MIASec: Enabling Data Indistinguishability Against Membership Inference Attacks in MLaaS

Chen Wang, *Senior Member, IEEE*, Gaoyang Liu, *Student Member, IEEE*, Haojun Huang, Weijie Feng, Kai Peng, and Lizhe Wang, *Senior Member, IEEE*

Abstract—The emerging of machine learning has massively promoted the abilities of computational sustainability in natural resource management and allocation. Many Internet giants such as Google, Amazon, and Microsoft now provide Machine Learning as a Service (MLaaS) to meet the increasing demand for machine learning services. However, the prediction results of training data and testing data with the same machine learning model in MLaaS have remarkable differences, and thus the attackers can leverage machine learning techniques to launch the so-called membership inference attacks, i.e., to infer whether a record is in the training data or not. In this paper, we propose MIASec that can guarantee the data indistinguishability of the training data and thereby has the ability to defend against membership inference attacks in MLaaS. The key idea of MIASec is to narrow the dynamic ranges of vital features in the training data, such that the training data, the testing data, and even the synthetic data have almost semblable prediction results by the same machine learning model. With elaborated design on modifying the values of vital features in the training data, MIASec can thus reduce the differences between the model's outcomes of training data and testing data, thereby protecting the training data in effect while keeping the model's accuracy stable. We empirically evaluate MIASec on machine learning models trained by off-line neural networks and on-line MLaaS. Using realistic data and classification tasks, our experiment results show that MIASec can defend the membership inference attacks effectively. In particular, MIASec can reduce the precision and recall of attacks respectively by 11.7 and 15.4 percent in average, and by 18.6 and 21.8 percent at best.

Index Terms—Membership inference attack, MLaaS, K-Means, data indistinguishability

1 Introduction

Computational sustainability, a branch about the sustainability research in sustainable solutions and their implementation, has attracted the attentions and interests of researchers in computer science, while in recent years, machine learning has dramatically reduced the cost of human labors while improving the utilization of computing equipments [1], [2], [3].

To put this in context, machine learning as a service (MLaaS), a range of services offering machine learning tools as part of cloud computing services, has recently sprouted up to meet this demand by Internet giants such as Google, Amazon and Microsoft. MLaaS allows customer companies to get access to machine learning technologies, and to assess and learn from data without requiring in-house domain expertise.

1.1 Motivation

MLaaS platforms allow users to run various data analytics or model the data of their own. The obtained models via MLaaS can be further deployed to services of the users. During this process, however, potential vulnerabilities of MLaaS have been recently found, one of which is the so-called *membership inference attacks* [4], [5], i.e., determining whether t was used as part of T or not, given an instance t and black-box access to a machine learning model trained on a dataset T.

Membership inference attacks can be considered invasion of the privacy of both the individual participants and the owner of training data. The former refers to the membership privacy of individuals who are participated in the model training process while the latter concerns unauthorized leakage of business secrets risks. Unfortunately, MLaaS platform operators at this time neither explicitly give warning nor provide solutions to these risks [4], [5]. In this case, it is necessary to design effective defense techniques to protect against membership inference attacks in MLaaS.

At present, there is no effective defense technique specially designed against the membership inference attacks in MLaaS. Differential privacy may be one potential countermeasure, which can theoretically guarantee privacy leakage about data of one specific user. However, existing mechanisms achieving differential privacy (e.g., the Laplacian mechanism [6] and the exponential mechanism [7]) are either computationally infeasible on high dimensional data, or practically ineffective due to huge utility loss. For this reason, it is still not clear yet how to effectually defend against membership inference attacks in MLaaS.

- *C. Wang, G. Liu, H. Huang, W. Feng, and K. Peng are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China. E-mail: {chenwang, liugaoyang, hjhuang, wjfeng, pkhust}@hust.edu.cn.*
- *L. Wang is with the School of Computer and Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences, Wuhan, Hubei 430074, China. E-mail: lzwang@ceode.ac.cn.*

Manuscript received 1 Sept. 2018; revised 27 May 2019; accepted 15 July 2019. Date of publication 23 July 2019; date of current version 8 Sept. 2020. (Corresponding author: Haojun Huang.)
Recommended for acceptance by Y. Wu, Y. Pan, N. Georgalas, and G. Min.
Digital Object Identifier no. 10.1109/TSUSC.2019.2930526

1.2 Our Contributions

In this paper, we propose MIASec that can guarantee the *input data indistinguishability* and thereby has the ability to defend against membership inference attacks in MLaaS. The key idea of MIASec is to narrow the dynamic ranges of vital features in the training data, such that the training data, testing data, and even the synthetic data have almost semblable prediction results by the same machine learning model. With elaborated design on modifying the values of vital features in the training data, MIASec can thus reduce the difference between the model's outcomes of training data and testing data, thereby protecting the training data in effect while keeping the model's accuracy stable. We empirically evaluate MIASec on machine learning models trained by off-line neural networks and on-line MLaaS. Using the realistic data and classification tasks, our experimental results show that MIASec can defend the membership inference attacks effectively. In particular, MIASec can reduce the precision and recall of attacks respectively by 11.7 and 15.4 percent in average, and by 18.6 and 21.8 percent at best.

The remainder of this paper is as follows. Section 2 provides some preliminary knowledge. Section 3 describes the design of MIASec, followed by the performance evaluation in Section 4. Section 5 presents related work and finally Section 6 concludes the paper.

2 Preliminary

2.1 Machine Learning as a Service

The increasing demand for machine learning services leverages the emergence of MLaaS, which has been around for some time on the major Internet companies' cloud platforms. Typical MLaaS platforms include Google Prediction API,[1] Amazon Machine Learning (Amazon ML)[2], and Microsoft Azure Machine Learning (Azure ML).[3] MLaaS takes the shape of services in the cloud with automatic learning tools, with various MLaaS options that can be used solely in the cloud or in a hybrid fashion, depending on the company's preferences. Also, adaptability can be simply implemented on MLaaS, making them tailored to the learner's ability level to establish personalized learning without fully learning the complicated algorithms.

In general, the MLaaS platforms provide APIs for users to upload their data for model training and further to make use of the model in various applications. Some platforms even allow users to share the trained model with others via the platform's API for profit.[4] Though convenient, MLaaS is as some kind of a black-box process as the training algorithms are hidden from users. The type of the model may be chosen by the platforms adaptively, but the users are not warned about the consequences of over-fitting. This leads to the vulnerability captured by the membership inference attacks in MLaaS.

2.2 Membership Inference Attacks in MLaaS

Membership inference attacks in MLaaS was first proposed by Shokri et al. in [4]. In a membership inference attack, an attacker is given black-box access to a target classifier $C(T)$ and tends to infer whether a particular record t is included in the training set T or not.

The key idea of membership inference attack is that the machine learning models often have different prediction behaviours on the data that they were trained on versus the data that they "meet" for the first time. These different prediction behaviours are reflected in the target model's outputs and are taken advantage of by Shokri et al. to perform membership inference attacks using a binary classifier. To derive the data for training the classifier, they rely on shadow models to mimic the prediction behaviours of the target model. Due to the similarity, the prediction differences of shadow model also exist in the target model; thus the attackers can leverage the shadow model to find out the prediction differences of the training data and testing data of the target model.

Specifically, on the data sampled from T, the attacker trains multiple "shadow models" using the same MLaaS platform as the target model, and then queries each shadow model with two sets of records: the training set of the shadow model and a disjoint testing set. Since the shadow model is constructed by the attackers, they can exactly know the ground truth of the membership of the shadow model's training set. The prediction outputs of the shadow models are labeled with "in" (reps. "out") to identify the membership of the shadow model's training (reps. testing) data. Using the constructed dataset by shadow models, the attacker trains a neural network as "attack" classifier and uses it to infer whether the target record t is in the training data of the target model or not, given black-box access to the classifier $C(T)$.

When the training of "attack" model is completed, the attacker queries the target model with a record and the prediction result is classified by the "attack" model to infer whether the given record is in the training set of target model or not.

2.3 K-Means Algorithm

Modern machine learning models have great abilities for "memorizing" the important information of input data [8]. To protect the sensitive information of training data, an effective method is to cluster the data and replace the raw data with cluster centroids. Thus the detailed information of every record are hidden, which increases the difficulty for attackers to infer the member of training data.

K-Means [9] technique is widely used in identifying some inherent relationship presented in a set of objects. The purpose of K-Means cluster analysis is to classify raw data into subsets with some meaning in the context of a particular problem. Specifically, in clustering, a set of patterns, usually vectors in a multi-dimensional space, are grouped into clusters in such a way that patterns in the same cluster are similar in some sense while patterns in different clusters are not.

K-Means algorithm often consists of the following steps:

Step 1. Choosing K initial cluster centroids $z_1, z_2 \cdots z_K$ randomly from the input data.

Step 2. Assigning every record in the input data to a certain cluster $C_i, i \in 1, 2, 3 \cdots K$ if the distance between the record and the centroid of C_i is the smallest.

1. https://cloud.google.com/prediction
2. https://aws.amazon.com/machine-learning
3. https://studio.azureml.net
4. https://cloud.google.com/prediction/docs/gallery

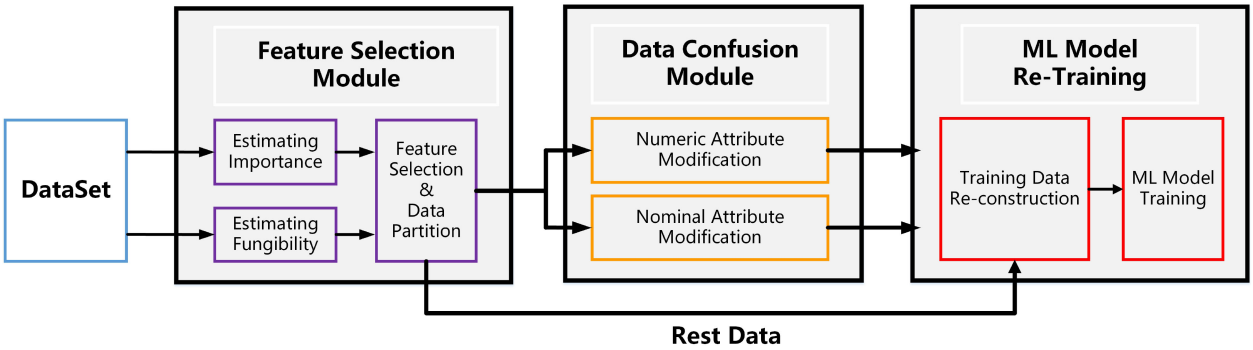


Fig. 1. System structure of MIASec.

Step 3. Calculating the total aggregation of distances of the every record to its cluster centroid, which is used to evaluate the performance of cluster.

Step 4. Updating the centroids $z_1, z_2 \cdots z_K$ by averaging the records in every cluster.

Step 5. Repeating *Steps 2∼4* until the cluster centroids no longer change.

Although K-Means performs well in unsupervised clustering, it requires a user to specify a vital parameter: the number of clusters K, which has a great impact on the clustering performance. In this paper, we propose a dynamic K selection algorithm to cluster the raw data in order to hide the sensitive information so that the input data indistinguishability can be achieved to defend against membership inference attacks in MLaaS.

3 MIASec Design

The basic idea of MIASec is straightforward: by modifying the values of vital features in the training data, MIASec can reduce the differences between the model's outcomes of training data and testing data. In this way, MIASec poses resistance to attackers to infer the membership of the training set by distinguishing the model's outcomes, thereby protecting the training data in effect. To this end, MIASec mainly consists of the following three steps (cf. Fig. 1).

(a) *Searching for appropriate features.* The first step of MIASec is to select a set of features which have prominent impacts on the predicting accuracy of a machine learning model trained on the same dataset. On top of that, MIASec identifies those vital features by comparing the information entropies with each other. For clarity, we denote $F = [f_1, f_2, \ldots, f_n]$ as the selected features, where f_i is the ith selected feature in F.

(b) *Confusing the data of selected features.* For feature f_i in F, it is not sufficient to to protect the data effectively by modifying the values of f_i with a fixed method or noising the values with Laplace noise which is widely used in differential privacy. In MIASec, we proposed a data confusion method based on K-Means algorithm, such that differences between the training data and testing data are narrowed.

(c) *Re-training the machine learning model.* In the third step, we re-train the machine learning model using our confused data obtained in the second step. With the re-trained model, we can reduce the performance difference of the new model among the training data and test data; thus the new model is able to increase the difficulty for attackers to infer the membership of the training set.

3.1 Feature Selection

Modifying all features of a record in the dataset, the valid information carried by the data will be reduced dramatically, and thus the machine learning model trained using the totally confused data is more likely to perform worse than the original model. To guarantee the usability for machine learning model while increasing the difficulty for attackers, MIASec selects a part of features which are vital and irreplaceable to the machine learning model, and then modifies the values of these features. We introduce two metrics to quantify the importance and fungibility of a certain feature: one is the predicting accuracy of machine learning model, and the other is the cross entropy of this feature.

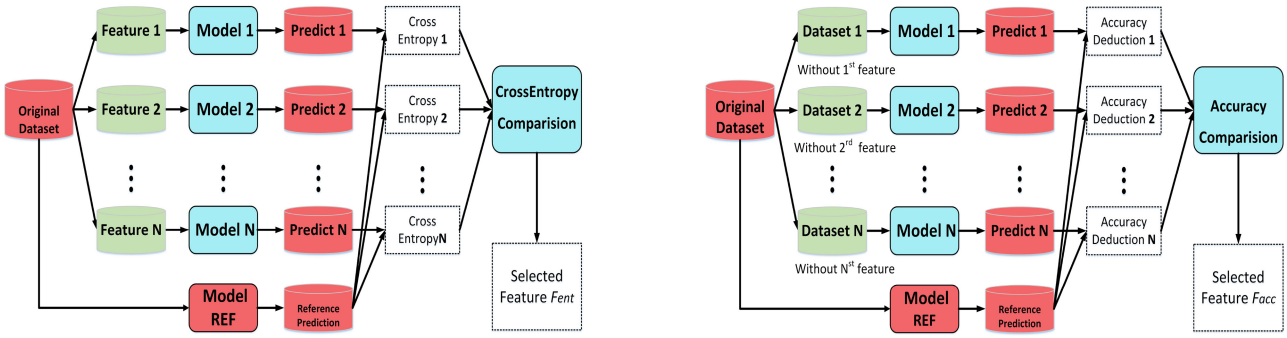
Next, we elaborate the cross entropy corresponding to each feature in the training data. Cross entropy measures the performance of a classification model and is denoted as follows:

$$H(y, \hat{y}) = -\sum_{c=1}^{C}\sum_{i=1}^{M} y_i^{(c)} log\left(\hat{y}_i^{(c)}\right), \tag{1}$$

where y indicates which class the record belongs to, and $\hat{y}$ shows the probability that the record belongs to a certain class. C is the number of classes and M is the number of records in the dataset.

Fig. 2a shows the first step of feature selection of MIASec. We first pick out the ith feature (we assume that the original dataset has N features and $i \in [1, 2 \cdots N]$, and denote the values of the picked feature as $Feature_i$). Then we train a set of models on different $Feature_i$ using the same algorithm and training settings of the reference model $Model_{REF}$ (shown as the red rectangle in Fig. 2a), and denote these models as $Model_i$. Next we calculate the standard cross-entropy between the predictions of $Model_{REF}$ and $Model_i$. The value of cross-entropy can quantitatively indicate the impact of $Feature_i$ on the predicting results. The higher the cross entropy is, the less important the corresponding feature is. At the end of this step, we choose the features with the smallest cross-entropies as the Select Feature F_{ent}.

However, it is not sufficient to select features just by comparing how much one single feature contributes to the predict results. In some cases, one feature may have huge



(a) Crutial Feature Selection (b) Irreplaceable Feature Selection

Fig. 2. Feature selection of MIASec.

impacts on the predicting performance of the machine learning model, but the rest features can be combined or constructed to counteract the performance loss caused by lacking one important feature. For example, although the "Avenue Category" attribute in the Location dataset is significant to our classifier of habit of users, we can leverage the construction of users' coordinates and Avenue locations to replace "Avenue Category" while the accuracy of our classifier is the same as before.

To select the features which cannot be replaced by others, we measure the fungibility of each feature by analyzing the deduction of machine learning model accuracy without this feature. Fig. 2b shows the second step of feature selection of MIASec. MIASec removes the ith feature of the original data , and get the sub-dataset which we denote as $Dataset_i$. Similar with the first step, MIASec trains a reference model on the original dataset and trains a series of models on different $Dataset_i$ alternatively using the same training settings of the reference model. Then we leverage the standard accuracy metric to evaluate the fungibility of each feature. We analyze the accuracy of $Model_i$ and reference model, and obtain the accuracy deduction of machine learning model accuracy without ith feature. At the end of the second step, we choose the features that have the largest accuracy deduction as the Select Feature F_{acc}.

By now, we get the crucial features F_{ent} and the irreplaceable features F_{acc}. The features that appear in both sets are selected as the final results of our feature selection module.

3.2 Data Confusion

In our adversary model, attackers know the format of the inputs and outputs of the victim model, including their number and the range of values they can take. Attackers need to generate the synthetic data for obtaining the predictions of the victim model. If the predictions of synthetic data are similar with the training data, the attackers can determine that the synthetic data are members of the training data.

The basic idea of our data confusion module is to narrow the range of values of the selected features F properly, such that the dynamic range of the synthetic data is limited, and thereby the victim model always predicts semblable results with the synthetic data. By doing so, the confused data can thus increase the difficulty for attackers to distinguish the training data and non-training data by comparing the predicting results.

In this paper, we propose a machine learning based method that can confuse the training data, while the model can still have a relatively stable accuracy. The type of data can be classed into numeric attributes and nominal attributes. Next, we will describe our methods on different data type separately.

Numeric Attribute. MIASec turns the values of numeric features into segments and leverages the segment indicator to replace the original value. Our method changes the range of values that the selected features into a specific set of labels. Naturally, there are a question: how many segments are suitable for a certain attribute. Too many segments cannot protect the training data effectively, while too little segments will cause severe information loss. To achieve the balance between performance of protection and loss of information, we design a dynamic segment algorithm based on K-Means cluster.

For every numeric feature F_{num} in selected features, we construct the values of F_{num} and the true labels into a new data D_{num}. Then we cluster D_{num} into $C = [2, 3, 4 \cdots\cdots]$ successively using K-Means algorithm and obtain the sequence of cluster losses. For clarity, we denote the loss sequence as L_{km}. To find the proper number of segments, we design a metric as follows:

$$C_{bal} = \arg\min \left| \frac{L_{km}(c)}{L_{km}(2)} + \frac{L_{km}(c+1)}{L_{km}(c)} - 1 \right| \quad s.t. \quad c \in [2, 3, 4 \cdots\cdots], \tag{2}$$

where C_{bal} is the chosen number of segments, and $L_{km}(c)$ is the cluster loss for c clusters.

If $L_{km}(c)$ is close to $L_{km}(c+1)$, while $L_{km}(c)$ is much smaller than $L_{km}(2)$ at the same time, we choose c as the number of the segments empirically. Fig. 3 shows a demo of the selection of C_{bal}. At last, MIASec makes use of the boundaries of C_{bal} classes to serve as the segment points of the feature's values.

Nominal Attribute. For the nominal features, MIASec reduces the label classes that the attributes can take by cluster the values of features according to the probability distributions that different kind of labels belongs to each class. To achieve the balance between the performance of protection and loss of information, we adopt a similar approach to the previous process of numeric attributes.

For every Nominal feature F_{nom} in selected features, we calculate the probability distributions that different kind of labels belongs to each class. We construct the probability

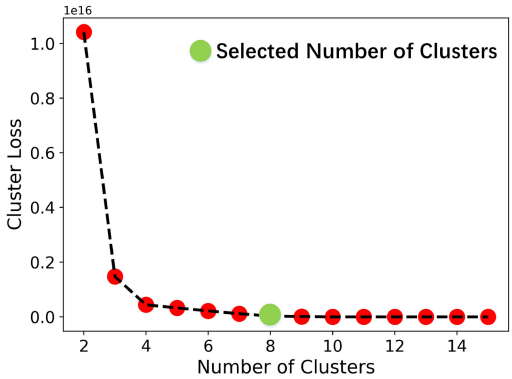


Fig. 3. A example of the number of clusters selection.

distributions of F_{nom} and the true labels into a new data D_{nom}. We cluster D_{nom} into $C = [2, 3, 4 \cdots\cdots]$ successively using standard K-Means algorithm and obtain the sequence of cluster losses. By leveraging Equ. (2), MIASec finds the proper number of clusters which can balance the performance of protection and information loss. At the end of this module, the indicators of the K-Means' clusters serve as the new labels to replace the label values of F_{nom}. Fig. 4 shows an example of the processing of labels clustering and re-grouping.

3.3 Model Re-Training

Now that we have got the confused data, we next need to re-train the machine learning model to make sure that our data plays a role in protection against membership inference attacks. Since the value's ranges of the selected features are restrained by MIASec, the changes of re-trained model's predicting results are also narrowed as the values of confused data vary. This makes attackers hard to distinguish the training data by leveraging the differences in model's predictions.

We first describe how to re-train the machine learning model with our confused data on local neural networks. We build our original model with the help of Theano, which provides the whole structure of neural networks. Then the original data are imputed into the model to train it. We adjust the parameters of our network to increase the prediction accuracy of this model, including the number of layers, the number of nodes, learning rate, and so on. Every time the parameters are changed, we need to train the model with original again until we get an acceptable performance of prediction. After processing original data with MIASec, we replace the original input data with our confused data while keeping all the parameters unchanged, and get the re-trained model.

The procedure for re-training models with MLaaS is much easier than the local neural networks. The MLaaS platforms only provide the interfaces for uploading the training data, and for training and querying models. The details of the models and the training algorithms are hidden from the data owners. Thus in this step, we only need to re-upload the confused data to the MLaaS platforms to obtain the re-trained machine learning models.

4 Performance Evaluation

In this section, we first introduce the datasets used for evaluation, followed by the description of our experimental setups. Then we show the performance of MIASec in different datasets and machine learning platforms.

Original Label	Predict Class 1	Predict Class 2	Predict Class 3	
Coffee	0.7	0.2	0.1	Label: 1
Milk	0.8	0.1	0.1	
Cola	0.3	0.4	0.3	Label: 2
Beer	0.2	0.1	0.7	Label: 3
Wine	0.1	0.1	0.8	

Fig. 4. A toy demo that shows the processing of labels clustering and re-grouping.

4.1 Dataset Description

UCI Adult (Census Income). This Adult dataset is available on the UCI Machine Learning Respository[5] and contains 48,842 records with 14 attributes such as age, gender, education, marital status, occupation, working hours and native country. The feature set contains both continuous (e.g., age, working hours) and discrete (e.g., education, gender) values. This dataset presents a binary classification task to predict if a person makes over $\$50K$ a year based on the census attributes. We use 10,000 randomly selected records to generate a model whose training set needs to be protected from membership inference attacks.

Purchases. Our purchase dataset comes from Kaggle competition, "Acquire valued shoppers" challenge[6], which provides researchers with shopping histories of several thousand individual customers. The purpose of this competition is to design accurate coupon promotion strategies. The Kaggle's dataset contains the transaction and the offer history records of each customer over a year. Specifically, the transactions include many fields such as product name, store chain, quantity, and data of purchase. The offers include coupon type, category, quantity, company, brand and value of the coupon.

For our experiments, we derived a simplified purchase dataset, where each record consists of 15 features. Each feature corresponds to a certain field in the history of transactions and offers. In order to evaluate the performance of MIASec under different experimental conditions, we cluster the records with a different number of classes $\{2, 5, 10, 30, 50\}$, each class representing a different consumption style. The classification task is to predict the consumption style of a user given the input feature vector. We randomly select 10,000 records from the purchase dataset to train the target model whose training data needs to be protected from membership inference attacks. The rest of the dataset contributes to the testing set and the training set of the attack models.

Location. We design our classification task based on the publicly available set of mobile users' location "check-ins" in the Foursquare social network, restricted to the New York City and collected from April 2012 to February 2013.[7] The

5. http://archive.ics.uci.edu/ml/datasets/Adult
6. https://www.kaggle.com/c/acquire-valued-shoppers-challenge
7. https://sites.google.com/site/yangdingqi/home/foursquare-dataset

check-ins dataset contains 1,083 users, 227,428 check-ins records and 251 location types. We filtered out users with fewer than 150 check-ins and venues with fewer than 150 visits, which left us with 558 users and 120,850 records. In our resulting location dataset, each record has 7 features representing the venue, the category of venue, the coordinates of users, timezone offset in minutes and the check-in time.

The purpose of our classification task is similar to the purchase classification, and it is to predict the geosocial type of an user by giving his or her location record. We first cluster the location dataset into 2 classes, each representing a different geosocial type. Then we use 5,000 randomly selected records to train the model. Same with the purchase dataset, the rest of the location dataset contributes to the testing set and the training sets of attack models.

4.2 Experimental Setup

To evaluate the performances of MIASec, we conduct attack and defense experiments on three types of machine learning models: two models are constructed by MLaaS and one we implemented locally. For all the platforms, we assume the models as black boxes. We do not know the structure of the model they create, nor the values of hyper parameters during the training process.

MLaaS. The first cloud based machine learning service in our study is Amazon Machine Learning. With this service, even an amateur user, who is not familiar with machine learning, can build a machine learning model and get prediction results by simply uploading a dataset. In Amazon ML, the user cannot choose the model types but can modify a few parameters, including *maximum number of passes* over the training data and *L2 regularization amount*. The former determines the number of training epochs and the latter tunes how much regularization is performed on the model parameters in order to avoid overfitting. In our experiments, we used the Amazon ML platform in the default setting, in which the number of epochs is 10, and the L2 is 10^{-6}.

The other cloud service that we used in our experiments is bigML. Unlike the Amazon ML platform, the users of bigML have the ability to manipulate a model's attributes, such as the machine learning model types, thresholds or branch size of decision tree. In our experiments, we built all classification models based on Deep Network classifier, and use the same default parameters of each classifier.

Neural Networks. The Neural Network is a proven and well-known technique that is popular in large-scale machine learning. In our experiments, we use Theano library[8] to build our local neural networks, and then train a series of neural networks corresponding to different datasets. To ensure the accuracy of our experiments, we train a set of neural networks on the same dataset, while keeping the hyper parameters and the activation function of models consistent.

Data Settings. The training set and the test set of each victim model and attack model are randomly selected from the respective datasets. In our assumptions, the attacker cannot obtain any detailed record of victim model's training data, thus there is no overlap between the datasets of victim models and those of attack models[4]. However, the datasets used for different attack models can overlap with each other. Specifically, we set the training set size to 10,000 for the purchase dataset and UCI Adults dataset. As for location dataset, we set it to 5,000 in our experiments.

The key capacity of MIASec is to confuse the values of important features while keeping the prediction accuracy stable. To compare the information leakage with and without MIASec, we train victim models both on original and confused training data of UCI Adults, Purchase, Location datasets respectively.

For the purchase dataset (with $\{2, 5, 10, 30, 50\}$ classes), we built victim models with both original and confused dataset on all platforms, including Amazon, bigML, and local neural networks, thus enabling us to compare the performances of MIASec between different models. As for the UCI Adults dataset, we run the experiments both on the Amazon Machine Learning platform and local neural networks. The experiments on the Location dataset were run both on the bigML platform and local.

In our experiments, for the victim models trained by local neural networks, we train a machine learning model with original data and adjust the parameters of our model to achieve an acceptable prediction accuracy at first. Then we leverage the same set of parameters of neural networks, including the number of layers, the number of nodes, learning rate and epochs, to get the new victim model re-trained with our confused data. For the MLaaS platforms, since the details of the models and the training algorithms are hidden from the data owners, it's impossible for us to adjust the models deeply. Thus we evaluate the performance of MIASec with the default settings of these platforms. We just upload the original data and confused data to these online platforms, and obtain the machine learning models trained by MLaaS automatically.

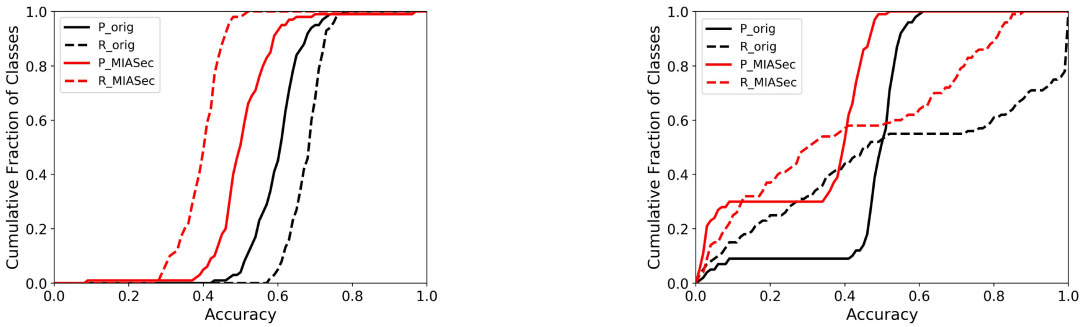
4.3 Performance of MIASec

The purpose of MIASec is to protect the training data of victim models from membership inference attacks. We evaluate the performance of MIASec by comparing the success rates of attack with and without our data confusion approach separately. In our evaluation experiments, we set the number of members equal to the number of non-members, in order to achieve the baseline accuracy to be 0.5.

We evaluate the performance of MIASec using the standard *precision* and *recall* metrics of membership inference attacks. For clarity, we denote P_{orig} and R_{orig} as the precision and recall of attacks using original data respectively, and denote P_{mlsec} and R_{mlsec} as the precision and recall of attacks using confused data with MIASec respectively. We compare the precision and recall variations of attacks to evaluate our algorithm. Specifically in this paper, precision presents the proportion of the records predicted as member of the training data that are indeed members. Recall presents the fraction of the training records that the attackers can correctly infer as members. In other words, recall measures the coverage of the attacks. A small recall represents that attackers can hardly detect the members of training set from the test set, and MIASec could protect the training data of an victim model pretty well. In the following part, we will show the effects of MIASec on defending membership inference attacks with different datasets.

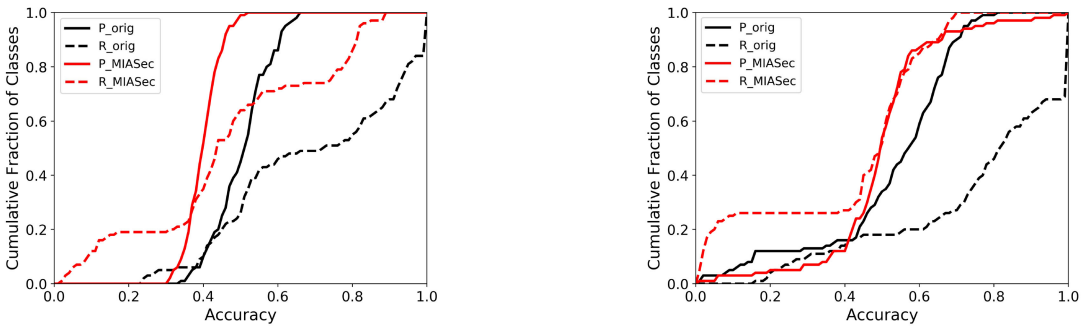
For the UCI Adults dataset, we evaluated our protection methods on victim models trained using Amazon Machine

8. http://deeplearning.net/software/theano/



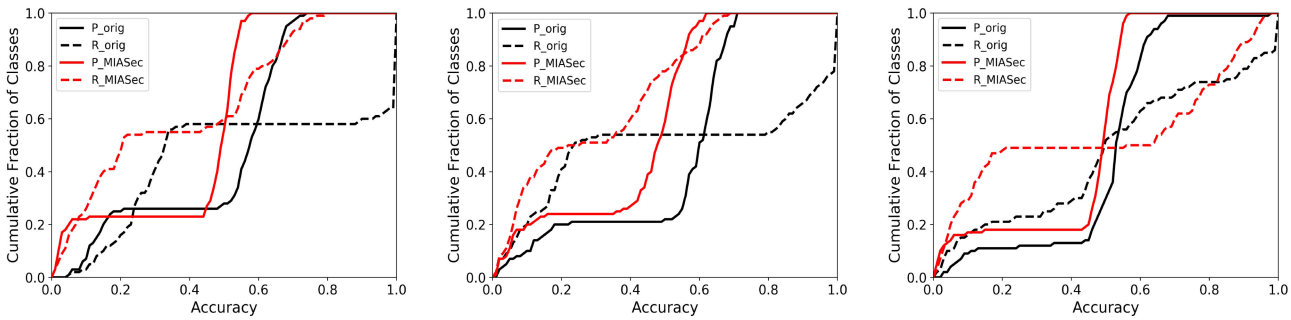
(a) UCI Adults, Amazon ML (b) UCI Atults, Neural Networks

Fig. 5. Empirical CDF of precision and recall of the membership inference attack on different machine learning platforms.



(a) Location, BigML (b) Location, Neural Networks

Fig. 6. Empirical CDF of precision and recall of the membership inference attack on different machine learning platforms.



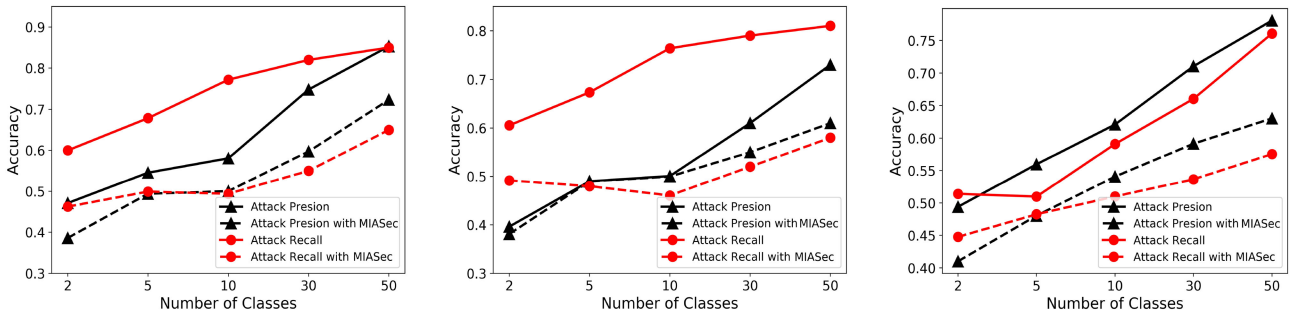
(a) Purchase(2), Amazon ML (b) Purchase(2), BigML (c) Purchase(2), Neural NEtworks

Fig. 7. Empirical CDF of precision and recall of the membership inference attack on different machine learning platforms.

Learning and local neural networks. Fig. 5 shows empirical CDF the precision and recall of the membership inference attacks with the effects of MIASec. From the Fig. 5a we can see that with our confused training data, the mean precision of Amazon machine learning model was reduced by 12 percent comparing with the model trained using original data whose attack precision is about 0.592. The mean recall decreases from 0.681 to 0.396 with the impact of MIASec. As for the local models, Fig. 5b shows that the precision and recall decreases 9.4 and 13.1 percent under the influence of our confusion methods.

Fig. 6 shows the impacts of MIASec on membership inference attacks among the victim models trained by bigML platform and local neural networks. The precision of most models is around 0.5, which is close to the baseline accuracy. Nevertheless, the models trained using MIASec process have a significant reduction of recall which is 11.7 percent. For the models trained by neural networks, most recalls are close to 1.0, while by using the confused data, the recalls decrease over 20 percent remarkably.

For the Purchase dataset, we evaluated the performance of MIASec on both MLaaS and local platforms. Fig. 7 shows the empirical CDF of attack precision and recall with and without MIASec confusion process respectively. To be consistent with experiments on other datasets, we cluster the purchase data into 2 classes and then compare the precisions and recalls between using confused data and using original data. The experiment results show that the precision decreases from 0.497 to 0.371, and the recall decreases from 0.613 to 0.468 under the impacts of MIASec. Fig. 7b shows the impact of MIASec on models trained by Amazon ML. The experiment results show that the precision decreases only



(a) Purchase, Amazon ML (b) Purchase, BigML (c) Purchase, Neural NEtworks

Fig. 8. Precision of the membership inference attack against different purchase classification models trained on Amazon, BigML, and neural networks.

by 0.01, while the recall decreases from 0.624 to 0.487. Fig. 7c shows empirical CDF of the precision and recall of different bigML models. The mean precision and recall of bigML models decrease by 8.7$ and 7 percent respectively.

4.4 Effect of the Number of Classes

In our adversary model, the number of output classes of the victim model affects the extent of the model leakages. The more classes, the more detailed prediction results are available to the attackers.

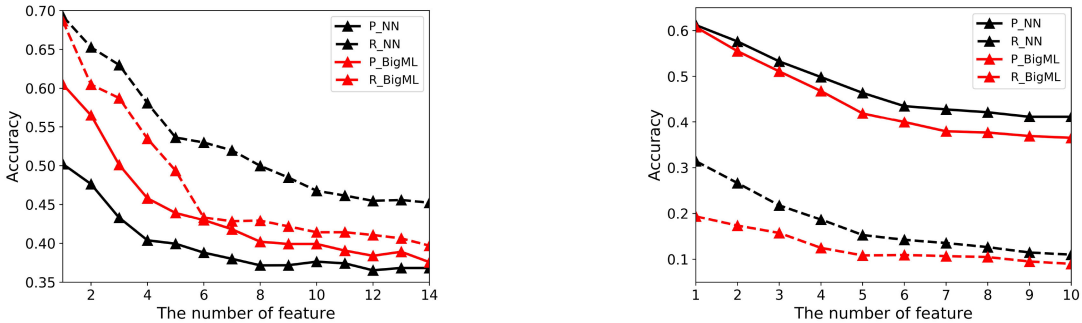
To evaluate the effect that the number of classes has on the performance of MIASec, we train a series of victim models using Amazon ML on the purchase dataset with $\{2, 5, 10, 30, 50\}$ classes. Fig. 8 presents the relationship between the number of classes and the precision of membership inference attack, and meanwhile shows the relationship between class numbers and the attack recall. From the results we can see that as the number of classes increases, so does the precision and recall of attacks no matter the training data is confused or not. In a multi-classification issue, the models need to extract more distinctive features from training data to be able to classify input records with a high accuracy. Informally, the training data will match the models well, thus prediction outputs can vary a lot between the input records from training set and test set.

With MIASec, the precision and recall of membership inference decrease remarkably as show in Fig. 8. The precision and recall of multiple classification reduce respectively by 11.4 and 16.7 percent in average. In extreme cases, the precision can reduce from 0.779 to 0.593, and the recall can decrease from 0.751 to 0.533. Our data confusion approach can confuse the values of selected features which contributes to prediction accuracy of victim models enormously. After confusing the dataset with MIASec, the value range of an important feature is narrower than the original range, that is to say, the value differences between the training data and the test data become smaller than before. The performance of test data on a machine learning model becomes much closer to the training data.

4.5 Effect of the Number of Selected Features

To evaluate the effects of the number of selected features, we train a series of victim models using local neural network and BigML on two datasets. For each dataset, we first sort the feature order according to the standard cross-entropy values and accuracy deductions as discussed in Section 3. Then we gradually increase the number of selected features, choose the top few features as the selected features, and confuse their values. We test the performance of inference attacks on the models trained on the corresponding confused data. The experiment results are shown in Fig. 9.

From the results we can find that for the UCI Adults dataset, when the number of selected features is less than 6, the precision and recall of inference attacks are decreasing dramatically with the feature number growing. Afterward, the performance of the inference attacks declines slowly for both the local neural networks and BigML models. The similar deduction tendency also can be observed in the results of Purchase dataset in Fig. 9b: when the number of selected



(a) UCI Adults, Neural Networks and BigML (b) Purchase(2), Neural Networks and BigML

Fig. 9. The relation between the number of features and the performance of MIASec.

TABLE 1
The Training and Testing Accuracy of Different Dataset on Amazon ML Platform

Dataset	Without MIASec		With MIASec	
	Training Accuracy	Testing Accuracy	Training Accuracy	Testing Accuracy
UCI Adults	0.869	0.860	0.868	0.857
Location	0.958	0.957	0.944	0.940
Purchase(2)	1.000	0.999	0.993	0.989
Purchase(5)	0.947	0.940	0.925	0.918
Purchase(10)	0.885	0.862	0.831	0.796
Purchase(30)	0.816	0.752	0.763	0.687
Purchase(50)	0.764	0.609	0.714	0.513

TABLE 2
The Training and Testing Accuracy of Different Dataset on BigML Platform

Dataset	Without MIASec		With MIASec	
	Training Accuracy	Testing Accuracy	Training Accuracy	Testing Accuracy
UCI Adults	0.976	0.938	0.977	0.942
Purchase(2)	1.000	0.980	1.000	0.923
Purchase(5)	0.999	0.887	1.000	0.778
Purchase(10)	0.998	0.783	0.997	0.674
Purchase(30)	0.926	0.718	0.927	0.581
Purchase(50)	0.873	0.658	0.874	0.554

features is smaller than 7, the values of attack precision and recall decline fast. After that, the performance degradation of inference attacks gradually become slow, and the rates of attack precision hardly decrease when the feature number increases from 8 to 10.

The reason for such decline tendency is that the first few selected features contribute the most information of the data which is highly relative to the classification results. The model trained with the confused data will output similar prediction on different data records, and thus the attackers can hardly determine whether a given record is in the victim model's training set or not. Since the later chosen features carry little information relative to the classification task, confusing the values of these features will have a slight impact on the prediction outputs of the victim model.

4.6 Prediction Accuracy with MIASec

MIASec modifies the values of some vital features in the data, and the performance of models trained on modified data must be different from that on the original data. To quantify the effect that MIASec has on the prediction accuracy of machine learning models, we leverage the original data and confused data respectively to train models on all machine learning platforms. Meanwhile we guarantee that all the models have the same parameters.

Table 1 shows the training and testing accuracies of models constructed using local neural networks for different datasets. As can be seen from Table 1, for the binary classification issues, the model trained using the data that is preprocessed by MIASec has a similar testing accuracy with the model trained with original data. Our protection approach causes 2 percent degradation of testing accuracy for the UCI Adults dataset. For the other datasets, the decline of testing accuracy is less than 1 percent which is acceptable in practical. With the number of classes increasing, the impacts of MIASec are growing gradually. When the number of class goes up to 50, the degradation of testing accuracy reaches its maximum which is 6 percent.

Tables 2 and 3 show the accuracy of models trained by different cloud machine learning services. We can get the similar conclusion from these two tables as above: the more classes the dataset has, the severer the impact of MIASec is. However, there is still an important difference between the models trained by cloud machine learning services and local algorithms. The testing accuracy of cloud-based models decreases much more severely when dealing with multi-classification issues as local models. As the number of classes increases, the generalization ability of the MLaaS models is relatively poorer than the local models. The degradation of testing accuracy even reaches a value of 20 percent when the classes of purchase dataset goes to 50.

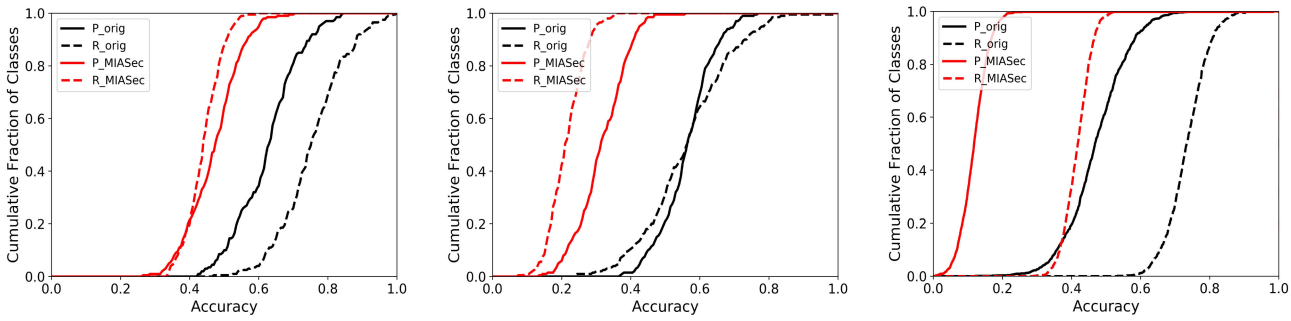
There could be two reasons for why MLaaS impacts machine learning models more and more severely with the number of classes increasing. First, the amount of training data is not sufficient to train an accurate machine learning model. Second, the data confused by MIASec lose a part of valid information. In MIASec, we use the cluster centers to replace the original value of the selected features, and the number of clusters of feature values is aways less than the number of classes. That means the machine learning model cannot obtain adequate information to construct a precise model. To overcome this drawback, we can turn one multi-classification problem into a combination of several binary classification problems. As for why local models perform better compared with cloud-based models, the reason could be that the local models are not overfitted. The training and testing accuracies of model trained using MLaaS have a large difference.

4.7 Performance of MIASec on Other Models

In this section, we evaluate the performance of MIASec on Random Forest, Xgboost and Support Vector Machine (SVM) models. We compare the success rates of membership inference attack against the models trained on the data with and without using MIASec. In our experiments, we set

TABLE 3
The Training and Testing Accuracy of Different Dataset on Local Neural Networks

Dataset	Without MIASec		With MIASec	
	Training Accuracy	Testing Accuracy	Training Accuracy	Testing Accuracy
UCI Adults	0.869	0.860	0.868	0.857
Location	0.958	0.957	0.944	0.940
Purchase(2)	1.000	0.999	0.993	0.989
Purchase(5)	0.947	0.940	0.925	0.918
Purchase(10)	0.885	0.862	0.831	0.796
Purchase(30)	0.816	0.752	0.763	0.687
Purchase(50)	0.764	0.609	0.714	0.513



(a) UCI Adults, Random Forest (b) UCI Adults, Random Forest (c) UCI Adults, SVM

Fig. 10. Empirical CDF of precision and recall of the membership inference attack on different machine learning models.

the number of members equal to the number of non-number to guarantee the baseline of the attack accuracy to be 0.5 which is similar with the settings in Section 3.2. For clarity, we still use P_{orig} and R_{orig} to represent the precision and recall rates of inference attack against the model trained on the original data, and use P_{MIASec} and R_{MIASec} to represent the precision and recall corresponding to the models built with confused data.

From the results in Fig. 10, we can observe that, for the Random Forest models, MIASec can reduce the precision of membership inference attacks by 16.4 percent while the attack against the Random Forest models trained on original data can achieve a mean precision of 61.7 percent. The mean recall of inference attack decreases from 71.9 to 43.7 percent with the protection of MIASec. As for the Xgboost models, Fig. 10b shows that the mean rates of attack precision and recall decrease 24.1 and 27.3 percent respectively with the effects of MIASec. MIASec achieves the best performance of protection on the victim models constructed with SVM algorithm. Specifically with MIASec, the mean attack precision and recall are reduced by 32.7 and 37.6 percent respectively.

Besides, our experiments also show that MIASec can successfully defend the membership inference attack on the Random Forest, Xgboost and SVM models while not inducing much performance loss of protected models. MIASec results in only 2 percent degradation of Random Forest model's training accuracy, while the decline of testing accuracy of the same model is less than 4 percent. For the Xgboost models, MIASec only involves less than 2 percent of accuracy degradation of both training and testing set. Nevertheless, MIASec caused a large degradation of the training and testing accuracies of the SVM model which are around 9 percent. The reason is that the Xgboost and Random Forest models have more powerful learning capabilities than SVM models. Even the training data is modified, the Random Forest and Xgboost models can achieve high training and testing accuracies which is close to that of the original models without MIASec.

5 Related Work

5.1 Membership Inference Attacks

Membership inference attacks have been successfully launched in different domains. Homer et al. [10] propose the first membership inference attack on genomic data. Thereafter Backes et al. [11] generalize this attack to other types of biomedical data. Lately the aggregate mobility traces are also proved to be vulnerable to membership inference attacks [12], where the attack is modeled as a distinguishability game. They also evaluate different defense mechanisms, e.g., differential privacy, to estimate the membership inference risks.

Membership inference attacks against machine learning models is recently proposed in [4], where the shadow model training is devised, aiming at mimicking the target model's behavior so as to generate training data for the attack model. Salem et al. [5] further point out that, by only relying on the posterior's entropy, one pair of shadow model and attack model is sufficient to perform an effective attack. Following this line, several other studies have been proposed from different perspectives on membership inference attacks against machine learning models [13], [14], [15], [16], [17], [18].

5.2 Attacks on Machine Learning Models

Besides membership inference attacks, an attacker with some background knowledge can launch other types of attacks, which try to infer different information of the machine learning models. Fredrikson et al. [19] present the model inversion attacks (i.e., to infer the missing attributes of the victim) in biomedical data setting. This kind of attacks is later generalized to a broader scenario in e.g., face recognition [20]. In [21], Tramer et al. propose model stealing attacks against machine learning models, aiming at stealing the model's learned parameters. One defense method against model stealing attacks was proposed recently by Juuti et al. [22].

Recent studies on adversarial examples [23], [24], [25], [26] show cases in which attackers successfully fool a trained machine learning model to misclassify the data by adding a small amount of noise to the data. This may bring about severe risks in many applications such as autonomous driving, face/voice recognition, et al. Meanwhile, adversarial examples can also contribute to help protect users' privacy in online social networks [27], [28], [29].

5.3 Privacy-Preserving Machine Learning

Another relevant line of work is privacy preserving machine learning techniques. Mohassel et al. [30] present efficient protocols for training linear regression, logistic regression, neural networks in a privacy-preserving manner. Bonawitz et al. [31] propose a multi-party computation based protocol for secure aggregation over high-dimensional data for

distributed machine learning. Homomorphic encryption is also employed to guarantee both input and output private for three machine learning classifiers, namely hyperplane decision, Naive Bayes and decision trees [32]. On this basis, privacy-preserving Random Forest classifier is further devised for medical diagnosis [33]. Besides, there are some other recent studies on security and privacy in machine learning, e.g., [34], [35], [36], [37].

6 Conclusion

In this paper, we proposed MIASec which can confuse the important features of the training data, while keeping the model's accuracy stable. Through narrowing the value ranges of the data's sensitive features, MIASec can decrease the differences of a machine learning model's output between different data records. The smaller prediction differences among different data, the less information we can obtain from the prediction results. Thus MIASec can increases the difficulty for attackers to infer whether a record is in the training data or not. We empirically evaluate MIASec on machine learning models trained by local neural networks and MLaaS. Using real-world data and classification tasks, the results show that MIASec can defend the membership inference attacks effectively. Besides, even these inference attacks adopts different strategies, MIASec could effectively reduce the inference precisions of these attacks as long as these attackers need to find the dissimilarities between the training set and testing set's prediction results.

As for the impact of MIASec on the prediction accuracy, the accuracies of machine learning models trained with raw data and MIASec confused data are really similar to each other when the data has less classes. In an extreme case, the accuracies between original and re-trained machine learning models only have a difference of 0.1 percent. With the number of classes increasing, the impact of MIASec is also expanding while the accuracy loss is still acceptable.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 61872416, 61671216, 61871436, 51479159, 51879210, 51879210, 61702204 and 61872415; by the Fundamental Research Funds for the Central Universities of China under Grant 2019kfyXJJS017; by the Natural Science Foundation of Suzhou/Jiangsu Province under Grant BK20160385; and by the fund of Hubei Key Laboratory of Transportation Internet of Things under Grant 2018IOT004.

References

[1] Y. Wu, F. Hu, G. Min, and A. Y. Zomaya, *Big Data and Computational Intelligence in Networking*. Boca Raton, FL, USA: CRC Press, 2017.

[2] H. Huang, H. Yin, G. Min, H. Jiang, J. Zhang, and Y. Wu, "Data-driven information plane in software-defined networking," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 218–224, Jun. 2017.

[3] H. Huang, H. Yin, G. Min, J. Zhang, Y. Wu, and X. Zhang, "Energy-aware dual-path geographic routing to bypass routing holes in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 6, pp. 1339–1352, Jun. 2018.

[4] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Security Privacy*, 2017, pp. 3–18.

[5] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," in *Proc. Netw. Distrib. Syst. Security Symp.*, 2019.

[6] C. Dwork, F. Mcsherry, and K. Nissim, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory Cryptography Conf.*, 2006, pp. 265–284.

[7] F. Mcsherry and K. Talwar, "Mechanism design via differential privacy," in *Proc. 48th Annu. IEEE Symp. Foundations Comput. Sci.*, 2007, pp. 94–103.

[8] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *CoRR*, arXiv: 1611.03530, 2016.

[9] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[10] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, and D. W. Craig, "Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays," *PLoS Genet*, vol. 4, no. 8, 2008, Art. no. e1000167.

[11] M. Backes, P. Berrang, M. Humbert, and P. Manoharan, "Membership privacy in MicroRNA-based studies," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2016, pp. 319–330.

[12] A. Pyrgelis, C. Troncoso, and E. De Cristofaro, "Knock knock, who's there? membership inference on aggregate location data," in *Proc. Netw. Distrib. Syst. Security Symp.*, 2018, pp. 1–15.

[13] Y. Long, V. Bindschaedler, and C. A. Gunter, "Towards measuring membership privacy," *CoRR*, arXiv: 1712.09136, 2017.

[14] S. Yeom, M. Fredrikson, and S. Jha, "The unintended consequences of overfitting: Training data inference attacks," *CoRR*, arXiv: 1709.01604, 2017.

[15] Y. Long, V. Bindschaedler, L. Wang, D. Bu, X. Wang, H. Tang, C. A. Gunter, and K. Chen, "Understanding membership inferences on well-generalized learning models," *CoRR*, arXiv: 1802.04889, 2018.

[16] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei, "Towards demystifying membership inference attacks," *CoRR*, arXiv: 1807.09173, 2018.

[17] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *Proc. IEEE Symp. Security Privacy*, 2019, pp. 1–15.

[18] G. Liu, C. Wang, K. Peng, H. Huang, Y. Li, and W. Cheng, "Socinf: Membership inference attacks on social media health data with machine learning," *IEEE Trans. Comput. Social Syst.*, to be published, doi: 10.1109/TCSS.2019.2916086.

[19] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *Proc. USENIX Conf. Security Symp.*, 2014, pp. 17–32.

[20] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Security*, 2015, pp. 1322–1333.

[21] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction APIs," in *Proc. USENIX Conf. Security Symp.*, 2016, pp. 601–618.

[22] M. Juuti, S. Szyller, A. Dmitrenko, S. Marchal, and N. Asokan, "PRADA: Protecting against DNN model stealing attacks," *CoRR*, arXiv: 1805.02628, 2018.

[23] Y. Vorobeychik and B. Li, "Optimal randomized classification in adversarial settings," in *Proc. Int. Conf. Autonomous Agents Multiagent Syst.*, 2014, pp. 485–492.

[24] B. Li and Y. Vorobeychik, "Scalable optimization of randomized operational decisions in adversarial classification settings," in *Proc. 18th Int. Conf. Artif. Intell. Statistics*, 2015, pp. 599–607.

[25] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Security*, 2017, pp. 506–519.

[26] A. Kurakin, D. Boneh, F. Tramèr, I. Goodfellow, N. Papernot, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–20.

[27] J. Jia and N. Z. Gong, "Attriguard: A practical defense against attribute inference attacks via adversarial machine learning," in *Proc. USENIX Security Symp.*, 2018, pp. 513–529.

[28] S. Joon Oh, M. Fritz, and B. Schiele, "Adversarial image perturbation for privacy protection–a game theory perspective," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1491–1500.

[29] Y. Zhang, M. Humbert, T. Rahman, C.-T. Li, J. Pang, and M. Backes, "Tagvisor: A privacy advisor for sharing hashtags," *CoRR*, arXiv: 1802.04122, 2018.
[30] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in *Proc. IEEE Symp. Security Privacy*, 2017, pp. 19–38.
[31] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. IEEE Symp. Security Privacy*, 2017, pp. 1175–1191.
[32] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," in *Proc. Annu. Netw. Distrib. Syst. Security Symp.*, 2015, pp. 4324–4337.
[33] M. Backes, P. Berrang, M. Bieg, R. Eils, C. Herrmann, M. Humbert, and I. Lehmann, "Identifying personal dna methylation profiles by genotype inference," in *Proc. IEEE Symp. Security Privacy*, 2017, pp. 957–976.
[34] C. Song, T. Ristenpart, and V. Shmatikov, "Machine learning models that remember too much," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2017, pp. 587–601.
[35] C. Huang, G. Min, Y. Wu, Y. Ying, K. Pei, and Z. Xiang, "Time series anomaly detection for trustworthy services in cloud computing systems," *IEEE Trans. Big Data*, to be published, doi: 10.1109/TBDATA.2017.2711039.
[36] T. Hunt, C. Song, R. Shokri, V. Shmatikov, and E. Witchel, "Chiron: Privacy-preserving machine learning as a service," *CoRR*, arXiv: 1803.05961, 2018.
[37] Y. Ma, Y. Wu, J. Ge, and L. Jun, "An architecture for accountable anonymous access in the internet-of-things network," *IEEE Access*, vol. 6, pp. 14 451–14 461, 2018.

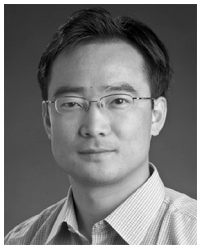
Chen Wang (S'10-M'13-SM'19) received the BS and PhD degrees from the Department of Automation, Wuhan University, China, in 2008 and 2013, respectively. From 2013 to 2017, he was a postdoctoral research fellow with the Networked and Communication Systems Research Lab, Huazhong University of Science and Technology, China. Thereafter, he joined the faculty of the Huazhong University of Science and Technology where he is currently an associate professor. His research interests include the broad areas of wireless networking, Internet of Things, and mobile computing, with a recent focus on privacy issues in wireless and mobile systems. He is a senior member of the IEEE and ACM.

Gaoyang Liu (S'19) received the BS degree in information engineering from the Huazhong University of Science and Technology, China, in 2015. He is currently working toward the PhD degree in electronics and information engineering at the Huazhong University of Science and Technology, China. His research interests include machine learning, mobile sensing, and data privacy protection. He is a student member of IEEE.

Haojun Huang received the BS degree from the School of Computer Science and Technology, Wuhan University of Technology, China, in 2005, and the PhD degree from the School of Communication and Information Engineering, University of Electronic Science and Technology, China, in 2012. He was a post-doctoral researcher with the Research Institute of Information Technology, Tsinghua University, Beijing, from 2012 to 2015, and an assistant professor with Wuhan University, China, from 2015 to 2017. He is currently an associate professor at the Huazhong University of Science and Technology, China. His research interests include wireless networks, big data, and software-defined networking.

Weijie Feng received the BE degree from Xiangtan University, China, in 2016. He is currently working toward the ME degree in electronics and information engineering at the Huazhong University of Science and Technology, China. His research interests include machine learning and Internet of Things.

Kai Peng received the BS, MS, and PhD degrees from the Huazhong University of Science and Technology, China, in 1999, 2002, and 2006, respectively. He is now the faculty of the Huazhong University of Science and Technology as a full professor. His current research interests include the areas of wireless networking and big data processing.



Lizhe Wang (SM'09) received the BE and ME degrees from Tsinghua University, and the doctor of engineering degree from University Karlsruhe (Magna Cum Laude), Germany. He is a "ChuTian" chair professor in the School of Computer Science, China University of Geosciences (CUG), and a professor at the Inst. of Remote Sensing & Digital Earth, Chinese Academy of Sciences (CAS). His main research interests include HPC, e-Science, and remote sensing image processing. He is a fellow of IET and the British Computer Society and senior member of the IEEE.