Security and Privacy Attacks on Machine Learning Algorithms

A Sangeetha , P Ruchitha Reddy, Ananya Puppala
Assistant Professor, Student, Student
Department of Computer Science and Engineering
CBIT, Hyderabad, Telangana, India

Abstract- Machine Learning has gained a significant increase in popularity in the recent times. ML models are being used in almost every other field including medicine, finance and many more. As machine learning has increasingly been deployed in critical real-world applications, the dangers of manipulation and misuse of these models has become of paramount importance to public safety and user privacy. In applications such as online content recognition to financial analytics to autonomous vehicles all have shown to be vulnerable to adversaries wishing to manipulate the models or mislead models to their malicious ends. Technical community's understanding of the nature and extent of these vulnerabilities remains limited even though there has been a growth in recognition that ML exposes new vulnerabilities in software systems. Identifying various types of privacy and security attacks possible on ML models and demonstrating those attacks is the focus of the project. For security part adversarial attacks on Machine Learning models will be introduced and privacy part model inversion attack and membership inference attack will be performed to show that ML models leak information.

Keywords – Security and Privacy, Machine Learning, Security Attacks, Privacy Attacks

I. INTRODUCTION

Security and Privacy in Machine Learning has been in research and continues to be as Machine Learning models are vulnerable to several types of attacks. Machine Learning is the electricity and foundation of modern technologies and plays significant role in growing web-based services because of its wide applications. It is provided as service by Amazon, Google, Microsoft and many more. These companies provide services like training API, where the user can upload data to the cloud and train the model for example, a classification model. Later, user can use these models using prediction API's and do prediction. Prediction output is vector of probabilities that assign probability to each class to classify the object.

Example in the CIFAR-dataset, it takes a picture of a Car and assigns probability to the classes to predict whether it is a car, truck, airplane, submarine, etcetera. These training API's are good examples of black box models, where the training model stays on the cloud and the user has no information about the architecture or parameters of the model, just can get the prediction vector. The prediction outputs have no information of the model nor information on predictions of the intermediate steps. Such black box models are very useful. Many mobile application developers use such services to predict the responses of the new features. There is no access to the training datasets of the training model, so when the prediction is made, there is no interaction with the dataset of the machine learning model, only the prediction vector is given as the output.

The understanding of the threats, attacks and defences of systems built on ML is fragmented across several research communities including ML, security, statistics, and theory of computation. This motivates and challenges people to put effort to systematize knowledge about the myriad of security and privacy issues that involve ML.Most of the Machine Learning models which are in use right now are not completely robust and they do not preserve the privacy of users. Big companies have invested a lot in this area and are trying to improve the models. There have been significant improvements like amazon recognition was initially was not immune to adversarial attacks, but now they have improved their robustness and are immune to these attack.

Various security and privacy attacks on machine learning models have explored the attack surface of systems built upon ML. They have painted a picture about the vulnerabilities of ML and the theoretical countermeasures used to defend against. Defences for all the attacks are yet unknown, yet a science for understanding them is slowly emerging[1]. Recent findings on adversarial examples for deep neural networks have also summarized few methods for generating these examples. They further discussed about countermeasures for adversarial examples and explored the challenges and the potential solutions[2].

Model Inversion Attacks that Exploit Confidence Information and Basic Counter measures. Model inversion a privacy attack in which the attack gets adversarial access to an ML model and abuses it by learning sensitive genomic information about individuals. It showed how to recover recognizable images of people's faces given only their name and access to the ML model. Inspecting facial recognition APIs, it turns out that it is common to give floating-point confidence measures along with the class label (person's name). This enables us to craft attacks that cast the inversion task as an optimization problem. They found the input that maximizes the returned confidence, subject to the classification also matching the target[3].

Membership Inference Attacks Against Machine Learning Models focused on how machine learning models leak information about the individual data records on which they were trained. Their focus was mainly on basic membership inference attack, determining whether a data record or a sample is present in the model's training dataset. To perform membership inference against a target model, a shadow model and attack model were trained to recognize differences in the target model's predictions on the inputs that it trained on versus the inputs that it did not train on[4].

The rest of the paper is organized as follows. Proposed algorithms are explained in section II. Experimental results are presented in section III. Concluding remarks are given in section IV.

II. RESEARCH METHODOLOGY

2.1 Security and Privacy Attack Model

The idea of this attack is to fool the machine learning model by feeding it malicious input which looks very similar to original input but that cause the model to make false predictions or categorizations. Adversarial examples are very real and therefore need to be planned for in the machine learning security plan.

The term "adversary" is used in the field of computer security to describe people or machines that may attempt to penetrate or corrupt a computer network or program. Adversaries can use a variety of attack methods to disrupt a machine learning model, either during the training phase which is called a "poisoning" attack or after the classifier has already been trained which is called an "evasion" attack.

The fast gradient sign method is a evasion attack which works by using the gradients of the neural network to create an adversarial example. For an input image, the method uses the gradients of the loss with respect to the input image to create a new image that maximizes the loss. This new image is called the adversarial image. The gradients are taken with respect to the input image. This is done because the objective is to create an image that maximises the loss. A method to accomplish this is to find how much each pixel in the image contributes to the loss value, and add a perturbation accordingly. This works pretty fast because it is easy to find how each input pixel contributes to the loss by using the chain rule and finding the required gradients. Hence, the gradients are taken with respect to the image. In addition, since the model is no longer being trained (thus the gradient is not taken with respect to the trainable variables, i.e., the model parameters), and so the model parameters remain constant. The only goal is to fool an already trained model.

When the target model M is given an image I of class X, the targeted attack wants the model M to misclassify it as class Y, the target class. Whereas, the untargeted attack does not have any target class which it wants the model to misclassify the image as. Instead, the goal is simply to make the target model misclassify by predicting the adversarial example, I, as a class, other than the actual class X.

In privacy attacks, the goal of the adversary is to gain knowledge which is private to the users and developers, such as knowledge about the training data or information about the model, or even extracting information about properties of the data. The attacks where the adversary does not know the model parameters, architecture or training data are black-box attacks. In recent times, personal data is been leveraged by internet companies to train their machine learning models that power machine learning-based applications. These models should not reveal information about the data used for their training as training data may include private information. In white-box attack, the adversary has either complete access to the target model parameters or their loss gradients during training.

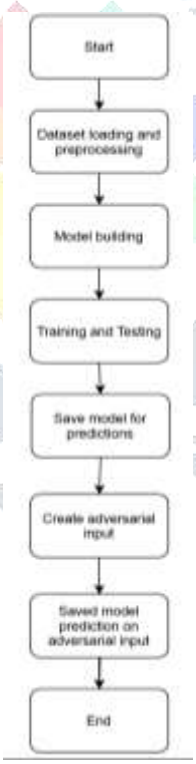

Figure 1. Security Attack Model

The security attacks used are fast gradient sign method, basic iterative method and targeted fast gradient sign method.

2.1.1 Fast gradient sign method

The fast gradient sign method works by using the gradients of the neural network to create an adversarial example. A new image that maximises the loss is created by using gradients of the loss with respect to the input image. This new image is called the adversarial image. This is summarised using the following expression:

$adv\_x = x + \epsilon * sign(\nabla x J(\theta, x, y))$

where

adv_x : Adversarial image.

x : Original input image.

y : Original input label.

ϵ : Multiplier to ensure the perturbations are small.
θ : Model parameters.
J : Loss.

2.1.2 Basic Iterative Method-

FGSM was improved to create a more powerful attack called Basic Iterative Method. It suggested applying the same step as FGSM multiple times with a small step size and clip the pixel values of intermediate results after each step to ensure that they are in an ε-neighbourhood of the original image. Mathematically, the attacking scheme can be demonstrated as:

$$\boldsymbol{X}_0^{adv} = \boldsymbol{X}, \quad \boldsymbol{X}_{N+1}^{adv} = Clip_{X,\epsilon}\left\{\boldsymbol{X}_N^{adv} + \alpha\,\mathrm{sign}\left(\nabla_X J(\boldsymbol{X}_N^{adv}, y_{true})\right)\right\} \tag{1}$$

In essence, FGSM is to add the noise whose direction is the same as the gradient of the cost function with respect to the data. The noise is scaled by epsilon, which is usually constrained to be a small number via max norm. The magnitude of gradient does not matter in this formula, but the direction (+/-).

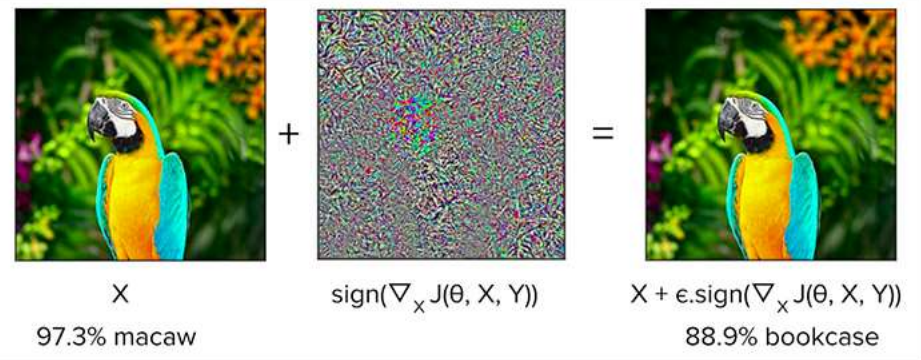

Figure 2 : The Fast Gradient Sign Method (FGSM) for adversarial image generation

2.1.3 Targeted fast gradient sign method (T-FGSM)

Targeted fast gradient sign method is similar to the FGSM, in this method a gradient step is computed in the direction of the negative gradient with respect to the target class. The loss function is calculated with respect to target class instead of the original class. This method reduces the loss of the target class so that the model's confidence of the target class increases.

adv_x=x - ϵ∗sign(∇xJ(θ,x,y))
where
adv_x : Adversarial image.
x : Original input image.
y : Target class label.
ϵ : Multiplier to ensure the perturbations are small.
θ : Model parameters.
J : Loss.

2.1.4 Privacy Attack

Consider a dataset of personal data $\mathbf{A} = a_{i,j} \in \mathbb{R}^{m \times n}$ as shown in the Figure 4.3.2, where each $[a_{1,\star}, a_{2,\star} \ldots a_{m,\star}]$ within is a row of personal characteristics relating to one of the n data subjects in the set $|DS_1| = n$, with each of the m variables indexed by j. Consider, they also have access to a model M(B), which is an ML model trained on personal data $\mathbf{B} = b_{i,j} \in \mathbb{R}^{x \times y}$, where each $[b_{1,\star}, b_{2,\star} \ldots b_{x,\star}]$ within is a row of personal characteristics relating to one of x data subjects in the set $|DS_2| = y$, and each one of the x variables is a feature in the trained model.. Also, $DS_1 \cap DS_2 > 0$: that is, some individuals are in both the training set and the additional dataset held.

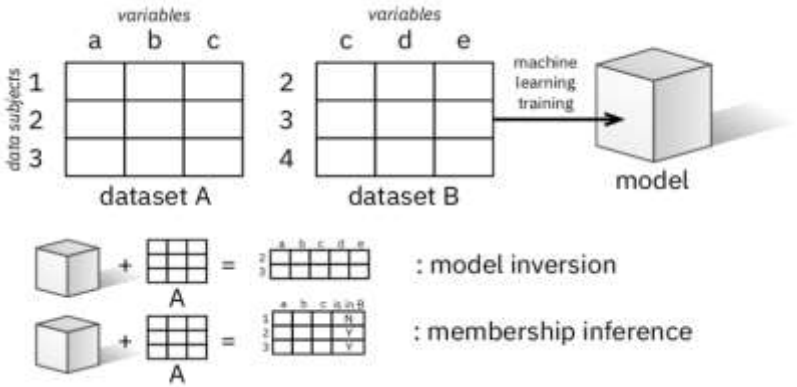

Figure 3: Model inversion and membership inference attacks.

2.2 Model Inversion Attack

Under a model inversion attack, a data controller who does not initially have direct access to B but is given access to A and M(B) is able to recover some of the variables in training set B, for those individuals in both the training set and the extra dataset A. These variables connect to each other, such that the new personal dataset in question has all the variables of A and some of B.

There may be error and in exactitude in the latter, but the data recovered from those in the training dataset will be more accurate than characteristics simply inferred from those that were not in the training dataset.

The basic idea of this attack was to input random noise through the model that was being attacked (target model) and backpropagate the loss from this random noise input but instead of changing the weights, the input image was changed. Here, instead of optimizing the weights to minimize the loss, the image was optimised to minimize the loss i.e., generating the image that the model thinks is the most likely sample of a class.

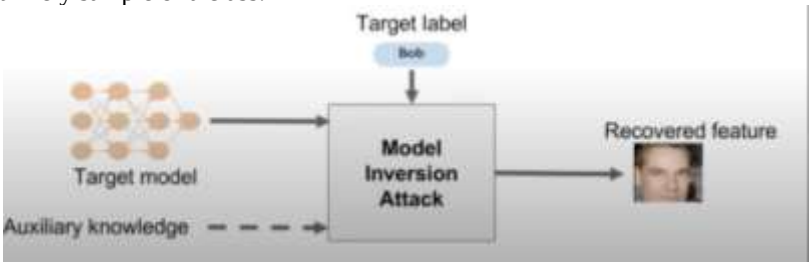

Figure 4: Model Inversion attack

1: **function** MI-FACE($label, \alpha, \beta, \gamma, \lambda$)
2: $c(\mathbf{x}) \stackrel{\text{def}}{=} 1 - \tilde{f}_{label}(\mathbf{x}) + \text{AUXTERM}(\mathbf{x})$
3: $\mathbf{x}_0 \leftarrow \mathbf{0}$
4: **for** $i \leftarrow 1 \ldots \alpha$ **do**
5: $\mathbf{x}_i \leftarrow \text{PROCESS}(\mathbf{x}_{i-1} - \lambda \cdot \nabla c(\mathbf{x}_{i-1}))$
6: **if** $c(\mathbf{x}_i) \geq \max(c(\mathbf{x}_{i-1}), \ldots, c(\mathbf{x}_{i-\beta}))$ **then**
7: **break**
8: **if** $c(\mathbf{x}_i) \leq \gamma$ **then**
9: **break**
10: **return** $[\arg\min_{\mathbf{x}_i}(c(\mathbf{x}_i)), \min_{\mathbf{x}_i}(c(\mathbf{x}_i))]$

Figure 5: Algorithm for inversion attack on face recognition model

Figure 4 shows model inversion attack and Figure 4.3.4 shows the algorithm for inversion attack on face recognition model.

2.3 Membership Inference Attack

Membership inference attacks do not recover training data, but instead ascertain whether a given individual's data were in a training set or not as shown in Figure 6 Under a membership inference attack, the holder of **A** and M(**A**) does not recover any of the columns in **B**, but can add an additional column to dataset **A** representing whether or not a member of DS_1 is in the set Z: that is, whether or not they were also part of the training set participants DS_2.

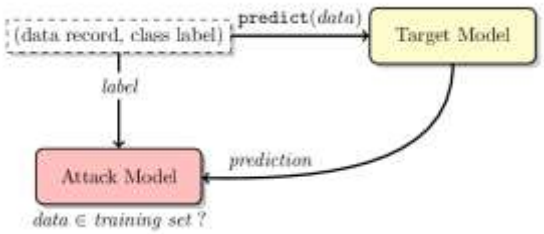

Figure 6: Membership inference attack in black-box setting

2.3.1 Membership Inference Attack Models

The models or networks used in membership inference attack, shown in Figure 7 are described here.

Table 1: Membership Inference attacks

Network Type	Purpose	Inputs	Outputs
Target Network	**Perform some classification task**	**Samples from a multi-class data distribution**	**Categorical distribution over class**
Shadow Network	**Produce 2 sets of classification probability vectors to be used in training the attack network**	**Samples from a multi-class data distribution**	**Categorical distribution over class labels (probability vector with length equal to the number of classes)**
Attack Network	**Perform membership inference by learning to classify probability vectors coming from the in-training set versus the out-of-training set**	**Probability vectors generated from either the in-training set or out-of-training set**	**Probability the input is a member of the in-training set**

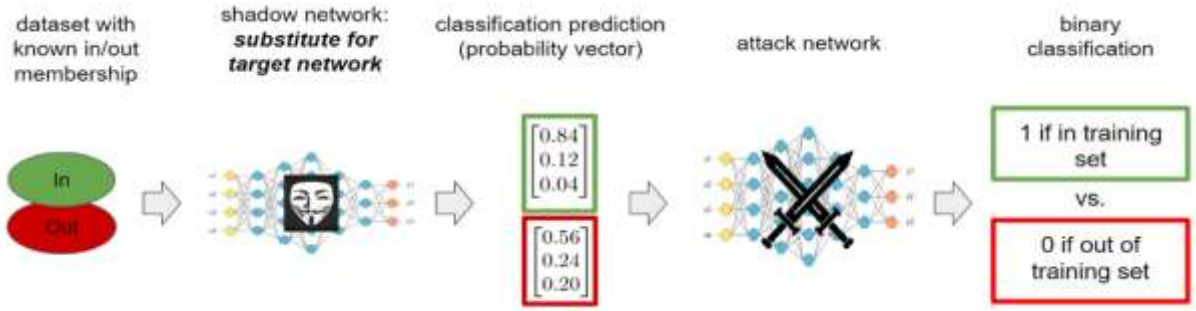

Figure 7: Training process for membership inference attack

The entire process of training and testing the membership inference attack is summarized in the following steps:

1.Split the original dataset into 4 disjoint sets representing target in, target out, shadow in, and shadow out sets.

2.Train the shadow network using the shadow in set.

3.Train the attack network using the outputs of the shadow in and shadow out set when sent through the shadow network.

4.Train the target network using the target in set.

5.The attack network using the outputs of the target in and target out set when sent through the target network.

III. RESULT AND DISCUSSION

The results of all the attacks performed are shown in this chapter.

3.1 Security Attack

3.1.1 Fast Gradient Sign Method

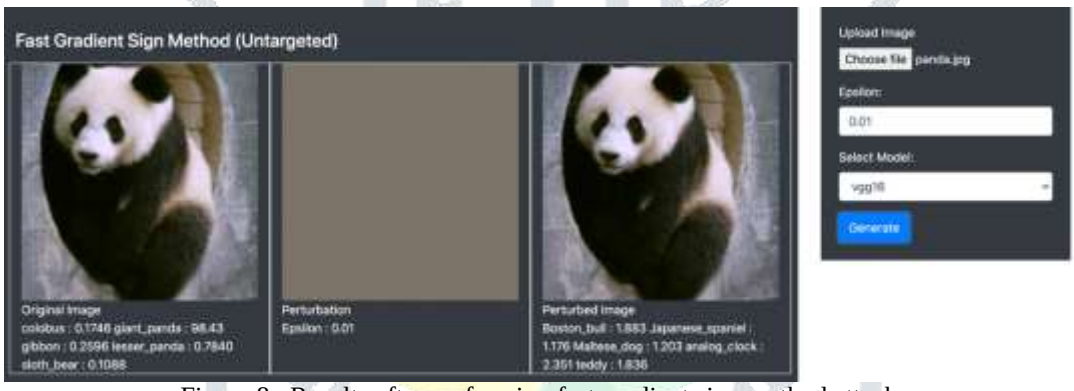

Figure 8. Results after performing fast gradient sign method attack

VGG16 ImageNet classification has confidence of 98.43% that the given image is of giant panda as shown in Figure 8. When perturbed image is given to the model, the model's confidence that the given image is of giant panda has become zero.

3.1.2 Basic Iterative Method

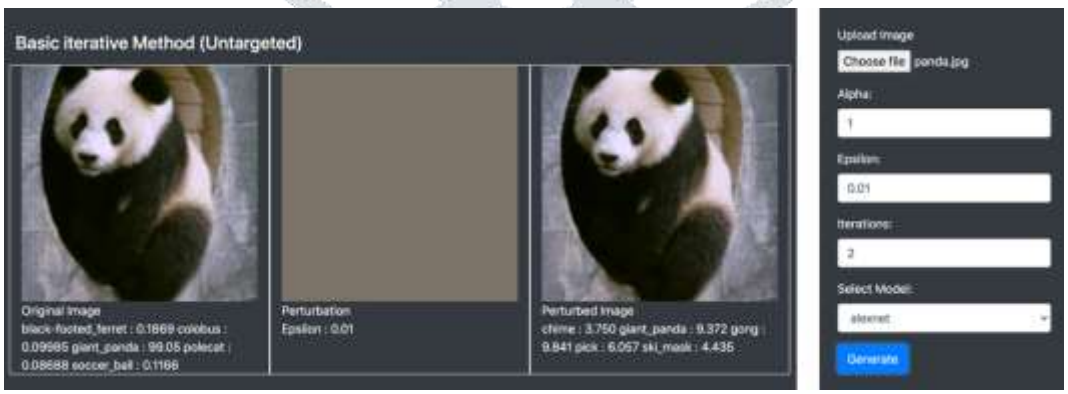

Figure 9 Results after performing basic iterative method attack

Alexnet ImageNet classification has confidence of 99.05% that the given image is of giant panda as shown in Figiure 9. When perturbed image is given to the model, the model's confidence that the given image is of giant panda has become 9.372.

3.1.3 Targeted Fast Gradient Sign Method

Alexnet ImageNet classification has confidence of 91.23% that the given image is of crane as shown in Figure 10 When this perturbated image is given to the model, the model's confidence that the given image is of crane has become 9.372% and that of school bus is 97.91%. The Target Class index of school bus in ImageNet dataset is 779.

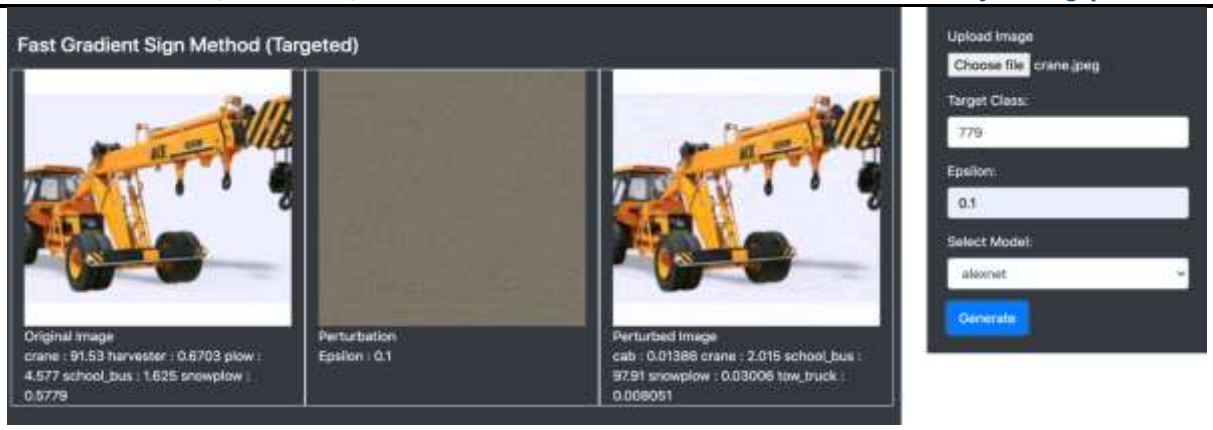

Figure 10 Results after performing targeted fast gradient sign method attack

3.2 Model Inversion Attack

It is a type of privacy attack which gets the private details of the dataset and training model. The dataset used for the training the face recognition model was AT & T faces. Model inversion attack reconstructs the faces which were used for training the model.

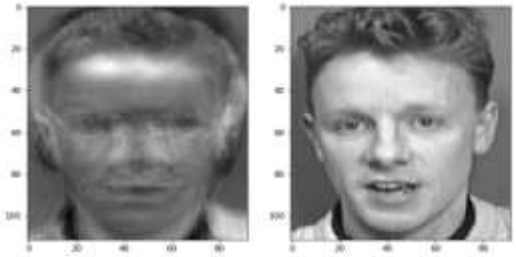

Figure 11 Result of model inversion attack (class label s21)

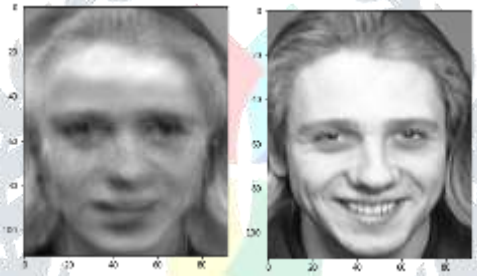

Figure 12 Result of model inversion attack (class label s12)

In the Figure 11 the left image is the reconstructed image of the person belonging to class s21. The right-side image is one of the original images of class s21 which was used while training the model. In Figure 12 the images belong to person belonging to class s12.

3.3 Membership Inference Attack

The dataset used to demonstrate this attack was CIFAR-10 dataset. This attack is used to know if a particular sample was present while training the model.

```
For shadow model 0
Training accuracy = 1.000000
Validation accuracy = 0.465200

For shadow model 1
Training accuracy = 1.000000
Validation accuracy = 0.470800

For shadow model 2
Training accuracy = 1.000000
Validation accuracy = 0.453200

For shadow model 3
Training accuracy = 1.000000
Validation accuracy = 0.467200

For shadow model 4
Training accuracy = 1.000000
Validation accuracy = 0.483600

For shadow model 5
Training accuracy = 1.000000
Validation accuracy = 0.460000

For shadow model 6
Training accuracy = 1.000000
Validation accuracy = 0.485600

For shadow model 7
Training accuracy = 1.000000
Validation accuracy = 0.492000

For shadow model 8
Training accuracy = 1.000000
Validation accuracy = 0.455600

For shadow model 9
Training accuracy = 1.000000
Validation accuracy = 0.471200
```

Figure 13: Accuracy of shadow models

10 shadow models were trained and the accuracy of each model is shown in Figure 13.

```
For ds =  2500
Attack Precision:  0.7708911501695961
Attack Recall:  1.0
Attack Accuracy:  0.8514

For ds =  10000
Attack Precision:  0.715000715000715
Attack Recall:  1.0
Attack Accuracy:  0.8007

For ds =  15000
Attack Precision:  0.6884523590967505
Attack Recall:  1.0
Attack Accuracy:  0.7737333333333334
```

Figure 14: Precision and accuracy of attack model

The precision and accuracy of attack model with different size of dataset that is 2500,1000 and 15000 is shown in the Figure 14

```
Out[17]: Text(0.5, 1.0, 'CIFAR-10, CNN , Membership Iinference Attack')
```

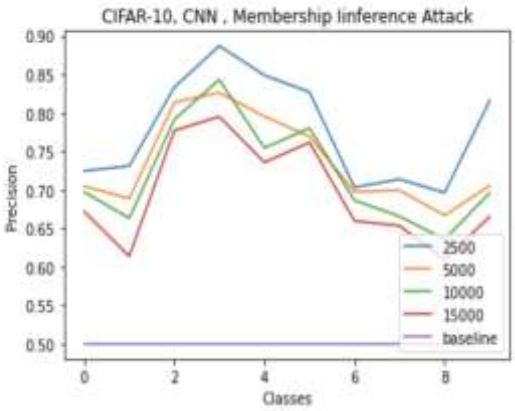

Figure 15 Graphical representation of precision of attack model with respect to each class in CIFAR dataset

The graphical representation precision of attack model with respect to each class in CIFAR-10 dataset is shown in the Figure 15 and the precision of attack model with respect to different training set sizes is shown in the Figure 15

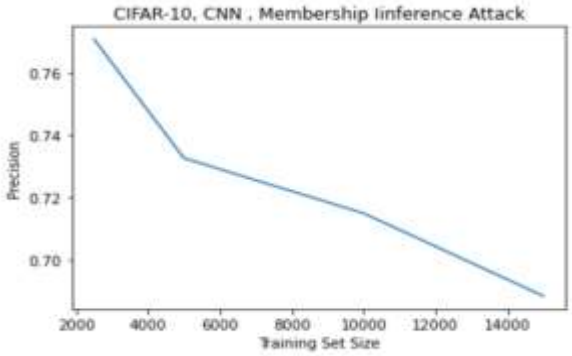

Figure 16 Graphical representation of precision of attack model with size of training dataset

```
For ds =  2500
Attack Precision:  0.7023004059539919
Attack Recall:  0.2076
Attack Accuracy:  0.5598
For ds =  5000
Attack Precision:  0.601123595505618
Attack Recall:  0.0642
Attack Accuracy:  0.5108

For ds =  15000
Attack Precision:  0.631578947368421
Attack Recall:  0.0008
Attack Accuracy:  0.5001666666666666
```

Figure 17 Precision and accuracy of attack model after applying regularization

The membership inference attack exploits the fact that model was overfitted to the training data. Therefore as a defense mechanism regularization was employed so that overfitting in the target model reduces. This made the target model less susceptible to these attacks. After applying L-2 regularization the accuracy of the attack model reduced. The attack model accuracy reduced from 77 % to 50% when the size of the dataset is 15000. The accuracy of attack model after applying regularization is shown in Figure 17. The graphical representation of accuracy of attack model with respect to each class in CIFAR dataset is shown in Figure 18.

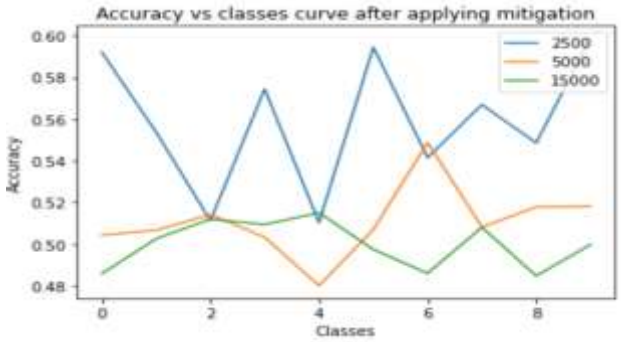

Figure 18 Graphical representation of accuracy of attack model with respect to each class in CIFAR dataset after applying regularization

IV.CONCLUSION

This paper throws light on the vulnerabilities that ML models have and introduce about the threats against these models. The following few things were implemented in this paper.

1. Adversarial examples were created and given to the ImageNet classification model to fool them. The ML models were tricked by the adversarial examples which resulted in change of original output and affected the confidence of model. A UI was created to demonstrate how the ImageNet model is being fooled by adding perturbation on the image.
2. For showing the privacy concerns on face recognition model which was trained on AT&T Database of faces model inversion attack was performed. Model inversion attack was able to reconstruct the faces of a class when the name of the person was given.
3. Membership inference attack on CIFAR-10 image-classifier was performed by creating an attack model which used the output from the shadow models. The attack was successful as the target model was prone to overfitting. As a defense mechanism regularization was performed which that helped to overcome overfitting and reduce attack model accuracy.

Security and Privacy in Machine learning is a huge field and a lot of researchers across the world are contributing their work. A small part of this field was considered to demonstrate the attacks possible on ML models. Implementing adversarial training for complex models like face recognition, Image classifiers is out of the scope due to computation power required to train. For privacy it will always be a trade-off between accuracy and preserving the data as regularization does not always maintain the accuracy of the model.

REFERENCES

[1] N. Papernot, P. McDaniel, A. Sinha and M. P. Wellman, "SoK: Security and Privacy in Machine Learning," 2018 IEEE European Symposium on Security and Privacy (EuroS&P), 2018, pp. 399-414.

[2] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy, “Explaining and harnessing adversarial examples”, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, USA, May, 2015.

[3] Alexey Kurakin, Ian J. Goodfellow, Samy Bengio, “ADVERSARIAL EXAMPLES IN THE PHYSICAL WORLD”, 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April, 2017.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, 2012, pp. 1097–1105.

[5] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 1322–1333. ACM, 2015.

[6] R. Shokri, M. Stronati, C. Song and V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models," *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 3-18.

[7] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart, “Stealing machine learning models via prediction apis”, In USENIX Security Symposium, 2016, pp. 601–618.