MCE and ECE MIAShield implementation on Location-30 dataset

Final Report - Group 4

Grover Susanibar
CECS
University of Michigan
Dearborn, MI, US
groverhs@umich.edu

Jasleen Chahal
CECS
University of Michigan
Dearborn, MI, US
jasleenk@umich.edu

Manasi Kshirsagar
CECS
University of Michigan
Dearborn, MI, US
maksh@umich.edu

Sayali Agalave
CECS
University of Michigan
Dearborn, MI, US
sagalave@umich.edu

Spencer Scott
CECS
University of Michigan
Dearborn, MI, US
sgscott@umich.edu

ABSTRACT

The purpose of this paper is to implement and compare the Model confidence based exclusion (MCE) and Exact signature based exclusion (ESE) oracles as a defense to the Membership Inference attacks. The membership inference attacks are referred to as MIAs. MIAs are the common attacks where an adversary can access the samples used to train the model. An example would be sensitive healthcare data being used to train a model that can be used to classify other individual's records if accessed by an adversary.

In order to overcome this problem, MIAShield was introduced which works on the principle of "primitive exclusion" of target. The core idea is to divide the training datasets into disjoint datasets and then train the model in such a way that each sample belongs to one of the dataset. This avoids the problem of overfitting and takes into account the exclusion oracle.

In MIAShield, the training data set is first split into n disjoint subsets which are used to train the model. It assures that the target data point belongs to only one dataset. The excursion oracle eliminates the model that contains the data point given for prediction.

In this paper, We have implemented the MCE and ESE oracles based on one benchmark Location dataset. The dataset consists of binary data.

KEYWORDS

Location-30, MCE, ESE, data privacy, MIAShield, Machine Learning, Adversary Attack.

1 Introduction

The privacy threat in Machine Learning Algorithms is that models can leak information about training data points. This data leak may lead to violation of user privacy. The paper elaborate more on one of the information leakage attacks named Membership Inference Attacks (MIA). The goal of the MIA attack is to identify if the given data point is used to train the model. These data points can be identified by output of the decision function. For matching data points, models show high confidence.

The paper explains about MIAShield mechanisms. These mechanisms based on preemptive exclusion of matching members (data points) from the training dataset, results in the weakening of the signal. The success of the defensive model is based on the effectiveness of exclusion of Oracles. The paper explains 5 different models based on exclusion Oracle:

1. Model Confidence Based Exclusion (MCE)

In this implementation, the most confident model will be excluded. Since the most confident model will most likely include the data sample.

2. Exact Matching Based Exclusion (ESE)

In this implementation, exact matching models will be excluded.

3. Approximate Matching Based Exclusion (ASE)

In this implementation, approximate matching models will be excluded.

4. Classifier Based Exclusion (CBE)

In this implementation, classifier or design a predictive model is trained to decide whether to exclude the particular model or not.

5. Chain of Exclusion Oracles (COE)

In this implementation, data is validated over exact matching. If matching occurs, it will exclude that dataset.Then data will be revalidated over Approximate matching, if match found it will exclude that dataset. After that data will be validated over the classifier based exclusion, and exclude if a match occurs.

In the project we are implementing Model Confidence Based Exclusion (MCE) and Exact Matching Based Exclusion (ESE) on Location 30 dataset.

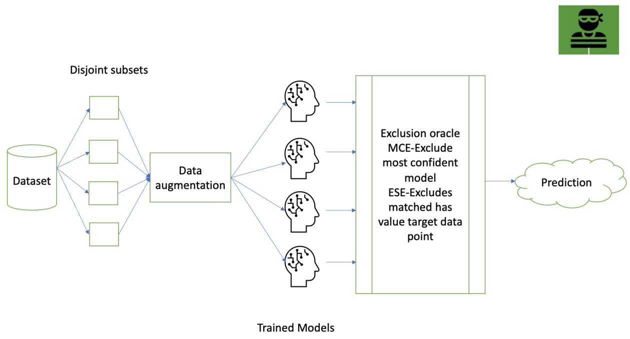

Disjoint subsets
Dataset
Data augmentation
Exclusion oracle
MCE-Exclude most confident model
ESE-Excludes matched has value target data point
Prediction
Trained Models

Figure 1. MIAShield Flowchart

2 Related Work

2.1 Location-30 Dataset Origin

The work done in this paper is heavily based on the MIAshield paper [1]. Given that the major change introduced in this paper is using MIAshield oracles with the Location-30 dataset. Therefore, the related work in this section will focus on Membership Interference Attacks (MIA) using the Location-30 database. There have been several examples of MIA on the Location-30 dataset [4,5].

2.2 MIA using the Location-30 Dataset

A model stacking defense method is proposed by A. Salem et al [4] in which the model is trained with smaller subsets of the full dataset. THis is very similar to the method used by MIAshield labeled Classifier-Based Exclusion (CBE)[1]. Model stacking was shown to reduce the MIA performance by 20% for the precision and 30% for the recall [4]. Precision is defined as the number of relevant items returned divided by all items returned. While recall is the number of relevant items returned divided by the total number of all relevant items.

2.3 MIA using other Datasets

The origin paper MIAshield demonstrates the defense of MIA on the CIFAR-10, CIFAR-100, and CH-MNIST datasets [1]. While the work done by M. Nasr et al [6] provides an analysis of MIA on the CIFAR-100, Purchase100, and Texas100 datasets.

3 Dataset

The Location-30 dataset is a preprocessed dataset that clusters the data into 30 different geosocial classes created by Shokri et al [2] based upon the full Foursquare dataset [3]. The origin data collection was done by tracking social media check-ins of users [7]. The Location-30 dataset contains the location check-in records for thousands of individuals. The dataset has 5,010 user records and 446 binary features. The feature is whether the location or region is visited or not. Furthermore, the dataset is divided into 30 geosocial types. The geosocial types were assigned by evaluating the location visited of each of the 5010 users [2].

4 Methodology

4.1 Data Analysis

First of all, it is very important to perform an exploratory data analysis of the dataset in order to understand the target response and the interaction between the features. The target geosocial type has thirty categories and we can see its distribution in Figure 2.

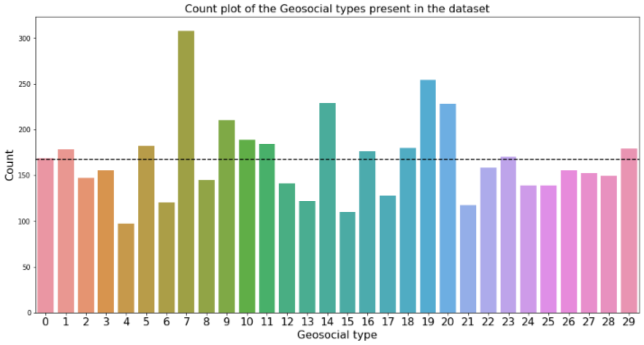
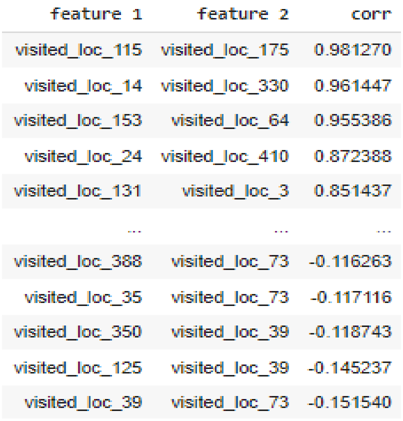


Figure 2. Count plot of the geosocial types

In the count plot of the geosocial types, we can see that type 7 is the most common geosocial type, it represents 6% of the cases. On the other hand, type 4 is the less common geosocial type as it represents less than 1% of the cases. Then the dataset has imbalanced classes.



feature 1	feature 2	corr
visited_loc_115	visited_loc_175	0.981270
visited_loc_14	visited_loc_330	0.961447
visited_loc_153	visited_loc_64	0.955386
visited_loc_24	visited_loc_410	0.872388
visited_loc_131	visited_loc_3	0.851437
...	...	...
visited_loc_388	visited_loc_73	-0.116263
visited_loc_35	visited_loc_73	-0.117116
visited_loc_350	visited_loc_39	-0.118743
visited_loc_125	visited_loc_39	-0.145237
visited_loc_39	visited_loc_73	-0.151540

Table 1. Feature Correlation

Then we can analyze the correlation between the features, this is shown in Table 1. We can see that there are some features highly correlated with other ones meaning that they share almost the same information. For example, feature 115 is highly correlated with feature 175 (correlation 0.98), thus these two variables almost give the same information and we should use only one of those variables. Something similar occurs between features 14 and 330, features 153 and 64, features 24 and 410, and features 131 and 3.

In Figure 3, boxplots for some features are plotted to see the relationship between these features and the geosocial type. We can conclude that people who visited places 307, 413, and 125 are more likely to belong to a higher geosocial type.

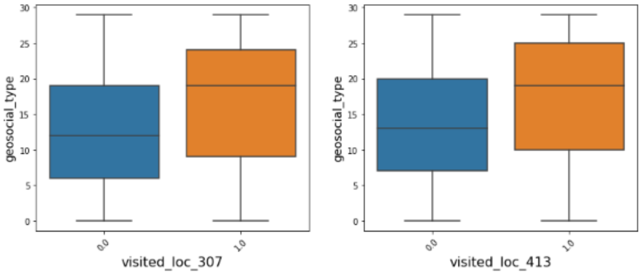


Figure 3. Boxplots for some features

4.2 Data Augmentation

The Location-30 dataset has 5'010 records, and this may not be enough data to properly train a supervised model. And even worse if later we split the dataset into small subsets to implement the oracle methods. Thus, we need to perform a data augmentation. There are several methods to perform data augmentation on tabular data like oversampling minor classes, and autoencoder-based augmentation. In our case, we saw that we have an imbalanced dataset, thus oversampling is the best method to apply.

Specifically, we applied the Synthetic Minority Oversampling Technique (SMOTE) method to deal with the imbalanced data and increase the samples from the minority classes. In this way we increased the size of the dataset from 5'010 to 9'215 records, i.e. an increase of 84% of the original size.

4.3 Machine Learning Model

For this case, we decided to train a XGBoost classifier. Also known as Extreme Gradient Boosting, XGBoost is an ensemble method that trains a sequence of decision trees and here is the main difference with Random Forest algorithm. Random Forest has individual decision trees in parallel while a gradient boosting algorithm has decision trees in sequence. Each decision tree or weak learner tries to compensate for the errors from the preceding models. These errors from the preceding models are calculated by the gradient of the output response and each train data point has a relative weight. In this way the next model in the sequence focuses on the data points with larger errors. The advantage of the XGBoost is its execution speed and it allows to add a regularization term like Ridge or Lasso penalization terms to the weights of the learners. In this way we can avoid overfitting.

4.4 Hyper-parameter Tuning

Before training the XGBoost model, many hyper-parameters have to be set. The XGBoost algorithm has several hyper-parameters like the number of estimators, criterion Gini vs entropy, maximum depth of the trees, minimum number of samples to split a leaf. So we can tune these hyper-parameters to have better performance. Typically, there are two ways to perform a hyperparameter tuning: Grid Search and Random Search. In the Grid Search method, we can define a list of possible values for each hyper-parameter and then make the cartesian product and train the algorithm with all the possible combinations. It returns the combination with the best validation performance. In the Random Search method we can define a distribution for each hyper-parameter and then randomly sample from the possible combinations. So Random Search can try less combinations but the final combination could not be the optimal hyper-parameter combination.

4.5 Cross Validation

Since we are dealing with a small dataset, splitting the dataset into train and test sets would not be a good idea. Then, We decided to apply cross validation to evaluate the performance of the model in five subsets of the data. Cross validation helps to avoid training and testing on the same subsets of data and it's suitable for cases in which there is not enough data. This method helps to have more reliable results in test performance. Specifically, we applied K-fold cross validation using k=5.

4.6 MIAShield MCE implementation

For the Model Confidence Based Exclusion implementation, first we divided the augmented dataset into five subsets, of course previously shuffling the data. Then for each subset we trained an XGBoost model and performed cross validation. Then we implemented the MCE function that receives a location array as input and then it got five predictions from the individual models. The most confidence model is identified and subsequently excluded. The final prediction is calculated using the remaining models.

4.7 MIAShield ESE implementation

For the ESE method, we used the same 5 individual models but this time we had to determine whether the location array to predict is part of the training set in any of the individual models or not. For this, we applied a hash function to each training location array and stored the results in a dictionary whose keys are the hash results and its values are the id of the individual model that contains such an image. Then we implemented an ESE function that receives a location array as input and then got five predictions from the individual models. We then verified if the input location array belongs to the training dataset by looking into the hash dictionary. If the hashed location is in the dictionary, the individual model is identified and excluded. The final prediction is calculated using the remaining models.

5 Experiment Evaluation

On evaluating and implementing the two exclusion oracles closely, following differences in Table 2 can be inferred:

	MCE (Model Confidence-Based Exclusion)	**ESE (Exact Signature Matching Exclusion)**
Goal	MCE serves as a baseline method to prevent root cause of MIA i.e overfitting by excluding the most confident model based upon the most voted prediction.	ECE uses cryptographic hash value comparisons as a signature matching method to eliminate the training models which have the same hash values.
Method	Eliminates the data points corresponding to the highest confidence score obtained from most voted prediction as it may contain target data points.	ESE uses SHA-1 hashing function, to generate a unique hash value for each sample. Then a search algorithm is used to perform sample lookup to eliminate the same hash values.
Drawbacks	Training models can predict wrong labels due to spurious correlations resulting in elimination of wrong data points.	2 potential limitations: (i) ESE fails to exclude x' ≈ x where x' is a slightly modified value of x. (ii). It is vulnerable to timing-attacks i.e The response time difference of prediction of members vs non member is different. It takes longer to search time for non-members, which gives a threshold response timing value that can be used to

		perform MIA. In order to solve this problem, reshuffling is done.
Limitations	As MCE provides baseline elimination but is not so sufficient, hence, more accurate alternatives elimination techniques are required.	As ESE fails to exclude the data points with small modifications, hence, ASE i.e Approximate Signature-based elimination technique is used to overcome this downfall.
Results	For implementing MCE, the dataset is divided into 5 disjoint models with an accuracy of 68.85%, 70.75%, 69.07%, 70.16% and 70.05% respectively. Using cross-validation we predict the highest voted class as 12 which is then eliminated.	For ESE implementation, hash length of 6899 was obtained and then ESE function was used to find exact class matching to predicted class and we obtained the predicted class as 12 that needs to be eliminated.

Table 2. Oracle Comparison

6 Discussion, Limitations and Future Work

6.1 Discussion

Our dataset was not enough to achieve the satisfying accuracy. In order to overcome the issue of imbalance data we have used oversampling. Further, We have applied cross validation to evaluate the accuracy of the model.

6.2 Limitations

In spite of the techniques and methods applied to the model the implemented 2 oracles MCE and ESE have few limitations. In case of MCE it may not always be the case that the model with the highest confidence would hold the target data point as the label can turn out to be the wrong one in the prediction. While the training is in progress it might happen that the model would choose the coincidental correlation rather than the original distinguishing feature.

In case of ESE oracle there are few potential limitations: one being the slight modification in the inputs by the adversary also fools the model resulting in non detection of the member and second being the model tends vulnerable to timing attacks due to predictable response time between member data points and non-member data points. There is a way to reshuffle the hash values which creates a confusion to an adversary but does not make the method completely free from the timing attacks.

Also, there is no way to perform exclusion oracle methods approximate signature based exclusion and classifier based exclusion as our dataset is location based and location can't be modified. The larger manipulations are likely to degrade the model accuracy.

6.3 Future work

Hence, from the accuracy perspective, there is still a major scope for improvement. The various exclusion can be considered for the same.

7 Conclusions

In the paper, MIA Shield is implemented, and two Exclusion oracles are used as a defense to the Membership Inference attacks by excluding the model containing target data points which is used to train the model. The two exclusion oracles implemented are Model confidence-based exclusion (MCE) and Exact signature based exclusion (ESE) oracles. Firstly data analysis is performed on the Location dataset containing 30 geosocial classes by calculating the correlation between them. Then data augmentation is performed to gain back the accuracy loss when splitting the dataset into disjoint subsets and increase the length of samples in the dataset. The Synthetic Minority Oversampling Technique (SMOTE) is used for the same. Then Extreme Gradient Boosting, XGBoost i.e. an ensemble method is used to train a sequence of decision trees in order to make a machine learning model. XGBoost increases the execution speed and allows addition of regularization terms to the weights of the learners so that overfitting can be avoided. The XGBoost algorithm has several hyper-parameters like the number of estimators, criterion Gini vs entropy, maximum depth of the trees, minimum number of samples to split a leaf which are tuned for better performance of the model. In two ways, a hyperparameter tuning is performed: Grid Search and Random Search. Finally, K-fold cross validation with k=5, is used to split the dataset into 5 subsets to have more reliable results in test performance.

When implementing and comparing the above models it can be inferred that MCE acts as a baseline and is not enough for actual exclusion of true data points while ECE provides accuracy but even with small change in the dataset it can fail to predict the similarity between input values. Overall, after implementing MCE and ESE, we tested the models using a random test data point and we got the same prediction class i.e class 12. Hence, as we used cross validation, it was hard to verify the overall effectiveness and compare the two exclusion oracles.

Finally, we can conclude that we were successful in implementing the two oracle methods and we test their performance using some random location data.

REFERENCES

[1] Ismat Jarin, Birhanu Eshete (2022). MIAShield: Defending Membership Inference Attacks via Preemptive Exclusion of Members. Proceedings of the 23rd Privacy Enhancing Technologies Symposium (PETS 2023).

[2] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. (2017). Membership inference attacks against machine learning models. In IEEE Symposium on Security and Privacy (SP).

[3] “Foursquare Dataset.” *Dingqi YANG's Homepage*, https://sites.google.com/site/yangdingqi/home/foursquare-dataset.

[4] A. Salem, Y. Zhang, M. Humbert, M. Fritz, and M. Backes. (2019). ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. NDSS.

[5] L. Song, and P. Mittal (2021). Systematic Evaluation of Privacy Risks of Machine Learning Models. USENIX Security Symposium.

[6] M. Nasr, R. Shokri and A. Houmansadr (2019) Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. 2019 IEEE Symposium on Security and Privacy (SP).

[7] D. Yang, D. Zhang, and B. Qu. (2016). Participatory Cultural Mapping Based on Collective Behavior Data in Location-Based Social Networks. ACM Trans. Intell. Syst. Technol. 7, 3, Article 30