SocInf: Membership Inference Attacks on Social Media Health Data With Machine Learning

Gaoyang Liu, *Student Member, IEEE*, Chen Wang, *Senior Member, IEEE*, Kai Peng, Haojun Huang, Yutong Li, and Wenqing Cheng, *Member, IEEE*

***Abstract*— Social media networks have shown rapid growth in the past, and massive social data are generated which can reveal behavior or emotion propensities of users. Numerous social researchers leverage machine learning technology to build social media analytic models which can detect the abnormal behaviors or mental illnesses from the social media data effectively. Although the researchers only public the prediction interfaces of the machine learning models, in general, these interfaces may leak information about the individual data records on which the models were trained. Knowing a certain user's social media record was used to train a model can breach user privacy. In this paper, we present SocInf and focus on the fundamental problem known as membership inference. The key idea of SocInf is to construct a mimic model which has a similar prediction behavior with the public model, and then we can disclose the prediction differences between the training and testing data set by abusing the mimic model. With elaborated analytics on the predictions of the mimic model, SocInf can thus infer whether a given record is in the victim model's training set or not. We empirically evaluate the attack performance of SocInf on machine learning models trained by Xgboost, logistics, and online cloud platform. Using the realistic data, the experiment results show that SocInf can achieve an inference accuracy and precision of 73% and 84%, respectively, in average, and of 83% and 91% at best.**

***Index Terms*— Generative adversary network, machine learning, membership inference attack, social media health data.**

I. Introduction

SOCIAL media is defined as web-based and mobile-based Internet applications that allow the creation, access, and exchange of user-generated content that is ubiquitously accessible [1], [2]. In the past few years, social media have shown a rapid growth of user counts and have been the objective of scientific researches. For example, Facebook has more than 2.2 billion monthly active users as of January 2018 [3] and Twitter counts more than 3.36 accounts in total [4]. The advent of social media has dramatically decreased the difficulty in creation and dissemination of people's contents, and as a consequence, more and more data are generated in various aspects of life for researchers to analyze user attitudes and behaviors. The enormous amounts of social data attract the interests of researchers of different disciplines including social scientists, data mining experts, and market researchers to analyze social media data as a source to monitor human emotions and behaviors in the physical world.

Recently, the emerging of machine learning greatly improves the abilities of computer systems to process large-scale data and understand the meanings behind the data [5]. A part of social media researchers turn to leverage machine learning to monitor the abnormal behaviors and extreme emotions of people through the massive data that the researchers collect from social media. Nevertheless, in order to build a machine learning model that can detect a certain abnormal behavior, the builders need to feed the model with a training data set that contains abundant samples of such abnormal behavior. If the social scientists publish their research findings as well as the machine learning model, we can breach the privacy of the training data by abusing the public model. For example, a research group discloses a machine learning model that can predict whether a person has a suicidal tendency [6]. If we know that a certain person's social media data were used to train this model, we can reveal that the person is more likely to suicide.

Knowing the data of a certain user contained in a data set can threaten the privacy of this user severely. Thus, in this paper, we focus on the fundamental problem known as membership inference against social media health data: determining whether a given social media record was used as the training data set of a model of abnormal behavior and emotion detection or not. Since we can hardly get the access to the training set as attackers or select the data which happens to be in the training set from massive social media health data, we investigate this problem in the most difficult scenario: we only have the black-box access to the machine learning model as attackers. There is already a part of researchers who proposed membership inference attack methods on machine learning models. Shokri *et al*. [7] invented the shadow training technique to breach the privacy of the machine learning models' training data set. The shadow training technique trains a series of "shadow" models that imitate the behavior of the victim model. Since the attackers know the ground truth about the membership in "shadow" models' training data,

Manuscript received October 31, 2018; revised March 9, 2019 and April 26, 2019; accepted May 6, 2019. Date of publication June 3, 2019; date of current version October 7, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61872416, Grant 61671216, Grant 61871436, Grant 51479159, Grant 51879210, Grant 51879210, Grant 61702204, and Grant 61872415, in part by the Fundamental Research Funds for the Central Universities of China under Grant 2019kfyXJJS017, and in part by the Fund of Hubei Key Laboratory of Transportation Internet of Things under Grant 2018IOT004. *(Corresponding author: Kai Peng.)*

The authors are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: liugaoyang@hust.edu.cn; chenwang@hust.edu.cn; pkhust@hust.edu.cn; hjhuang@hust.edu.cn; ytli@hust.edu.cn; chengwq@hust.edu.cn).

Digital Object Identifier 10.1109/TCSS.2019.2916086

the attackers then train an inference attack model on the labeled inputs and outputs of the "shadow" models. However, the "shadow" training technique requires the attackers obtain the training algorithm and model structure of the victim model, and the "shadow" models need a large amount of training data that are similar with the training data of the victim model. In practice, it is difficult to get the prior information about the victim model and its training data.

In this paper, we propose SocInf that can breach the membership privacy of a given machine learning model's training data set under a demanding scenario, in which SocInf can get nothing except the prediction results of the victim model on input data. The key observation of SocInf is that the machine learning models often have different prediction behaviors on the data that they were trained on versus the data that they "meet" for the first time. Therefore, SocInf first trains a machine learning model that has a similar prediction performance with the victim model. Due to the similarity, the prediction differences of the mimic model also exist in the victim model, and thus, we can leverage the mimic model to seek the prediction differences of the training data and testing data of the victim model. In this way, the prediction differences of the target model can be disclosed and these differences can be leveraged to infer the membership of the target model's training set.

Although the basic idea sounds straightforward, it is nontrivial to implement SocInf due to several challenges. The first challenge is that we need data sampled from the same data space with the victim model's training data when training the mimic model. We proposed two effective methods to address this challenge. The first method leverages statistics about the features' value distributions of the population from which the victim model's training data set was drawn. The second method is to synthesize the training data randomly based on the input format of the victim model. Although the second method requires to traverse all possible combinations of feature values, it is still exercisable when the data have few attributes and we do not have any prior information about the victim model's training data set.

Another challenge is that it is difficult to train a machine learning model that has similar prediction behavior with the victim model, since we cannot obtain the details of the victim model, including the training data set, training algorithm, and model structure. However, a neural network (NN) which has enough layers could mimic any type of machine learning models [8]. Thus, we build our mimic model based on the NNs no matter what structure the victim model has and then enhance its performance by modifying generative adversary networks [9]. After abundant epochs of contesting with a discrimination NN whose objective is to distinguish the prediction results of the mimic model from those of the victim model, SocInf can thus obtain an ideal mimic model.

We evaluate SocInf against the machine learning models trained using logistic regression [10], Xgboost [11], and a cloud machine learning platform. For our evaluation, we use realistic classification tasks and standard model training procedures on two different social media data set: one contains amounts of tweets and another contains numerous Weibo texts. To demonstrate that the attacks of SocInf are successful, we leverage standard metrics of machine learning to quantify how our attack algorithm performs. For classification models trained on the tweets data set, our membership inference achieves a mean accuracy of 73% and a mean precision of 84%. Our results for Weibo data set indicate that our membership inference attack can achieve a mean accuracy of 73% and a mean precision of 71%. The experiments show that no matter which machine learning algorithm the target model or attack model employs, SocInf has the power to breach the privacy of the victim model's training set.

The main contributions of this paper are as follows.

1) We present SocInf, the first membership inference attack technique against black-box machine learning models with unknown structure and parameters.
2) We propose a universal methodology to synthesize a set of data whose distribution is close to the target data set, without preknowledge of the statistical information about the training data of the target model.
3) We develop an imitation method based on the generative adversary network to mimic the prediction behavior of the target model, only with the synthetic data and the model's prediction application programming interfaces (APIs).
4) We evaluate the attack performance of SocInf against three different machine learning models on realistic Tweet data set and Weibo data set. The experiment results validate the effectiveness and efficiency of SocInf.

The remainder of this paper is as follows. Section II provides some preliminary knowledge. Section III describes the design of SocInf, followed by the performance evaluation in Section IV. Section V proposes several mitigation strategies for SocInf. Section VI presents some related work, and finally, Section VII concludes this paper.

II. Preliminary

A. Social Media Analytics

The presence of online social media contributes to the new dimensions to the production and dissemination of information. Social media enables the creation, entrance, and exchange of user-generated contents such as images, texts, videos, and records for social interactions between different online users with real-time publishing and communicating. Therefore, it replaces the traditional one-way media broadcast to consumer communication. Many Internet companies now offer social media services on their platforms. Examples include Twitter,[1] Facebook,[2] Youtube,[3], and Flickr.[4]

The popularization of social media enriches approaches for obtaining the persona of an individual user through mining

*In our paper, the *target model* and *victim model* have the same meaning, and can be replaced with each other.

[1]https://twitter.com/

[2]https://www.facebook.com/

[3]https://www.youtube.com/

[4]https://www.flickr.com/

and analyzing the social media data. In recent years, many companies focus on leveraging the social media analytics of users for business purposes including accurate advertising [12], custom information streaming [13], or purchase predicting [14]. Some companies also analyze social media to obtain insights in how to improve and promote products better [15]. For consumers, the abundance of information and opinions from different people helps them to make more wisdom and informed strategies. In addition, the customers can receive the customized information streams based on analyze their social media data which are vital in improving the user experience [16].

In addition, social media provides users with support with health information and communication approach between doctors and patients [17], [18]. A large amount of contents released by users through social media contains self-disclosure and other's judgment information, which can serve as valuable resources for online health information and health behavior analysis. By analyzing and processing the social media health data, the researchers can capture the epidemic transmission trends [19], predict the health emergency [20], and even prevent the suicide or risky behaviors of users [6]. To extract the health information behind the social media data, many research groups leverage machine learning technology to predict and prevent abnormal behaviors [6] and emotions [21] in advance. However, the social media models can leak the privacy of their training set through prediction results, and the privacy leakages of the health data have more serious consequences than other data leakages. Thus, in this paper, we focus on breaching the privacy of the machine learning models trained on social media health data.

B. Membership Inference Attacks in Machine Learning

Numerous machine learning algorithms are leveraged in many practical fields, such as medicine, biochemistry, finance, and so on. It is important that ML models trained on sensitive inputs not leak too much about the training data. In a membership inference attack [7], an adversary is given black-box access to a target classifier C_{Target}, and aims to infer whether a given record t is in the training data set or not.

The membership inference attack was first proposed by Shokri *et al.* [7]. The adversary leverages the target model C_{Target} to predict the classification result of a given record and then use the prediction on t to infer the membership status. The authors turn the membership inference problem into a binary classification problem. For each given record, there are two possible classes: class "in" means that the record in the training set, whereas the class "out" has the opposite meaning. To create the training data set for the membership inference classification, the authorship provides "shadow" training techniques. The adversary first synthesizes a series of data records and tags the labels of each record based on the prediction results obtained from C_{Target}. Then, the adversary trains a series of "shadow" models C_{Shadow} using the same machine learning algorithm of C_{Target} on these synthetic data. Then, the adversary queries each shadow model with two sets of records: the training set of the shadow model and a disjoint test set. Finally, the adversary aggregates the prediction results of shadow models and leverages these results to train a membership classifier C_{Attack} which can infer whether the target record t is in the training data set.

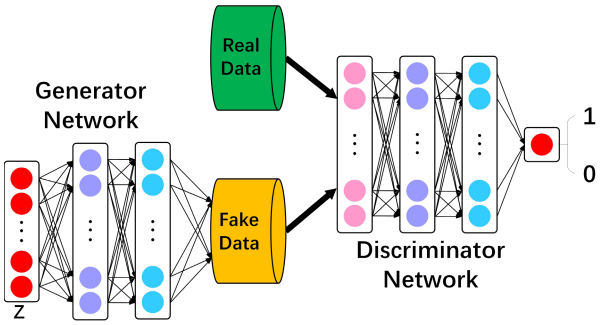


Fig. 1. Structure of GAN.

C. Generative Adversarial Networks

Generative adversarial networks (GAN) are a class of artificial intelligence algorithms used in unsupervised machine learning, implemented by a system of two NNs contesting with each other in a zero-sum game framework [9] (see the contesting procedure of the *Generator* and the *Discriminator* in GAN in Fig. 1). The *Generator* is a NN which is trained to synthesize fake data to "fool" the *Discriminator* network, while the *Discriminator* is a NN trained to discriminate real data samples from synthesized samples.

At the beginning, a known data set serves as the initial training data for *Discriminator* and the training keeps continuing until it reaches some level of accuracy. The *Generator* is seeded with a random noise sample to generate the synthetic data. Thereafter, the synthetic data and real data are evaluated by *Discriminator*. As the contest between the *Generator* and *Discriminator* continues, the *Generator* produces better synthetic data, while the *Discriminator* becomes more skilled at labeling synthetic data. In other words, the *Discriminator* learns the "rules" of the real data.

Since an NN which is deep enough could mimic any type of machine learning models [8], we build a *Generator* with multiple layers to learn the "rules" of prediction behavior of the unknown victim model. Then, the *Generator* serves as the simulated duplication of the victim model. However, we need to modify GAN because it cannot be deployed directly in our works. The details of our modified GAN will be discussed in Section III.

III. SocInf Design

A. Overview of SocInf

The basic idea of SocInf is straightforward. By training a mimic machine learning model which has the similar prediction performance with the victim model, SocInf can leverage the mimic model to seek the prediction differences of the training data and testing data of the target model. In this way, SocInf can infer whether a given record is in the training data of the victim model, thereby breaching the privacy of training

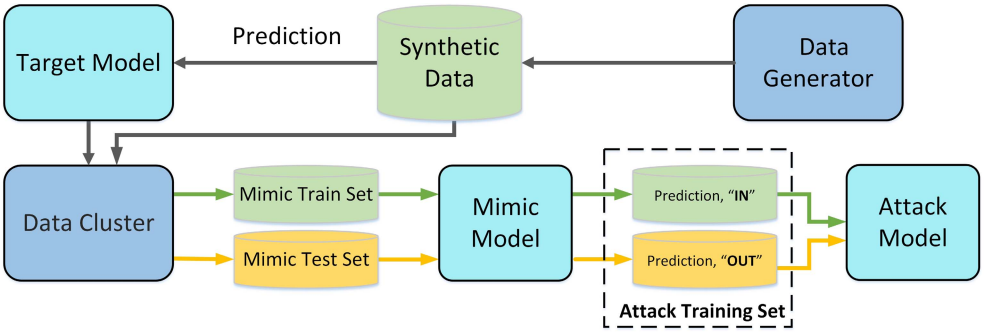


Fig. 2. System structure of SocInf.

data effectively. To this end, SocInf mainly consists of the following four steps (cf. Fig. 2).

1) Generating the Synthetic Data: In order to train the mimic model, the first step of SocInf is to synthesize a set of data which has the same format with the training data of the victim model. Then, we utilize the target model's predicting interface to get the predictions of these synthetic data and classify the synthetic data according to their predicted classes. For clarity, we denote F_{target} as the target model which SocInf is expected to imitate, C as the set of class labels that F_{target} can take, X_s is the synthetic data, and X_s^c is the synthetic data that are predicted belonging to class $c \in C$.

2) Synthesizing Data Cluster: To extract the part of X_s^c that are similar with the training data of the target model, SocInf clusters the data X_s^c into two classes according to their prediction results queried from F_{target} without supervision. Then, we mark the part of data which have a small variance with the label "in," and mark the other part with the label "out." We aggregate the "in" data part of every X_s^c and, thus, get the most similar part of X_s to the training data (denoted as X_s^{in}, and the rest of the synthetic data are denoted as X_s^{out}).

3) Mimicking the Target Model: With the synthetic data X_s^{in} and the prediction interface of F_{target}, SocInf aspires to build a machine learning model that has the almost identical prediction behavior with the victim model. To achieve that, we get the prediction of X_s^{in} on F_{target}, and then leverage the GAN to build and evolve a mimic model such that the outputs of the mimic model are indistinguishable from the outputs of F_{target} on the same data set X_s^{in}. For clarity, we denote the constructed mimic model as F_{mimic}.

4) Training the Attack Model: In this step, we obtain the prediction results of F_{mimic} on its training set and testing set. To be more general, X_s^{in} serves as the training data and X_s^{out} serves as the testing data. Then SocInf leverages the prediction differences between the training set and testing set to train an attack model with supervision. After SocInf trained the attack model, SocInf inputs the given record to F_{target} and gets the prediction result. Then, SocInf utilizes the attack model to infer whether the given record is in the training set of the model F_{target} or not.

For clarity, we summarize the notations in Table I.

TABLE I
SUMMARY OF NOTATIONS

Notation	Description
F_{target}	The target or victim model
F_{mimic}	The mimic model
F_{attack}	The attack model
C	Class labels of F_{target}'s training data
X_t	Target model's training data
X_s	The synthetic data
X_s^c	The synthetic data of certain class c
X_s^{in}	The part of X_s similar with X_t
X_s^{out}	The part of X_s not similar with X_t
Y_s	The predictions of F_{target} on X_s
Y_s^c	The synthetic data of certain class c
Y_s^{in}	The predictions of F_{target} on X_{in}
$\hat{Y}_s^{in}$	The predictions of F_{mimic} on X_{in}
Y_s^{out}	The predictions of F_{target} on X_{out}
D_{real}	Combination of X_s^{in} and Y_s^{in}
D_{fake}	Combination of X_s^{in} and $\hat{Y}_s^{in}$
D_{attack}^{train}	The training data of attack model

B. Synthetic Data Generation

The key idea of SocInf is to train an NN that has similar prediction behavior with the target model F_{target}. To train the mimic model F_{mimic}, we need training data distributed similarly to the target model's training data. We provide two methods for generating synthetic data.

1) Synthesis Based on Statistics: In some scenarios, we may have some statistical information about the distribution of the target model's training data. For example, a model's training data consist of the age information of all data providers in a certain district, while we can get the statistical information about the age information of this district from the population statistics released by the local government, such as the single nucleotide polymorphism database.[5] Then, we can simply synthesize the age values by sampling from the age's distribution. Once we have the marginal distributions of different features, we can generate the synthetic data for F_{mimic} by independently sampling the value of each feature from its own marginal distribution. However, some features are correlated with other features in practice. For example, marital status is highly correlated with the number of children the data provider has.

[5]https://www.ncbi.nlm.nih.gov/SNP/

To overcome the difficulty in acquiring the joint probability distribution of different features, we leverage Markov chain Monte Carlo (MCMC) [22] to obtain the synthetic data by sampling in the data space.

In SocInf, we also need some data that are disjoint with the training data for extracting the prediction performance differences of F_{target}. After we get the sampled data based on the marginal distributions of all the features, we purposely add random noise to the sampled data. If the noised data record comes up for the first time, SocInf accepts this data record to the synthetic data set.

2) Synthesis Based on Data Format: In some scenarios, we cannot get the real training data of the target model nor any statistics about its distribution. The information that we can achieve is the prediction interface of the target model F_{target} and the format of the input data. In this case, the synthesis process runs as the following steps.

First, we select a class c from C which we aspire to generate the synthetic data. Then, we initialize a record randomly and sample the value for each feature uniformly at random from among all possible values of that feature. In every iteration, a new record candidate is proposed by changing several randomly selected features of the last accepted record. To be more specific, SocInf randomly changes the values of selected attribute features or adds random noise to the values of the selected numeric features. Then, the new record candidate will be accepted to the synthetic data set if this record shows up for the first time. At the end, we can obtain the synthetic data X_s.

In order to generate a proper set of synthetic data, SocInf needs to explore the space of all possible inputs. It may not work if the inputs of the target model are images, videos, or sound records. To avoid the huge computation and time costs for traversing the entire data space, we stop the synthesis process once the number of synthetic data is enormous enough. However, the social media emotion data has much fewer features than multimedia data and the costs of traversing the data space are acceptable to SocInf.

C. Synthetic Data Cluster

Since the synthetic data are generated by sampling from the training data space essentially, X_s can cover or even override the data space corresponding to the training data of the target model if we sample sufficient data. X_s mainly consists of two parts: one is of the synthetic records which are very similar to the training data of F_{target}, and the other is of the sampled records which are distinctly different with the target model's training data. To make the imitation performance of F_{mimic} as close as possible to the target model, SocInf needs to extract X_s^{in} from the synthetic data and then train the mimic model with X_s^{in}.

We first leverage the target model F_{target} to obtain the prediction results of X_s which represent the probabilities that the data belong to each class in C, and we denote the results as Y_s. As shown in Fig. 3, we divide X_s and Y_s into $|C|$ parts according to their predicted classes in the following step. For clarity, we denote Y_s^c as the predicted probability results whose class belongs to c.

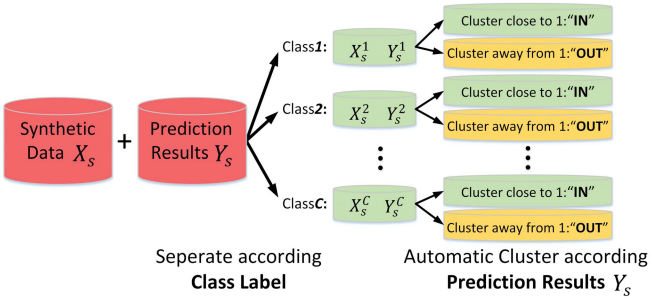


Fig. 3. Process of the synthetic data cluster.

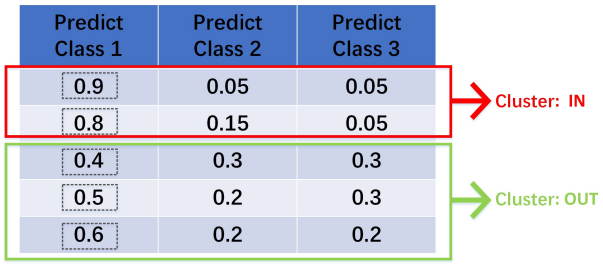


Fig. 4. Toy example of the synthetic data cluster.

The insight here is that if a data record is in the training data of a machine learning model, the model will predict this record to a certain class with a high probability when the model meets the training data again. As shown in Fig. 4, the predicted probabilities of a certain class are very close to 1 if the testing inputs are the training data. For each class in C, we use standard k-Means algorithm [23] to cluster Y_s^c into two clusters. Then, we compare the cluster centroids of Y_s^c and mark the data cluster of Y_s^c whose centroid is close to 1 with the label "in" based on our insight. The other cluster's data are marked with the label "out." The label "in" means that the marked data are in the training data set, while the label "out" has an opposite meaning. We aggregate the "in" part of Y_s^c and corresponding X_s^c of all classes, and then we get the similar part of synthetic data to the target model's training data, X_s^{in} (Y_s^{in} is denoted as the prediction results of model F_{target} on X_s^{in}). We can get the "out" aggregations of X_s^{out} and Y_s^{out} in the same way. Fig. 4 shows a toy example of how SocInf selects the data records similar to the training set of the target model from the synthetic data. In Fig. 4, five synthetic data points are all predicted to belong to **Class** 1 by F_{target}. Then, we cluster the prediction results into two clusters using standard k-Nearest Neighbor (KNN) algorithm. Obviously, the first and second predictions should be divided into one same cluster, and the rest should be grouped into another cluster. Since the centroid of the first cluster is much closer to 1, we mark the first and second data records with the label **"IN"**, while the rest are labeled with **"OUT"**.

D. Mimic Model Construction

Now that we have obtained the synthetic data X_s^{in} which are close to the target model's training data. Next, SocInf needs to build a mimic model which has a similar prediction performance with the target model. However, we know nothing

about the target model's training algorithm and parameters, except the prediction interface. We cannot build a mimic model that is consistent with the target model's structure or settings, and we cannot even train our mimic model by using the same machine learning algorithm. Fortunately, an NN which has enough layers could mimic any type of machine learning models [8]. Thus, we leverage GAN to construct our mimic model.

Specifically, we replace the input noise of GAN with our synthetic data X_s^{in}, and replace the "real data" with the predictions of the target model on the synthetic data. As the contest between the *Generator* and the *Discriminator* going, the *Generator* will learn the prediction rules of the target model gradually. Finally, the *Generator*'s predictions will become almost indistinguishable from the target model's. When the prediction results of F_{target} and Generator on the same data set are difficult to distinguish, SocInf will stop training and take the *Generator* as the mimic model F_{mimic} (because the target model and the mimic model have almost identical prediction behaviors at this time). The process of *Generator* in GAN can be expressed as

$$\hat{Y}_s^{\text{in}} = F_{\text{mimic}}\left(X_s^{\text{in}}\right) \tag{1}$$

where $\hat{Y}_s^{\text{in}}$ is denoted as the prediction results of F_{mimic} on the synthetic data.

In the process of the discriminant model, to provide abundant information to the discriminant model, not only the prediction results Y_s^{in} and $\hat{Y}_s^{\text{in}}$ but also the synthetic data X_s^{in} serve as the inputs of *Discriminator*. We combine X_s^{in} and Y_s^{in} as the real data D_{real}, while X_s^{in} and $\hat{Y}_s^{\text{in}}$ as the fake data D_{fake}. The *Discriminator* is as the following equation:

$$Z = \text{Discriminator}(D_{\text{fake}}, D_{\text{real}}) \tag{2}$$

where Z is the discriminant result of *Discriminator* that shows whether the input data is generated by the mimic model F_{mimic}.

According to (1) and (2), we can get the training procedure of our modified GAN as

$$Z = \text{Discriminator}\left(\left[X_s^{\text{in}}, F_{\text{mimic}}\left(X_s^{\text{in}}\right)\right], \left[X_s^{\text{in}}, F_{\text{target}}\left(X_s^{\text{in}}\right)\right]\right). \tag{3}$$

In each training iteration of our modified GAN, the discriminant model (respectively, the mimic model) improves its prediction performance by leveraging the backward propagation algorithm on (2) [respectively, (3)]. After adequate rounds of training, we finally obtain an expectable model F_{mimic} which has a similar prediction behavior with the target model. For more detailed information about the backward propagation algorithm, please refer to [24].

E. Attack Model Construction

In this section, we describe how to train the attack model elaborately. The key insight of our membership inference attack is that machine learning models often behave differently on the data that they were trained on versus the data that they meet for the first time. The objective of SocInf is to build an attack model that can recognize such differences in the target model's behavior and use them to distinguish members from nonmembers of the target model's training data set based on the target model's prediction outputs.

To construct our attack model, SocInf generates the synthetic data X_s and constructs the mimic model F_{mimic} which has a similar prediction behavior with the target model. In contrast to the target model F_{target}, we know the ground truth of the mimic model whether a given a record was in its training data set or not. Thus, we can leverage supervised training algorithm on the prediction outputs of the mimic model to teach the attack model how to distinguish the mimic model's outputs on members of its training data set from the predictions on nonmembers.

For clarity, we denote F_{attack} as the attack model whose objective is to infer whether a given record is in the training data set of the target model from the corresponding prediction of F_{target}. Since the mimic model F_{mimic} is trained on the synthetic data X_s^{in}, we mark the prediction results Y_s^{in} of the mimic model on X_s^{in} with label "in," while we mark Y_s^{out} with label "out." Then, we combine the labeled prediction results of the mimic model, $(Y_s^{\text{in}}, \textbf{in})$ and $(Y_s^{\text{out}}, \textbf{out})$, to get the attack training data set $D_{\text{attack}}^{\text{train}}$. At the final step of SocInf, it utilizes supervised training on $D_{\text{attack}}^{\text{train}}$ to obtain the attack model F_{attack}. The second layer of Fig. 2 shows how to train the attack model in detail. Specifically, X_s^{in} serves as the mimic training set and X_s^{out} works as the mimic testing set.

IV. Performance Evaluation

A. Data Set Description

1) Tweet Emotion Intensity Data Set: This data set comes from a shared task on Emotion Intensity (EmoInt),[6] which provides researchers with a data set of tweets and the associated emotion status. The purpose of EmoInt is to construct a system to automatically determine the intensity or degree of a certain emotion felt by the speaker when given the system a tweet [21]. The data set of EmoInt contains 7097 tweets marked with four emotions: joy (1611 tweets), sadness (1533 tweets), fear (2252 tweets), and anger (1701 tweets). The emotion status of the tweet data is obtained by manual annotation.

For our experiment, we build a simplified Tweet data set where each tweet text is turned into a vector of 200 binary features. Each feature corresponds to a keyword corresponding to a certain emotion and represents whether the tweet context consists of this word or not. In order to evaluate the performance of SocInf, we randomly select half of the Tweet data set to train the target model. The rest of the data set contributes to the testing set and a training set of the mimic model. In order to evaluate the impact of the target model's outputs, we first train a four-class classification model with the Tweet data's original labels. Then we mark the joy tweets with the label "positive emotion" and the rest tweets with the label "negative emotion," and train a two-class classification model on the relabeled data for SocInf's evaluation.

2) Weibo Data Set: This Weibo data set is obtained through public collection and extraction from Sina Weibo and opened on the NLPIR platfrom.[7] It contains 23 000 records with

[6] http://saifmohammad.com/WebPages/EmotionIntensity-SharedTask.html

[7] http://www.nlpir.org/?action-viewnews-itemid-263/

attributes such as ID, article, discuss, *et al*. We mainly use the contents of the article to set up a keyword list according to different emotions and count the keywords to get the value of each feature. We first divide the data set into three classes and create three keywords lists of positive, negative, and neutral emotion, including 22, 28, and 23 keywords, respectively. We randomly select 2000 records for each emotion and construct our training data set which has three classes. Then, we divide Weibo data set into five classes and each class are corresponding to a certain emotion. The keyword list for the five emotions includes 20, 20, 20, 20, and 16 keywords separately. For our experiment, the feature vector in Weibo data set corresponds to the emotion keyword list and represents whether the corresponding word comes up in the content of Weibo or not. We select half of Weibo data set to train the target model, while the rest to contribute to the mimic model.

B. Experimental Setup

1) Target Model: To evaluate the performance of SocInf, we conduct attack experiments on target models trained with Xgboost, logistics, and BigML. For all target models, we assume the models as block boxes. We do not leverage any prior knowledge of the target model to conduct the attack, neither the structure of the target model nor the values of hyperparameters during the training process.

The first target model we take into account is trained by a cloud machine learning service platform: BigML. Here, we briefly introduce machine learning as a service (MLaaS). In order to help nonprofessionals build a machine learning model on their own data promptly, MLaaS offerings have recently sprouted up to meet this need by Internet giants such as Google,[8] Amazon,[9] and Microsoft.[10] MLaaS is a range of services that offer machine learning tools as part of cloud computing service, as the name suggests. With MLaaS, researchers can get access to machine learning technologies and learn from data without in-house domain expertise. MLaaS only provides little control of the regularization and the training process is hidden from the data owners. In the BigML platform, the users have no authority to manipulate an NN's attributes, including layer numbers, active function, or epochs. Therefore, we treat the BigML model as a black box. All attacks we demonstrate in this paper are performed entirely through the service standard APIs.

As for the local target models, we select two typical training algorithms to train the target model. The machine learning models can be divided into two classes roughly: a linear model and a nonlinear model. In order to evaluate the attack performance of SocInf sufficiently, our experiments consider both linear and nonlinear machine learning models. We select logistic regression model [10] which is the most representative model in linear models to build our target model. As for the nonlinear model, Xgboost [11] is chosen to train the target model.

In machine learning field, the logistic regression is a widely used linear model that uses a logistic function to model a binary dependent variable. We leverage scikit-learn[11] to build the local target model. In our experiments, we use the python library of Xgboost[12] to build our local model, and then train a series of Xgboost models corresponding to different data sets and experimental settings. To ensure the accuracy of our experiments, we keep the hyperparameters consistent when training the Xgboost models on the same data set.

2) Mimic Model: Since an NN with multiple layers can simulate any type of machine learning models, we leverage NNs to construct our mimic model. In our experiments, we use Pytorch library to build our mimic model. We build our mimic model with 10 layers and each layer with 50 nodes at least, such that the mimic network has enough potential to simulate the target model.

3) Attack Model: The objective of the attack model is to infer whether a given record is in the target model's training data; therefore, the membership inference attack can be regarded as a binary classification. We still use Xgboost technology to build our attack model because our experiments prove that Xgboost has the best performance of attack.

4) Data Settings: The training set of the victim model and mimic model are randomly selected from the respective data sets. Since we assume that we can't obtain any detailed record of the target model's training data set, there is no overlap between the data sets of target models and mimic models. Specifically, we use the training set of Emotion Intensity to train the target model, and we use the synthetic data as the testing set of our attacks. The training set has 3613 tweets while the testing set has 3142 tweets. As for Weibo data set, we randomly divide it into two sets. One set serves as the training data of the target model, and the other set becomes the synthetic data of SocInf. The size of each subset is 2000.

C. Performance of SocInf

The purpose of SocInf is to infer the member of the target model's training data set. We evaluate the performance of our attacks on different machine learning models separately. In our evaluation experiments, we set the number of members equal to the number of nonmembers, in order to achieve the baseline accuracy to be 0.5.

We evaluate the performance of SocInf using standard *accuracy*, *precision*, and *recall* metrics of membership inference attacks. Specifically, *precision* presents the proportion of the data records predicted as member of the training data set that are indeed in the training set of the target model. The *recall* presents the fraction of the training records that we can correct infer as the training set's members. The *recall* measures the coverage of our inference attacks in other words. The closer the *recall* to 1, the better SocInf detects the members of the training set from the test set, which also proves that SocInf threatens the data security of the target model's training data set severely. In the following part, we will show the

[8]ttps://cloud.google.com/prediction
[9]https://aws.amazon.com/machine-learning
[10]https://studio.azureml.net
[11]http://scikit-learn.org/stable/
[12]https://github.com/dmlc/xgboost/

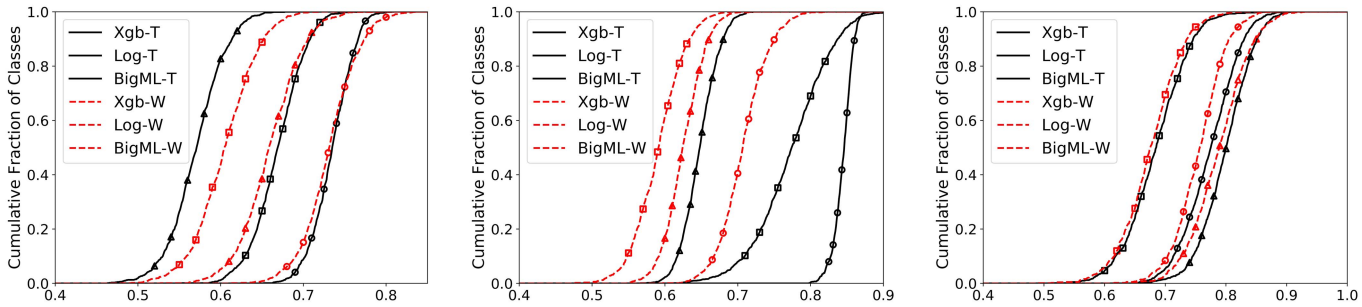


Fig. 5. Empirical CDF of accuracy, precision and recall of SocInf against different target models on tweets and Weibo data set. (a) Attack accuracy. (b) Attack precision. (c) Attack recall.

TABLE II
PERFORMANCE OF MEMBERSHIP INFERENCE ATTACKS AGAINST DIFFERENT CLASSIFICATION MODELS

Data	BigML			Logistics			Xgboost		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Tweet(2)	0.5073	0.5157	0.6871	0.6741	0.8274	0.7251	0.6154	0.7515	0.6381
Tweet(4)	0.5746	0.6483	0.7991	0.6684	0.7735	0.6839	0.7339	0.8439	0.7728
Weibo(3)	0.5228	0.5132	0.7023	0.5667	0.5460	0.6956	0.6168	0.5668	0.7229
Weibo(5)	0.6541	0.6202	0.7651	0.6086	0.5493	0.6867	0.7343	0.7082	0.7964

performance of SocInf on membership inference with different data sets and different machine learning models.

For the Tweet data set, we evaluate our attack methods on the victim models trained using BigML, logistic regression, and Xgboost algorithm. Fig. 5 shows the empirical cumulative distribution function (CDF) of the accuracy, precision, and recall of our attack on different target models. From Fig. 5(a), we can see that attack accuracies of all target models are higher than the baseline 0.5. In particular, for the attack on Xgboost model, we can achieve a mean accuracy of 0.7339. As for the attack precision of SocInf, we can achieve a mean accuracy of 0.8439 on Xgboost model. Even on the BigML model where SocInf performs the worst, we can still get a mean precision of 0.6483, which is much better than random guess. The attacks on target models trained with Xgboost and BigML have similar recall values, and the attack on logistic regression has a mean recall value of 0.6839.

For Weibo data set, Fig. 5 shows that our attack has the best performance on the models trained using Xgboost algorithm. The attack against the Xgboost model could achieve a mean attack accuracy and a mean attack precision of 0.7343 and 0.7082, respectively. The mean value of recall metric even can achieve to 0.7964. As for the target model trained using BigML cloud platform, the experiment shows that the accuracy and the precision of our attacks are both higher than 60%. The attack against the logistics models perform worst in our experiments and the reason is that the logistics algorithm has a poor performance on multiple-class classification tasks. For the attack on logistics models, SocInf can achieve a mean accuracy of 0.6086, which is much better than random guess. From the results, we can see that SocInf could infer the membership of the target model's training data effectively, and the attack effects are independent of the data set.

D. Impact of the Number of Classes

In SocInf, the number of prediction classes of the victim model affects the degree of model leakage. The more the classes, the more detailed prediction results are available to SocInf.

To evaluate the impact of the number of classes of SocInf, we train a series of victim models using BigML, logistics, and Xgboost on the Tweet data set with two classes. We mark the "joy" mood with the label "positive emotion," and the rest moods (including sadness, fear, and anger) are "negative emotion." From Table II, we can see that with more output information of the target model, our attack method performs better on membership inference of the target models trained by BigML and Xgboost. The accuracy and precision of attack on BigML increase by 6.73% and 13.26%, respectively. The recall of our attack model is increased higher than 10%. As for the Xgboost model, the performance of our attack has a significant improvement. The accuracy of our attack model increases by 11.85%, and the precision increases by 11.24%. There is an abnormal instance that the performance on the target model trained by logistics has a remarkable reduction with additional information. The main reason is that the classification accuracy of the logistics models is relatively lower than other target models. The logistics model has a classification accuracy of 85.33% on the Tweets with two classes, while the accuracy can only achieve 71.05% when the Tweets are clustered with four classes. By contrast,

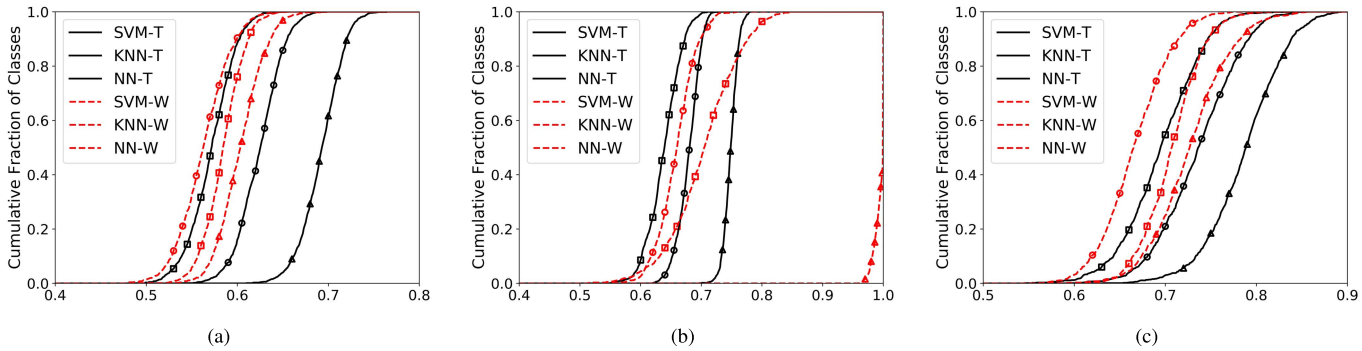


Fig. 6. Empirical CDF of accuracy, precision and recall of SocInf with different attack models on Tweets and Weibo data set. (a) Attack accuracy. (b) Attack precision. (c) Attack recall.

the accuracy rates of the Xgboost model and BigML are higher than 95%.

For Weibo data set, we train the target models with three classes, and the class label indicates which mood the data provider keeps in, including positive, negative, and neural mood. From Table II, we can see that the performance of SocInf improves with the class number increases. For the attack against the Xgboost model, the accuracy and precision increase by 11.75% and 14.14%, respectively. As for the target models trained with logistics classification algorithm, the attack accuracy increases by 4.19% with the increase in class number, while the metrics of precision and recall have negligible changes. The recall even becomes lower with increasing class number. Similar to the attack against Xgboost models, the performance of attacks against BigML achieves an improvement of 13.12% but the mean recall decreases by 11.7%. The recall rates of attacks against BigML and Xgboost models are both higher than 75%.

E. Impact of the Attack Models

To evaluate the impacts of the attack models on the SocInf performance, we test the attack effects of attack models trained by different machine learning algorithms, including support vector machine (SVM), KNN, and NNs. From Fig. 6, we can see that the NN attack model has the best attack effect, and this attack model can reach a mean accuracy of 0.6928 and a mean precision of 0.7481. The attack model trained with KNN has the worst performance, while its accuracy and precision are lower than the NN model by 12.19% and 11.03%, respectively. The mean accuracy of the SVM model is 0.6242 and the mean precision is 0.6783. As for the metric of recall, the mean values of all attack models are higher than 0.69, and even the maximal recall of the NN model can achieve 0.8869. The experiments on Tweets data set show that all attack models perform much better than the baseline accuracy of 0.5.

For Weibo data set, Fig. 6(a) indicates that the attack models trained by SVM, KNN, and NN have semblable attack accuracy and the accuracy of the best model differs by only 4.73% from that of the worst model. In general, the attack model trained by NN has the best performance. In particular, the precision of the NN attack model achieves a mean value of 0.9837, indicating that our NN attack model has a powerful ability to identify the data appearing in the target model's training set. The precision of the NN model is higher than that of the SVM model and KNN model by 34.9% and 27.98%, respectively. Nevertheless, the attack models of SVM and KNN still have high precision rates which are above 0.6. As for the recall metric, the NN model has a better performance than other attack models and the mean recall of NN model is 0.6958.

From the results, we can observe that the inference attacks using NN always achieve the highest rates of accuracy, precision, and recall compared with KNN and SVM. This phenomenon is caused by the strong learning ability of NN models. In practice, the prediction difference between the training and testing data on the same model always has a mass of regulars or distribution modes, and NN models can learn the hidden modes of the prediction differences between the training and testing data very well. Thus, the attack model trained with NN can infer the member and nonmember of the target model's training data precisely. As for the attack models trained with KNN and SVM, although these models perform worse than NN models, they can still achieve a mean precision above 0.6 and a mean recall above 0.65.

The experiments show that the mimic model that we build through modifying GAN has the ability to output adequate information about the input data that can be leveraged to determine whether the input data are in the target model's training set or not. Overall, no matter which machine learning algorithm the target model or attack model employs, SocInf has the ability to breach the privacy of the target model.

F. Impact of the Overfitting Level of Target Model

In this section, we evaluate the impact of overfitting on the attack performance of SocInf. To demonstrate the relationship between overfitting and membership inference, we attack a series of target models trained with different parameters, while the model's training set and training algorithm are kept the same. In order to quantify the overfitting level of a target model, we make use of the difference between its prediction accuracy on the training set and testing set as the indicator. Fig. 7 shows the performances of three attack models against the same set of target models with different levels of overfitting. It is obvious that with the decrease

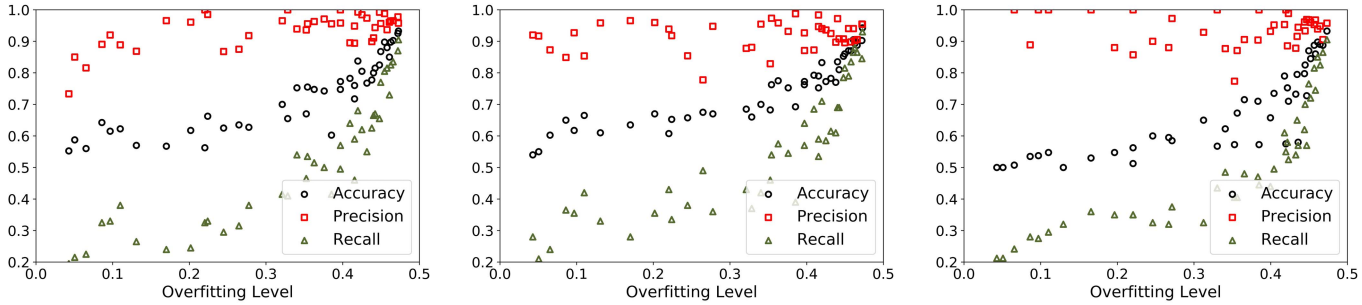


Fig. 7. Performance for different overfitting levels of the target model. (a) SVM attack model. (b) KNN attack model. (c) NN attack model.

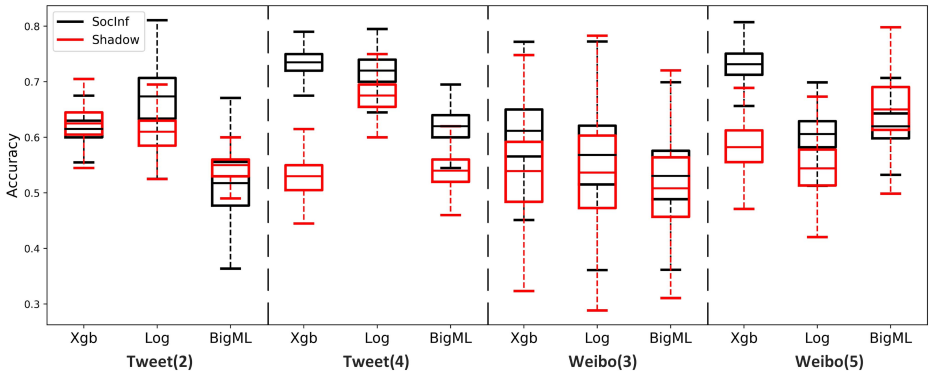


Fig. 8. Performance comparison of accuracy.

of the machine learning model's generalization, it is more vulnerable to membership inference attacks. For instance, our inference attack using SVM achieves a relatively weak accuracy around 0.5, and there is only 6% difference between the target model's training and testing accuracy. On the other hand, SocInf achieves accuracy and precision both around 0.9 with all attack models, while the corresponding target model provides an overfitting level of 47%.

Our experiments illustrate that overfitting can increase the risk of privacy leakage to a model's training data. An overfitting model will rely on the specific training data heavily, and its predictions of training and testing data set differ substantially. With the overfitting level of a model increasing, the prediction difference becomes larger. As such SocInf can distinguish the member and nonmember of the training set precisely and achieve a relatively high attack precision against an overfitting model.

G. Performance Comparison

In this section, we compare the membership inference attack performance of SocInf with that of "shadow training" technology (called Shadow) [7] on two different data sets. We still leverage standard *accuracy, precision, and recall* metrics to evaluate the attack performance as previously mentioned. Fig. 8 shows the comparison results of the attack accuracy. For the target models of two-class and three-class classifications, SocInf has similar attack accuracy with Shadow, and even the mean accuracy of SocInf is lower than that of Shadow by 0.88% on the target models trained by BigML. When it comes to the four-class and five-class classification models, our attack method achieves better performance of membership inference. Compared with Shadow on Tweet data set, SocInf gets accuracy augmentations on Xgboost, logistics, and BigML models by 20.65%, 8.67%, and 9.63% respectively. As for the five-class models trained on Weibo data set, our method achieves an accuracy improvement of 15.43% of inference attacks on Xgboost model, and our method gets an increase on the logistics model by 6.34%.

From Fig. 9, we can see that for the precision metric, SocInf performs better than Shadow on Tweet data set, while achieving resemble performance with Shadow on Weibo data set. For the two-class classification models, SocInf achieves precision enhancements by 11.78% and 18.94% for Xgboost and logistics, respectively. However, for the target model trained with BigML, SocInf attains a worse inference performance than the comparison algorithm. For the four-class classification models trained on Tweets, SocInf performs better than the contrast attack approach on all machine learning models that we test in our experiments. SocInf achieves precision improvement by 22.47%, 16.1%, and 6.65% on different training algorithms. Nevertheless, SocInf is not always better than the Shadow, especially for the three-class classification models trained with Weibo data set, where the attack precision of SocInf is extremely similar to Shadow. The mean precision differences are 5.89%, 0.6%, and 0.62% for Xgboost, logistics and BigML models respectively. As the number of classes increases, the precision improvement of SocInf is much

higher than that of Shadow. The inference results of SocInf are quite accurate when SocInf meets with the five-class classification models constructed by Xgboost and logistics algorithms.

As for the recall metric shown in Fig. 10, SocInf is able to achieve higher recall rates than the contrast algorithm on all target models. High recall rates mean that our approach has a strong ability to select the members of the target model's training set from the whole test set. The mean recall of SocInf is more accurate than that of Shadow by 18.47%. In particular, when the target model is constructed for four-class classification using Xgboost, our algorithm achieves a mean recall rate of 77.28% which is 26.48% more accurate than Shadow.

In summary, SocInf performs much better than Shadow on different machine learning models. Nevertheless, when the target model is trained by the cloud platform BigML, the attack effects of our algorithm are close to the effects of Shadow. There are several reasons which caused such a phenomenon. One reason is that the prediction accuracy of the target model trained using BigML is relatively inaccurate and the prediction results of the target model cannot provide the abundant information of training set for inferring the memberships. In addition, the simulating prediction behavior of the mimic model cannot reveal the prediction differences of the BigML model's training set and testing set. In SocInf, we leverage an NN with multiple layers to simulate the BigML's model; however, the target model has a simple structure. The deep NN would extract many redundant features which result in the overfitting of the synthetic data and their prediction results obtained from the target model. Thus, the predictions of the testing data obtained from the mimic model and the BigML's shallow NN have different distributions. The attack model trained on the prediction results of the mimic model does not have the ability to reveal the relationship between the predictions and the training set's membership of the target model. In addition, SocInf has a perfect performance of membership inference attacks on Xgboost and logistics models, even SocInf can only get the prediction interface of the target model, while the contrast algorithm trains the "shadow model" using the same parameters of the target model.

V. Mitigation Strategies for SocInf

As explained in Section III and IV, SocInf relies on the probabilities of each class predicted by the target model. Thus, making the target model return only the label of the most likely class seems like a feasible approach; however, it may result in incomprehensible errors in some scenarios. In this section, we propose several mitigation strategies aiming at decreasing the risks of membership privacy leakage.

A. Training Models With Differential Privacy

Differential Privacy is a technique that can provide means to maximize the accuracy of queries from databases while minimizing the privacy impact on individuals whose information is in the database. Training a machine learning model with differential privacy can reduce the prediction differences between the training and testing set caused by an individual, and therefore, it increases the difficulty for SocInf to infer whether this individual is in the training set or not.

B. Coarsen Precision of the Prediction Results

In order to mitigate the membership inference risk, we can only output few floating point digits of the prediction probabilities. The fewer significant digits the predictions have, the less information the model leaks. For instance, we can modify the prediction $[0.13, 0.27, 0.6]$ as $[0.1, 0.3, 0.6]$ and output the processed result.

C. Avoid Overfitting

The key observation of SocInf is that the machine learning models often have different prediction behaviors on the data that they were trained on versus the data that they "meet" for the first time. Overfitting can increase the prediction differences between the training and testing data. Thus, avoiding overfitting of the machine learning model can defend such attacks effectively. Generally, using regularization techniques can overcome overfitting in machine learning to some extent.

VI. Related Work

A. Attacks on Machine Learning Models

With the wide application of machine learning technology in various fields of the Internet, there exist multiple other types of attacks on ML. Model reversal attacks in biomedical data settings are proposed by Fredrikson *et al*. [25]. In this scenario, the goal of attacker is to infer the missing attributes of her victim based on the output of a trained ML model. The authors later generalize the model inversion attack to a wider range of scenario [26]. For instance, they show that an attacker can reconstruct a recognizable face of her victim through model inversion.

Tramèr *et al*. [8] propose another attack against the ML models, namely, model extraction attack. This type of attack is designed to steal the ML model, i.e., the learned parameters of the model, through the output of MLaaS API itself. They first propose an equation solving attack, where an attacker repeatedly queries MLaaS API and constructs a set of equations with the output posteriors. The attacker can obtain the weight of the ML model by solving these equations. Tramèr *et al*. [8] further propose a path-finding algorithm, which is the first practical method of stealing decision trees. In the end, Tramèr *et al*. [8] point out that even ML models which provide predictive labeling without providing prediction posteriors can still be stolen through retraining strategies, such as active learning. It is worth noting that we do not consider hiding posteriors as one effective defense mechanism in this paper due to the effectiveness of the model extraction attack.

Another major family of attacks against machine learning are adversarial examples [27]–[30]. In this setting, an attacker adds a controlled amount of noise to a data point that aims to fool the trained ML model to erroneously classify the data point. Adversarial examples can cause serious risks in multiple areas, such as autonomous driving and voice recognition. On the other hand, researchers have recently discovered that

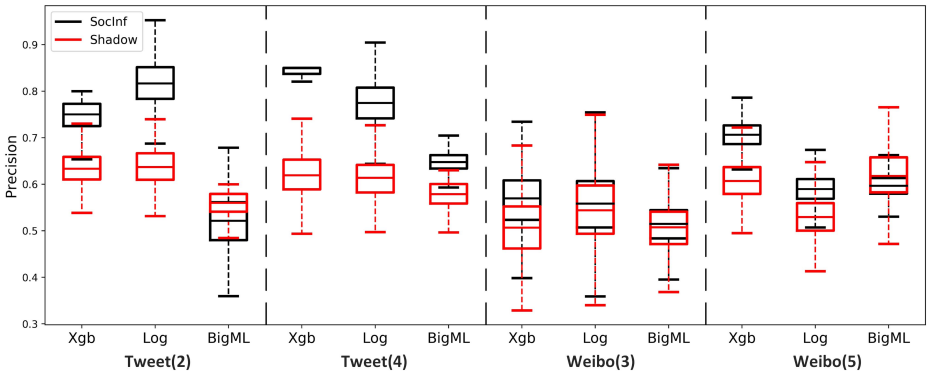


Fig. 9. Performance comparison of precision.

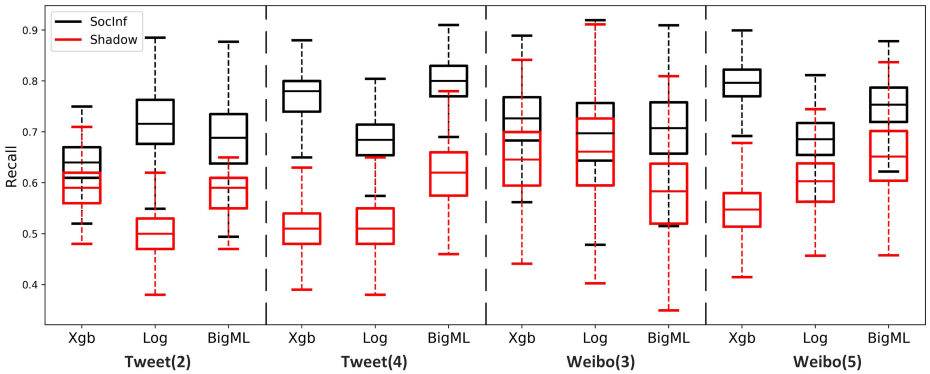


Fig. 10. Performance comparison of recall.

adversarial examples can also help protect the privacy of online social network users [31], [32].

Jagielski *et al*. [33] consider the setting of poisoning attacks where the attacker affects the training data and first systematically study countermeasures for linear regression models under different adversarial models. They design a new optimization framework for regression poisoning and introduce a fast statistical attack that requires minimal knowledge of the training process. On the defense axis, they take a principled approach to construct a defense algorithm called TRIM that largely outperforms existing robust regression methods.

B. Membership Inference Attacks

In addition to the attacks described above, there is another attack called membership inference attacks which have been successfully carried out in different areas. The first membership inference attack on genomic data was proposed by Homer *et al*. [34]. Later, Backes *et al*. [35] extend this attack to other types of biomedical data. Recently, aggregated mobility traces are also shown to be susceptible to membership inference attacks [36], where the attack is modeled as a distinguishability game.

The private attributes of a specific user (e.g., location and gender) can be inferred via leveraging its public data [37], and they propose a practical two-phase defense against attribute inference attacks called AttriGuard. First, AttriGuard finds a minimum noise for each attribute value via an evasion attack. And then, AttriGuard randomly selects one of the previously discovered noises and misleads the attacker's inference. The final goal of the defender is to add random noise to the user's public data to reduce the attacker's inference accuracy with a small utility loss.

Membership inference attacks based on machine learning models was lately proposed in [7]. The key contribution is the development of shadow model training, the purpose of which is to simulate the behavior of the target model and generate training data for the attack model. Following this work, several research methods of membership inference attacks against machine learning models are proposed from different perspectives [38]–[43]. Now many ML frameworks and services are available to data holders, but there may be a malicious ML provider which provides model-training code can remember the information about the training data set [44]. They prove that using third-party code to train ML models on sensitive data is risky even if the model is only released as a black box.

C. Privacy-Preserving Machine Learning

Another related domain of work is privacy-preserving machine learning. Mohassel and Zhang [45] present efficient scheme for training linear regression, logistic regression, NNs in a privacy-preserving way. Their protocols fall in

the dual-server mode where data is distributed across two noncolluding servers. The authors use two party calculation approach to implement these protocols. A secure aggregation protocol over high-dimensional data was proposed in [46], which is a key component of distributed machine learning. The protocol is also based on multiparty computing, and its authors prove the security under both honest-but-curious and active adversary settings. A large-scale evaluation validates the effectiveness of this protocol. Nasr *et al*. [47] mainly discuss how to protect the machine learning model from the impact of black-box membership inference attacks. They introduce a rigorous mechanism that ensures no adversary can distinguish between training set of the model and other data samples from the same distribution. This mechanism is formalized as a min-max privacy game to minimize the classification error and guarantee the max utility.

Besides privacy-preserving model training, other works study privacy-preserving classification. Bost *et al*. [48] design three protocols based on homomorphic encryption. They focus on three ML classifiers, such as hyperplane decision, Naive Bayes, and decision trees, and show that their protocols can be executed efficiently. Backes *et al*. [49] constructed a privacy-preserving random forests classifier for medical diagnosis based on [48].

In addition, some hardware facilities can also be used to protect the privacy of machine learning. A new system for privacy-preserving outsourced machine learning called Chiron can allow data holders to use MLssS without exposing their data to service providers where the premise is that the provider does not require to disclose its model, configuration parameters, and training algorithms [50]. Chiron is implemented with software guard extensions enclaves which protects code and data from all other software on the platform and Ryoan sandbox to achieve data privacy and model confidentiality.

VII. Conclusion

In this paper, we proposed SocInf which can determine whether a given record is in the target model's training data set. With SocInf, we can build a machine learning model that has the similar prediction behaviors with the target model, in the situation that only the prediction interface of target model can be obtained. We empirically evaluate our membership inference technique on machine learning models trained by local logistic regression, Xgboost, and BigML cloud platform. Using the realistic data and classification tasks, our experiments show that SocInf is generic and effective in inferring the membership of different type of machine learning models' training data set.

However, there are still some issues to be addressed. First, in the synthetic data generation step of SocInf, our inference attack needs to traverse the all possible combinations of possible values for all features. This method is acceptable when the data have a small number of features, but may fail for data with a large number of features such as an image data set or a video data set. Another issue needs to be noticed is that when we cluster the synthetic data into two clusters, we mark the cluster whose centroid is closer to 1 with the label "in." In practice, a part of training data may have prediction results far from 1. We empirically filter these data from the synthetic data, and such an approach will affect the mimic model's prediction behavior to some extent. We need to find a more proper metric to separate the synthetic data. In addition, we only focus on membership inference attack against supervised models in this paper. In the future, we will explore how to conduct the attacks against unsupervised and semisupervised models for more general cases.

References

[1] Y. Liu, Y. Han, Z. Yang, and H. Wu, "Efficient data query in intermittently-connected mobile ad hoc social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 5, pp. 1301–1312, May 2015.

[2] A. M. Kaplan and M. Haenlein, "Users of the world, unite! The challenges and opportunities of social media," *Bus. Horizons*, vol. 53, no. 1, pp. 59–68, Jan./Feb. 2010.

[3] (2018). *Wikipedia on Facebook Company Info*. [Online]. Available: https://en.wikipedia.org/wiki/Facebook

[4] (2018). *Wikipedia on Twitter Company Info*. [Online]. Available: https://en.wikipedia.org/wiki/Twitter

[5] E. Toch, B. Lerner, E. Ben-Zion, and I. Ben-Gal, "Analyzing large-scale human mobility data: A survey of machine learning methods and applications," *Knowl. Inf. Syst.*, vol. 58, no. 3, pp. 501–523, Mar. 2019.

[6] H.-H. Won *et al.*, "Predicting national suicide numbers with social media data," *PLoS ONE*, vol. 8, no. 4, Apr. 2013, Art. no. e61809.

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

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

[9] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.

[10] S. H. Walker and D. B. Duncan, "Estimation of the probability of an event as a function of several independent variables," *Biometrika*, vol. 54, nos. 1–2, pp. 167–179, Jun. 1967.

[11] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[12] H.-H. Chi, "Interactive digital advertising vs. Virtual brand community: Exploratory study of user motivation and social media marketing responses in Taiwan," *J. Interact. Advertising*, vol. 12, no. 1, pp. 44–61, Sep. 2011.

[13] W. L. Bennett and A. Segerberg, "Digital media and the personalization of collective action: Social technology and the organization of protests against the global economic crisis," *Inf., Commun. Soc.*, vol. 14, no. 6, pp. 770–799, Sep. 2011.

[14] X. Wang, C. Yu, and Y. Wei, "Social media peer communication and impacts on purchase intentions: A consumer socialization framework," *J. Interact. Marketing*, vol. 26, no. 4, pp. 198–208, Nov. 2012.

[15] R. Agnihotri, P. Kothandaraman, R. Kashyap, and R. Singh, "Bringing 'social' into sales: The impact of salespeople's social media use on service behaviors and value creation," *J. Pers. Selling Sales Manage.*, vol. 32, no. 3, pp. 333–348, 2012.

[16] D. E. Baird and M. Fisher, "Neomillennial user experience design strategies: Utilizing social networking media to support 'always on' learning styles," *J. Educ. Technol. Syst.*, vol. 34, no. 1, pp. 5–32, Sep. 2005.

[17] E. G. Spanakis *et al.*, "MyHealthAvatar: Personalized and empowerment health services through Internet of Things technologies," in *Proc. MOBIHEALTH*, Nov. 2014, pp. 331–334.

[18] J. Qi, P. Yang, A. Waraich, Z. Deng, Y. Zhao, and Y. Yang, "Examining sensor-based physical activity recognition and monitoring for healthcare using Internet of Things: A systematic review," *J. Biomed. Inform.*, vol. 87, pp. 138–153, Nov. 2018.

[19] M. Sharma, K. Yadav, N. Yadav, and K. C. Ferdinand, "Zika virus pandemic—Analysis of Facebook as a social media health information platform," *Amer. J. Infection Control*, vol. 45, no. 3, pp. 301–302, Mar. 2017.

[20] Y. A. Strekalova, "Emergent health risks and audience information engagement on social media," *Amer. J. Infection Control*, vol. 44, no. 3, pp. 363–365, Mar. 2016.
[21] S. Mohammad and F. Bravo-Marquez, "Emotion intensities in Tweets," in *Proc. 6th Joint Conf. Lexical Comput. Semantics*, Vancouver, BC, Canada, Aug. 2017, pp. 65–77.
[22] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan, "An introduction to MCMC for machine learning," *Mach. Learn.*, vol. 50, no. 1, pp. 5–43, Jan. 2003.
[23] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*. Reading, MA, USA: Addison-Wesley, 1974.
[24] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: Perceptron, madaline, and backpropagation," *Proc. IEEE*, vol. 78, no. 9, pp. 1415–1442, Sep. 1990.
[25] 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. 23rd USENIX Secur. Symp.*, 2014, pp. 17–32.
[26] 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. Secur.*, Oct. 2015, pp. 1322–1333.
[27] Y. Vorobeychik and B. Li, "Optimal randomized classification in adversarial settings," in *Proc. Int. Conf. Auton. Agents Multi-Agent Syst.*, May 2014, pp. 485–492.
[28] B. Li and Y. Vorobeychik, "Scalable optimization of randomized operational decisions in adversarial classification settings," in *Proc. AISTATS*, 2015, pp. 599–607.
[29] N. Papernot, P. Mcdaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM CCS*, Apr. 2017, pp. 506–519.
[30] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. (2017). "Ensemble adversarial training: Attacks and defenses." [Online]. Available: https://arxiv.org/abs/1705.07204
[31] S. J. Oh, M. Fritz, and B. Schiele, "Adversarial image perturbation for privacy protection a game theory perspective," in *Proc. IEEE ICCV*, Oct. 2017, pp. 1491–1500.
[32] Y. Zhang, M. Humbert, T. Rahman, C.-T. Li, J. Pang, and M. Backes, "Tagvisor: A privacy advisor for sharing hashtags," in *Proc. World Wide Web Conf.*, Apr. 2018, pp. 287–296. [Online]. Available: https://arxiv.org/abs/1802.04122
[33] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li. (2018). "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning." [Online]. Available: https://arxiv.org/abs/1804.00308
[34] N. Homer *et al.*, "Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays," *PLoS Genet.*, vol. 4, no. 8, Aug. 2008, Art. no. e1000167.
[35] M. Backes, P. Berrang, M. Humbert, and P. Manoharan, "Membership privacy in MicroRNA-based studies," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 319–330.
[36] A. Pyrgelis, C. Troncoso, and E. De Cristofaro, "Knock knock, who's there? Membership inference on aggregate location data," in *Proc. NDSS*, 2018, pp. 1–15.
[37] J. Jia and N. Z. Gong. (2018). "Attriguard: A practical defense against attribute inference attacks via adversarial machine learning." [Online]. Available: https://arxiv.org/abs/1805.04810
[38] Y. Long, V. Bindschaedler, and C. A. Gunter. (2017). "Towards measuring membership privacy." [Online]. Available: https://arxiv.org/abs/1712.09136
[39] S. Yeom, M. Fredrikson, and S. Jha. (2017). "Privacy risk in machine learning: Analyzing the connection to overfitting." [Online]. Available: https://arxiv.org/abs/1709.01604
[40] Y. Long *et al.* (2018). "Understanding membership inferences on well-generalized learning models." [Online]. Available: https://arxiv.org/abs/1802.04889
[41] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes. (2018). "ML-Leaks: Model and data independent membership inference attacks and defenses on machine learning models." [Online]. Available: https://arxiv.org/abs/1806.01246
[42] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei. (2018). "Towards demystifying membership inference attacks." [Online]. Available: https://arxiv.org/abs/1807.09173
[43] N. Buescher, S. Boukoros, S. Bauregger, and S. Katzenbeisser, "Two is not enough: Privacy assessment of aggregation schemes in smart metering," in *Proc. PETS*, Oct. 2017, pp. 198–214.
[44] C. Song, T. Ristenpart, and V. Shmatikov, "Machine learning models that remember too much," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 587–601.
[45] P. Mohassel and Y. Zhang, "SecureML: A system for scalable privacy-preserving machine learning," in *Proc. IEEE Symp. Secur. Privacy*, May 2017, pp. 19–38.
[46] K. Bonawitz *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1175–1191.
[47] M. Nasr, R. Shokri, and A. Houmansadr, "Machine learning with membership privacy using adversarial regularization," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 634–646.
[48] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," in *Proc. NDSS*, Feb. 2015, p. 4325.
[49] M. Backes *et al.*, "Identifying personal DNA methylation profiles by genotype inference," in *Proc. IEEE Symp. Secur. Privacy*, May 2017, pp. 957–976.
[50] T. Hunt, C. Song, R. Shokri, V. Shmatikov, and E. Witchel, "Chiron: Privacy-preserving machine learning as a service," in *Proc. PETS*, Oct. 2018, pp. 123–142.

Gaoyang Liu (S'19) received the B.S. degree of information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2015, where he is currently pursuing the Ph.D. degree in electronics and information engineering.

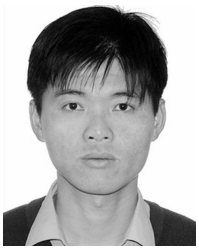
His current research interests include machine learning, mobile sensing and data privacy protection.

Chen Wang (S'10–M'13–SM'19) received the B.S. and Ph.D. degrees from the Department of Automation, Wuhan University, Wuhan, China, in 2008 and 2013, respectively.

From 2013 to 2017, he was a Post-Doctoral Research Fellow with the Networked and Communication Systems Research Lab, Huazhong University of Science and Technology, Wuhan. He was joined with the Huazhong University of Science and Technology, where he is currently an Associate Professor. His current research interests include wireless networking, Internet of Things, and mobile computing, with a recent focus on privacy issues in wireless and mobile systems.

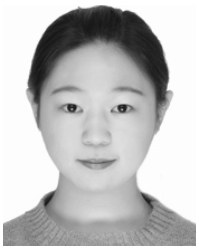
Kai Peng received the B.S., M.S., and Ph.D. degrees from Huazhong University of Science and Technology, Wuhan, China, in 1999, 2002, and 2006, respectively.

He is currently a Full Professor of the Huazhong University of Science and Technology. His current research interests include wireless networking and big data processing.



Haojun Huang received the B.S. degree from the School of Computer Science and Technology, Wuhan University of Technology, Wuhan, China, in 2005, and the Ph.D. degree from the School of Communication and Information Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2012.

From 2012 to 2015, he was a Post-Doctoral Researcher with the Research Institute of Information Technology, Tsinghua University, Beijing, China. From 2015 to 2017, he was an Assistant Professor with Wuhan University, Wuhan. He is currently an Associate Professor with the Huazhong University of Science and Technology, Wuhan. His current research interests include wireless networks, big data, and software-defined networking.



Yutong Li received the B.E. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2018, where she is currently pursuing the M.E. degree in electronics and information engineering.

Her current research interests include machine learning and Internet of Things.

Wenqing Cheng (M'07) received the B.S. degree in telecommunication engineering and the Ph.D. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1985 and 2005, respectively.

She is currently a Professor with the School of Electronic Information and Communications, Huazhong University of Science and Technology. Her current research interests include information systems and e-learning applications.