attack performance. So for this experiment, we replace our simple neural network-based target model, with a more denser network. We add more number of layers and more neurons per layer to introduce some amount of overfitting in the target model. We then measure the performance of our attack using the linear SVM attack model. We vary the number of shadow models keeping everything else constant. **We observe that attack accuracy has increased considerably for each value of shadow model count. This means overfitting has an effect on the performance of membership attacks. The more overfitted a model, the more it leaks.**

No of Shadow Models	Attack Accuracy on Dense Network	Attack Accuracy with Dropout
2	0.6370	0.5725
3	0.6370	0.5672
4	0.6075	0.5779
5	0.6344	0.5000
6	0.6263	0.5967
7	0.6370	0.5698
8	0.6263	0.6102
9	0.6317	0.5860

Table 4: Attack Accuracy with change in Training data size

Now we investigate how to mitigate this vulnerability of models to leak information because of overfitting. One way we can prevent overfitting is by decreasing the number of learnable parameters in the network. For example, we can reduce the number of layers. But this may not be feasible for all models. So we tried another technique which is regularization. For this, we introduced dropout in some layers of the network. We fix a dropout rate of 0.5 which means 50% of neurons of that layer will be deactivated. We now measure the attack accuracy for this regularized target model in the same setting. **We observe that after dropout is introduced, attack accuracy declines considerably for each attack we perform.** *The explanation behind this is that because dropout is preventing overfitting of the target model, the model leaks less information, and hence attack accuracy decreases.*

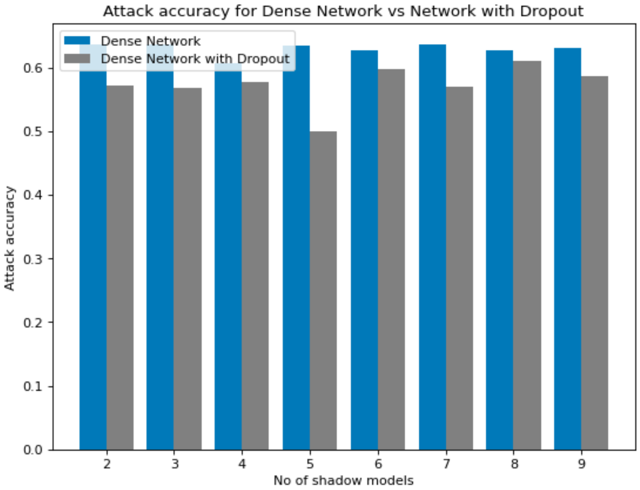

Figure 3: Attack Accuracy for Dense Network vs Network with Dropout

5.6 Significance of Results

Through the results presented in this section, it is evident that machine learning models are prone to leak a lot of information about the records that are used in their training. These results are important as they indicate that the privacy of users is at risk when their data is used to build machine learning models through the cloud service providers. Even with limited information about the training dataset, we could reach an attack accuracy of almost 70%. In the real world, the adversary may have access to some records of the actual dataset used to build the models. In that case, the attack can be more severe. We were able to verify most of the observations presented in the paper by Shokri et al. [2017] through our implementation on a different dataset. This means that these attacks are more generalizable and this adds to the severity of the risks associated with them. Therefore, we believe these results are significant towards our objective of exposing the vulnerability of machine learning models.

6 Conclusion

In summary, we worked on the implementation of the membership inference attack on a custom-designed neural network model built on a real-world image dataset. We performed a detailed analysis of what factors contribute to the performance of these attacks and we also presented how some techniques like regularization can make our model leak less information. Through our results, it is clear that these attacks are more broadly applicable in the real world. Now the companies that provide model services on user data must also account for the risk that their model will leak the user information it is trained on. They must also provide more transparency about what privacy mechanisms they are using to protect confidential user information. These attacks can be used as metrics to measure the effectiveness of privacy conservation techniques that are employed in the future and to select the best privacy-preserving model.

With respect to limitations, we assumed that each shadow model is trained in a similar way to the target model. But what if the attacker does not have access to the same cloud platform where the target model is built? In such a case, the scope of implementing these attacks gets reduced. So this is a limitation of our current implementation. Some areas of future work for this project include the incorporation of differential privacy mechanisms into these models and then performing membership attacks to analyze whether the models get some protection from such attacks. We could also do a more detailed analysis about how such attacks can be mitigated apart from applying regularization. Another scope of improvement can be to modify the shadow model training technique in such a way that even with a lesser number of shadow models, we are able to achieve similar results. That would make these attacks more efficient.

References

[1] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*, 2018.

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

[3] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[4] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282. IEEE, 2018.