International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT22856

Membership Inference Attacks on Machine Learning Model

Preeti[1], Irfan Khan[2]
[1]Research(M.TECH) Scholar (CSE), Shekhawati Institute of Engineering and Technology, Sikar, Rajasthan, India
[2]Assistant Professor(CSE), Shekhawati Institute of Engineering and Technology, Sikar, Rajasthan, India

Article Info
Publication Issue :
Volume 8, Issue 5
September-October-2022

Page Number : 31-38

Article History
Accepted: 01 Sep 2022
Published: 09 Sep 2022

ABSTRACT

Machine learning(ML) models today are vulnerable to several types of attacks. In this work, we will study a category of attack known as membership inference attack and show how ML models are susceptible to leaking secure information under such attacks. Given a data record and a black box access to a ML model, we present a framework to deduce whether the data record was part of the model's training dataset or not. We achieve this objective by creating an attack ML model which learns to differentiate the target model's predictions on its training data from target model's predictions on data not part of its training data. In other words, we solve this membership inference problem by converting it into a binary classification problem. We also study mitigation strategies to defend the ML models against the attacks discussed in this work. In this paper evaluation method on real world datasets: (1) CIFAR-10 and (2) UCI Adult (Census Income) using classification as the task performed by the target ML models built on these datasets.

Keywords : Membership inference attacks, deep leaning, privacy risk, differential privacy, FDR, FS, Dataset, Train, Test, Attack, Genetic Algorithm.

I. INTRODUCTION

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 (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 user cannot even download the model anyhow! 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. We don't have access to the training datasets of the training model, so in this paper we make prediction, there is no interaction with the dataset of the machine learning model, we just get the output prediction vector. But, the real question is, do these machine learning models tend to leak information about their training data? In this paper learnt about the tendency of the leakage through studying membership inference attack against the machine learning models. This paper aim is to find out as an attacker assuming that attacker has some information of the distribution or access to some part of the dataset, whether it was the part of the model's training dataset or not. It is challenging as we don't have direct access of the model or the dataset.

II. OVERVIEW

Machine learning model tend to behave differently with the training data as compared to the dataset that it hasn't seen. This phenomenon is called overfitting where the accuracy on training dataset is higher compared to testing dataset. The objective is to construct an attacking model that can classify the membership of the dataset used to query the target model. Attack model is collection of 'k' attack model, each designed for 'k' different classes. This simply increases the attack accuracy as target model generates distribution of probabilities. We have used supervised learning to design multiple shadow model and used its labeled inputs and outputs to train the attack model. Formal setting is as described.

Suppose $mtarget()$ is a target model and has a disjoint training dataset as $Dtarget\ train$ and contains labelled records in format of $\{xi\ ,\ yi\ \}target$ where xi is the input data and yi is it's true label taken from k classes. The predicted output is a vector of probabilities of 'k' size with probability ranging [0,1]. Summation of these probabilities is 1.

Similarly, $mattack()$ is an attack model that takes input $xattack$, which is combination of labelled record and prediction vector that is of size 'k'. This model is a binary classifier that infers the membership and outputs, 'out' or 'in'. Figure (1) shows the entire process. Here, a record {x, y} is used by the target model to predict a vector $\hat{y} = mtarget()$. We pass {y, $\hat{y}\}target$ to the attack model. The attack model computes the probability whether the {y, $\hat{y}\}target$ is in training set or testing set of $mtarget()$

III. Overview of Inference ML Attack

When compared to a dataset that it hasn't seen, an AI model will almost always behave differently with preparation information. Over fitting is a quirk where the precision of the preparation dataset is higher than the testing dataset. Building a model that can group the involvement of the dataset used to investigate the objective model is the objective.

Assault models are a collection of "k" different assault models, each designed for "k" different classes. As a result of the distribution of probabilities produced by the target model, the assault precision is essentially increased. In order to construct the attack model, we used directed learning to plan a number of shadow models and made use of their marked information sources and outcomes. The setting is formal as seen. Assume that mtarge() is an objective model and that Dtarget train also contains named records in the organisation of "xi, yi"target, where "xi" stands for the informational data and "yi" for the actual mark obtained from k classes.

A vector of probabilities with a size of "k" and a likelihood running between [0,1] is the expected outcome. The result of adding these probabilities is 1. Similarly, mattack() is an attack model that accepts as input xattack, a combination of named records and a forecast vector of size 'k'. This model is a two-class

classifier that determines participation and produces the outcomes "out" or "in." The entire cycle is seen in Figure 1. Here, the objective model uses a record x, y to predict a vector y = mtarge () We switch the "y, y" target to the attack mode. l. The assault model analyses the possibility that the "y, y" target is in a set that is being prepared or tested for a weapon. ().

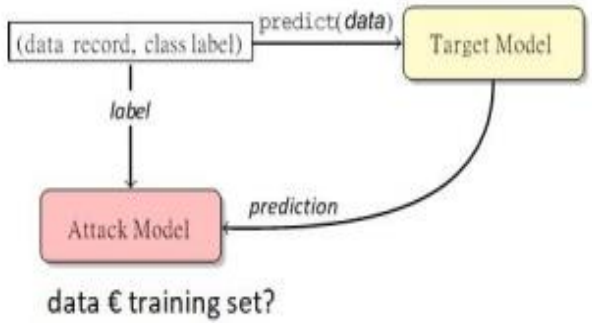


Figure 1: End-to-End Process

IV. Shadow Models

Setup: 'n' shadow models *mshadow i* () are created by the attacker. Each *i th* shadow model is trained on *Dshadowi train* , each of same type. In the worst case it is assumed that the *Dshadowi train* and *Dtarget train* may be disjoint. The shadow models are designed and trained in the same way as the target model (i.e. the user can use the same API used for training the target model if no information about the model architecture is known). As the number of shadow model increases, the accuracy of the attack increases. Figure (3) shows the above explanation.

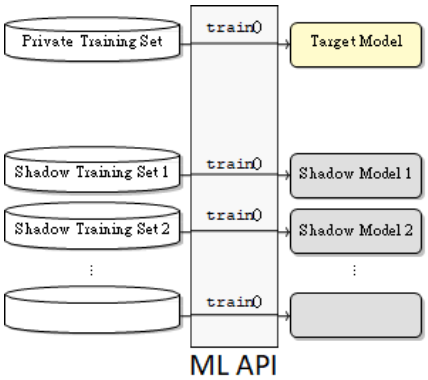


Figure 2: Shadow Model Trained using same API as the Target Model

V. Attack Model

The experimental setup of the training of the attack model is shown below. The setup shows that the shadow model's output is used to train the attack model and it learns how to infer the membership of the dataset of the shadow model and thus produces a sequence to predict the membership of the training set of the target model. The shadow model is queried using its own dataset for training and a disjoint testing dataset. The output generated by the training data is labelled as 'in' and the output by dataset for testing as 'out'. This record is used to train the attack model. Figure (3) shows, how we have trained the attack model. For each {x, y} € *Dshadowi train* , a prediction vector $\hat{y}$ along with its membership is added to the record of training set of attack model (y, $\hat{y}$, in). Similarly, for each {x, y} € *Dshadowi test* we get a record (y, $\hat{y}$, out). A *Dattack train* is formed using such records and is partitioned into 'k' partitions (k = number of classes). Each *Dattack train* partitioned is associated to its respective class. It is like for each 'y' train a different model that would predict the membership for every x, given $\hat{y}$. Attack model is thus basically a binary classifier.

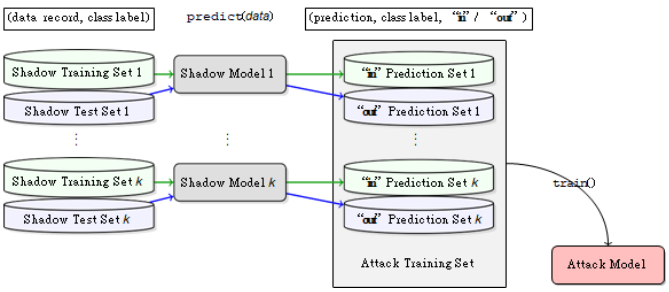


Figure 3: Training of Attack Model

VI. Setup

Description of the experimental setup for both CIFAR-10 and UCI Adult datasets is given below. The baseline accuracy for both experiments with shuffled dataset is assumed to be 0.5. A) CIFAR: CIFAR-10 is widely used in image recognition examples. It consists

of 10 classes, each containing 6,000 32x32 color images per class. 50,000 for training and 10,000 for testing. Thus, in total there are 60,000 32x32 color images. We have combined both datasets and shuffled them all and then first extracted samples for target model and from the rest for the shadow model. By this we increase the probability for the dataset for target and shadow being disjoint. We have used distinct size of datasets in our experiment to picture the changes in the accuracy due to different datasize. The main aim for the classifier to is decide that the object belongs to which class. Different datasize for our experiment are: 2500,5000,10000,15000 both for target and shadow model (i.e. 2500 each for training and testing of the target model and similarly for each shadow model). For each dataset, we have created 10 shadow models. The number of shadow models are selected according to number of classes for the datasets and here CIFAR 10 has 10 classes. We have compiled the target and shadow model using neural network. Two convolution layers with 'tanh' activation is used. The first convolution layer has kernel size = 5x5 and other has 3x3. Dense layer = 128 also with 'tanh' activation and at last a dense layer with 'softmax'. Categorical_entropy is used as a loss function because it is a multiclass classification. decay rate = 1e-7, learning rate = 0.001. For attack model we have used SVM. To reach to the final result we have run a cross validation loop with nfold = 5 for C_test = [0.1,1,10] and gama_test = [0.001,0.01,0.1]. Through this we selected the best C and gamma for SVM.

```
Target model summary
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 28, 28, 32)        2432
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 14, 14, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 12, 12, 32)        9248
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 6, 6, 32)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 1152)              0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               147584
_________________________________________________________________
dense_2 (Dense)              (None, 10)                1290
=================================================================
Total params: 160,554
Trainable params: 160,554
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 4: CIFAR10 Target Model Summary

For different datasize, we got average accuracy over total datasets as below: For ds = 2500
Attack Precision: 0.7849293563579278
Attack Recall: 1.0
Attack Accuracy: 0.863

For ds = 5000
Attack Precision: 0.7275902211874272
Attack Recall: 1.0
Attack Accuracy: 0.8128

For ds = 10000
Attack Precision: 0.7211365111415591
Attack Recall: 1.0
Attack Accuracy: 0.80665

For ds = 15000
Attack Precision: 0.7089516967577276
Attack Recall: 1.0
Attack Accuracy: 0.7947333333333333

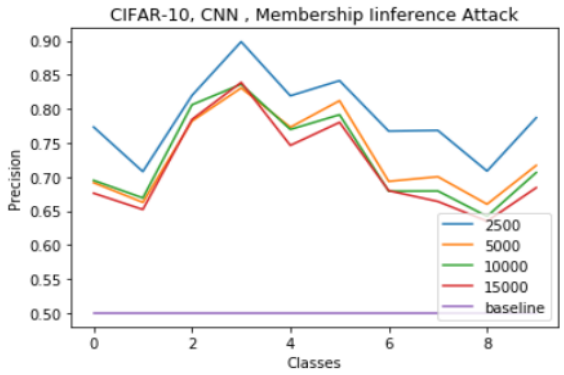
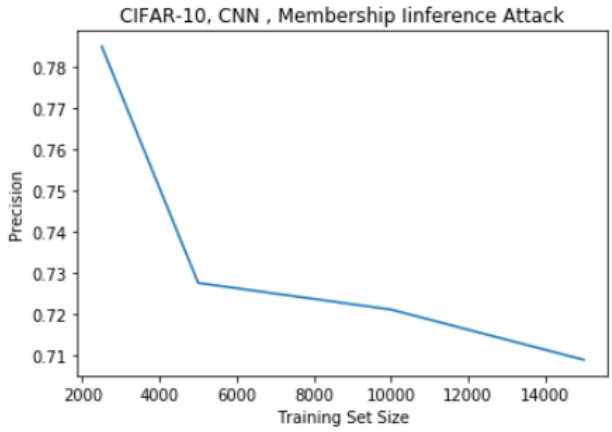


Figure 5: Precision v/s class Graph for different Datasize of CIFAR10



VII. Observation

In this paper attack works. This is because this model has overfitted. Also, large number of classes makes the job of the model hard as it would have to go through lots of information. This lead, to leak of more information. We have given one mitigation strategy for that 'use of regularization' that helps to overcome overfitting and it works.

Evaluation Criteria

Different measures can be used to assess how accurate our attack model is. Among them are:
Classification, first Measures accuracy by comparing the proportion of accurate predictions to all input samples [53].

2. Logarithmic Loss: This technique penalises the incorrect categorization [58].

3. Confusion Matrix: Identifies true positives, true negatives, false positives, and false negatives for the model's entire performance [82].
4. Area Under Curve: This indicates the likelihood that a randomly selected positive example would be chosen rather than a negative one. The ranking is based on the data's sensitivity and specificity [29].
5. Mean Absolute Error: This statistic measures the average discrepancy between actual and expected values [94].
6. Mean Squared Error: This statistic computes errors by averaging the squares of the variances between the actual and predicted values [90].

Precision is the model's capacity to return just pertinent examples [65].
Recall is the model's capacity to locate all pertinent instances [65].

Precision and recall are the two common metrics used to gauge attack accuracy. The precision of the attack model indicates the percentage of records that are actually members of the training dataset.

$$\text{precision} = \frac{\text{truepositives}}{\text{truepositives} + \text{falsepositives}}$$

But recall represents what fraction of the members of the training dataset are correctly inferred as members by the attacker.

$$\text{recall} = \frac{\text{truepositives}}{\text{truepositives} + \text{falsenegatives}}$$

As recall focused on correctly inferred members of the training dataset, we considered it as the evaluation metrics in our experiments.

VIII. UCI Adult (Census Income)

This dataset has total 48,842 samples and 14 census features like gender, occupation, native country, marital status, age, working hours, education, race, etcetera. The main aim for the classifier is to predict

whether a person earns more than $50K based on the features or not. Here we have randomly chose 10,000 train and testing samples for the target model and 10,000 random samples for training and testing shadow models.

We have created 20 shadow models; each shadow model gets 10,000 training sample from shuffled and disjoint dataset than the one used for the target model. Number of shadow models can be increased to increase the prediction accuracy. We have compiled all the models on the local machine. We have used keras library with tensorflow working in the backend for creating neural networks. All the features with 'object' data types are one-hot-coded

Neural network with 5 hidden layers, decay rate = 1e-7, learning rate = 0.001 and sigmoid activation gives us 79.9% and 81% training accuracy for the target and shadow models respectively. 100 epochs are run for each model. Summary of the target and shadow model with its accuracy is shown below:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
hidden (Dense)               (None, 5)                 530
_________________________________________________________________
output (Dense)               (None, 1)                 6
=================================================================
Total params: 536
Trainable params: 536
Non-trainable params: 0
_________________________________________________________________
None


For target model with training datasize = 10000
Training accuracy = 0.799100
Validation accuracy = 0.791500
```

Figure 6: UCI Adult Target Model Summary

```
Shadow Model Summary
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
hidden (Dense)               (None, 5)                 530
_________________________________________________________________
output (Dense)               (None, 1)                 6
=================================================================
Total params: 536
Trainable params: 536
Non-trainable params: 0
_________________________________________________________________
None
Shadow model no: 0


For shadow model with training datasize = 10000
Training accuracy = 0.811300
Validation accuracy = 0.813700
UCI_Adult_shadow_10000_0.h5
```

Figure 7: UCI Adult Shadow Model Summary

The output for the target and shadow model is a single class prediction output of probability whether the person earns more than $50K or not. 'binary_crossentropy' is used as the loss function. The same model format is used for the attack model as it is also a binary classifier. The attack model gets the training accuracy for 40,000 datasets to be 49.91 % and validation accuracy 50% (same as base line accuracy).

```
Attack Model Summary
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
hidden (Dense)               (None, 5)                 10
_________________________________________________________________
output (Dense)               (None, 1)                 6
=================================================================
Total params: 16
Trainable params: 16
Non-trainable params: 0
_________________________________________________________________
None


For attack model with training datasize = 400000
Training accuracy = 0.499115
Validation accuracy = 0.500000
```

Figure 8: UCI Adult Attack model summary

There are two reasons why membership inference appeared to fail for this model. (1) As model is not overfitted, the training and testing accuracy are almost similar). (2) The model is a binary classifier as the attacker just must infer the membership by studying the behavior of the model with single class. Since the outputs are complimentary, it is not enough for the attack model to infer the membership information.

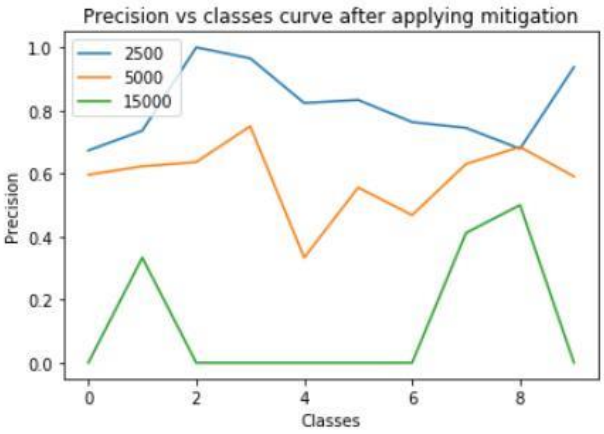
IX. Observation

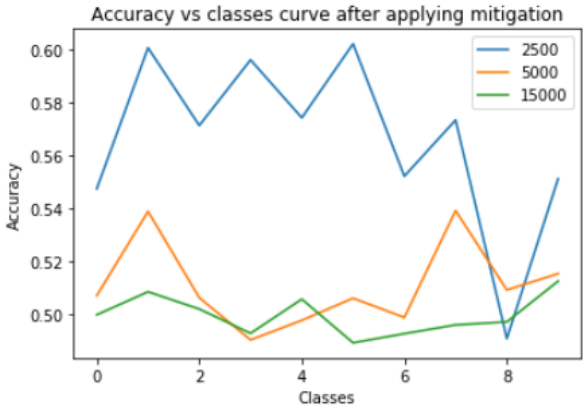
Models with few classes are less prone to leak their membership information. As the number of classes increase, the model gets more features from the sample to classify the input samples with higher accuracy. To simplify, models that have more classes have job to remember and classify more features and information about their training data and so are prone to leak more information.

X. Mitigation Strategy: (' Use of Regularization)

Regularization techniques are normally used to overcome overfitting in machine learning. We have used L-2 regularization that penalizes large no of parameters. We have use lambda = 5e-3

For ds = 2500
Attack Precision: 0.7863247863247863
Attack Recall: 0.184
Attack Accuracy: 0.567
For ds = 5000
Attack Precision: 0.61
Attack Recall: 0.061
Attack Accuracy: 0.511
For ds = 15000
Attack Precision: 0.4074074074074074
Attack Recall: 0.0007333333333333333
Attack Accuracy: 0.49983333333333335





XI. CONCLUSION

As AI becomes pervasive, mainstream researchers turns out to be progressively intrigued in its effect and aftereffects as far as security, protection, decency, and logic. This study directed a complete investigation of the cutting edge protection related assaults and proposed a danger model and a binding together scientific categorization of the various kinds of assaults dependent on their qualities. An inside and out assessment of the present status of the workmanship research permitted us to play out a definite investigation which uncovered normal plan examples and contrasts between them.

A few open issues that legitimacy further exploration were recognized. To start with, our investigation uncovered a fairly tight focal point of the exploration directed up to this point, which is overwhelmed by assaults on profound learning models. We trust that there are a few well known calculations and models in wording of certifiable organization and materialness that merit a nearer assessment. Second, an exhaustive hypothetical comprehension of the purposes for security spills is as yet immature and this influences both the proposed safeguarding strategies and our comprehension of the impediments of security assaults.

XII. REFERENCES

[1]. Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang, Deep learning with differential pri vacy, Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 308–318.

[2]. Giuseppe Ateniese, Luigi V Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici, Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers, Inter

national Journal of Security and Networks 10 (2015), no. 3, 137–150.

[3]. Jordan Awan, Ana Kenney, Matthew Reimherr, and Aleksandra Slavkovi´c, Benefits and pitfalls of the exponential mechanism with applications to hilbert spaces and functional PCA, 2019.

[4]. Michael Backes, Pascal Berrang, Mathias Humbert, and Praveen Manoha ran, Membership privacy in MicroRNA-based studies, Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 319–330.

[5]. Raef Bassily, Adam Smith, and Abhradeep Thakurta, Private empirical risk minimization: Efficient algorithms and tight error bounds, 2014 IEEE 55th Annual Symposium on Foundations of Computer Science, IEEE, 2014, pp. 464– 473.

[6]. Raphael Bost, Raluca Ada Popa, Stephen Tu, and Shafi Goldwasser, Machine learning classification over encrypted data., NDSS, vol. 4324, 2015, p. 4325.

[7]. Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu, A limited mem ory algorithm for bound constrained optimization, SIAM Journal on scientific computing 16 (1995), no. 5, 1190–1208.

[8]. Nicholas Carlini and David Wagner, Towards evaluating the robustness of neu ral networks, 2017 ieee symposium on security and privacy (sp), IEEE, 2017, pp. 39–57.

[9]. Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate, Differentially private empirical risk minimization, Journal of Machine Learning Research 12 (2011), no. 29, 1069–1109.

[10]. Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz, GAN-Leaks: A taxon omy of membership inference attacks against generative models, Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Se curity (New York, NY, USA), CCS '20, Association for Computing Machinery, 2020, p. 343–362.

[11]. James S Cramer, The origins and development of the logit model, Logit models from economics and other fields 2003 (2003), 1–19.

[12]. Sander Dieleman, Jan Schl¨uter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, Daniel Nouri, Daniel Maturana, Martin Thoma, Eric Battenberg, Jack Kelly, Jeffrey De Fauw, Michael Heilman, Diogo Moitinho de Almeida, Brian McFee, Hendrik Weideman, G´abor Tak´acs, Peter de Rivaz, Jon Crall, Gregory Sanders, Kashif Rasul, Cong Liu, Geoffrey French, and Jonas Degrave, Lasagne: First release., August 2015.

[13]. Irit Dinur and Kobbi Nissim, Revealing information while preserving privacy, Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Sympo sium on Principles of Database Systems (New York, NY, USA), PODS '03, Association for Computing Machinery, 2003, p. 202–210.

[14]. Pedro Domingos, A few useful things to know about machine learning, Com mun. ACM 55 (2012), no. 10, 78–87.

[15]. Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville, Adversarially learned inference, 2017.Adi, E 2012, „A design of a proxy inspired from human immune system to detect SQL injection and cross-site scripting", Procedia Engineering, vol. 50, pp. 19–28.

Cite this article as :

Preeti, Irfan Khan, "Membership Inference Attacks on Machine Learning Model", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 8 Issue 5, pp. 31-38, September-October 2022. Available at doi : https://doi.org/10.32628/CSEIT22856

Journal URL : https://ijsrcseit.com/CSEIT22856