Differential Privacy Protection Against Membership Inference Attack on Machine Learning for Genomic Data

Junjie Chen[1], Wendy Hui Wang[2] and Xinghua Shi[1]*

[1]*Department of Computer and Informatics Sciences, Temple University, Philadelphia, PA 19122, USA.*

[2]*Department of Computer Science, Stevens Institute of Technology, Hoboken, NJ 07030, USA.*

** To whom correspondence should be addressed. E-mail: mindyshi@temple.edu*

Machine learning is powerful to model massive genomic data while genome privacy is a growing concern. Studies have shown that not only the raw data but also the trained model can potentially infringe genome privacy. An example is the membership inference attack (MIA), by which the adversary can determine whether a specific record was included in the training dataset of the target model. Differential privacy (DP) has been used to defend against MIA with rigorous privacy guarantee by perturbing model weights. In this paper, we investigate the vulnerability of machine learning against MIA on genomic data, and evaluate the effectiveness of using DP as a defense mechanism. We consider two widely-used machine learning models, namely Lasso and convolutional neural network (CNN), as the target models. We study the trade-off between the defense power against MIA and the prediction accuracy of the target model under various privacy settings of DP. Our results show that the relationship between the privacy budget and target model accuracy can be modeled as a log-like curve, thus a smaller privacy budget provides stronger privacy guarantee with the cost of losing more model accuracy. We also investigate the effect of model sparsity on model vulnerability against MIA. Our results demonstrate that in addition to prevent overfitting, model sparsity can work together with DP to significantly mitigate the risk of MIA.

Keywords: Differential privacy; Membership inference attack; Machine learning; Genomics.

1. Introduction

Genomics has emerged into a frontier of data analytics empowered by machine learning and deep learning, thanks to the rapid growth of genomic data that contains individual-level sequences or genotypes at large scale. To build powerful and robust machine learning models for genomics analysis, it is critical to collect, aggregate, and deposit sufficiently large assembly of genomic data. However, genetic privacy is a growing and legitimate concern that prevents wide sharing and aggregation of genomic data. Since genomic data is naturally sensitive and private, the sharing of such data can potentially disclose an individual's sensitive information such as identity, disease susceptibility or family history.[1,2] The current strategies of protecting genomic privacy is centered around relevant regulations and guidelines (i.e. HIPAA[3]), together

with the controlled access of individual-level genomic data (e.g. dbGaP[4]). However, we are in great need of new techniques for protecting genetic privacy toward an overarching goal of achieving trustworthy biomedical data sharing and analysis. Specifically, it is imperative to develop computational strategies to mitigate leakage of genetic privacy including the following two types of privacy leakage:

- *Privacy leakage via sharing data*: an individual's genomic data record may be leaked by sharing raw genomic data or summary statistics data; and
- *Privacy leakage via sharing models*: the information that an individual's genomic data is included in the training dataset of a particular machine learning model, may be leaked by sharing the model.[5]

While most of the prior works focus on the former type of privacy leakage resulted from sharing data,[6–8] in this study, we mainly focus on the latter type of privacy leakage from sharing machine learning models. Several studies have recently showed that trained models might memorize training data and thus disclose privacy of data records.[9,10] Although there exists a wide spectrum of attacks on machine learning models, the *membership inference attack* (MIA)[11] has recently attracted research efforts that induces privacy leakage when sharing machine learning models. More specifically, MIA refers to an attack to infer if the target record was included in the target model's training dataset. MIA has been demonstrated as an effective attack on images and relational data.[5,11,12] However, it remains unclear if MIA is effective on genomic data that significantly differ from conventional data.

Although less explored in genomics study, membership privacy leakage does pose an emerging risk given the increasing application and sharing of machine learning models in genomic data analysis. One particular scenario is that a publicly accessible model trained on valuable patient data may leak the privacy of patient.[13] For example, suppose a cancer treatment center builds a machine model to predict therapeutic responses based on patients' genomic and other biomedical data. The cancer center then releases the trained model to the public (e.g. for publications or depositing the model into a public model repository) or deploys the model as a machine-learning-as-a-service platform (e.g. Amazon Web service, Microsoft Azure, Google Cloud). An adversary may use the model's output to infer if a person, whose genomic data the adversary has access to, is a cancer patient or cancer survivor, and such information may provide the adversary some additional information that can be exploited. Hence, in this study, we will investigate the efficiency of MIA on machine learning models for phenotype prediction based on genomic data, a widely assessed prediction task carried out in agriculture, animal breeding, and biomedical science.

To defend against various attacks including MIA, a few techniques have been developed to mitigate privacy leakage such as homomorphic encryption,[14] federated learning,[15] and differential privacy (DP).[16] While homomorphic encryption and federated learning are mainly used to provide privacy protection for data sharing,[17,18] DP provides a popular solution for publicly sharing information not only about the data[19] but also the models.[20] The idea behind DP is that the query results cannot be used to infer information about any single individual, if the effect of perturbing in the database is small enough.[16] Recently, multiple defense mechanisms against MIA[21–23] have been explored, with DP[16] standing out as an efficient strategy that

provides a rigorous privacy guarantee against MIA.[11] Previous studies on imaging data[24,25] have shown that DP is an effective solution for granting wider access to machine learning models and results, while keeping them private. Therefore, we will mainly consider DP as a defense mechanism against MIA, given its theoretical privacy guarantee and its applicability for data and models. In this study, we investigate the effectiveness of using DP as a defense mechanism against MIA for phenotype prediction on genomic data to prevent the risk of sharing two widely-used machine learning methods including Lasso[26]) and convolutional neural network (CNN[27]). The **main contributions** of our study lie in two folds:

First, we investigate the vulnerability of machine learning against MIA on genomic data, and evaluate the effectiveness of using DP as a defense mechanism. Particularly, we evaluate the trade-off between the defense power against MIA and the prediction accuracy of the target model under various privacy settings of DP. Our results show that the relationship between the privacy budget and target model accuracy can be modeled as a log-like curve, and hence there exists a trade-off between privacy and accuracy near the turning point.

Second, we evaluate the effect of model sparsity on privacy vulnerability to effectively defend against MIA. Genomic data is primarily high dimensional, where the feature size is significantly larger than sample size. Hence, adding sparsity (e.g. the regularization terms in Lasso models) to machine learning models is a critical and effective strategy to alleviate the curse of dimensionality and avoid overfitting high-dimensional genomic data. Our results show that model sparsity together with DP can significantly mitigate the risk of MIA, in addition to providing robust and effective models for genomic data analysis.

2. Related Work

Membership inference attack (MIA). MIA is a privacy-leakage attack that predicts whether a given record was used in training a target model based on the output of the target model for the given record.[11] Shokri *et al.*[11] is the first work that defines MIA and inspires a few follow-up studies. For example, Truex *et al.*[28] characterize the attack vulnerability with respect to the types of learning models, data distribution, and transferability. Salem *et al.*[5] design new variants of MIA by relaxing the assumptions of model types and data. Long *et al.*[12] generalize MIA by identifying vulnerable records and indirect inference. While most existing works focus on MIA against discriminative models, relatively fewer works have considered MIA against generative models.[29,30] Liu *et al.*,[31] Song *et al.*[32] and Hayes *et al.*[33] propose new MIA variants against deep learning models including variational autoencoders (VAEs) and generative adversarial networks (GANs). These MIA attacks require only black-box access to a trained model. In practice, many studies usually release their models with white-box access.[17] Such white-box access provides many additional properties of the training models, which make an MIA attack even easier.

Differential privacy (DP). DP[16] has become the most widely-used approach that measures the disclosure of privacy pertaining to individuals. The guarantee of a DP algorithm lies in that anything the algorithm might output on a database containing some individual's information, is almost as likely to have come from a database without that individual's information. DP strategies have been applied to preserve genome privacy in genome-wide association studies

(GWAS).[8] For example, Johnson *et. al*[34] developed privacy-preserving algorithms for computing the number and location of single nucleotide polymorphisms (SNPs) that are significantly associated with certain diseases. Uhlerop *et. al*[7] proposed a method that allows for the release of aggregate GWAS data without compromising an individual's privacy. Various DP mechanisms also have been developed[35] to preserve model privacy, including a logistic regression with DP[36] and a random forest algorithm with DP.[37] Going beyond classic machine learning models, Shokri *et al.*[38] adapted DP to deep neural networks. Abadi *et al.*[25] developed a differentially private stochastic gradient descent (SGD) algorithm for the TensorFlow framework.

3. Methods

In this section, we introduce the methods used in our study, including differential privacy and membership inference attack. The supplementary materials and source code are available at `https://github.com/shilab/DP-MIA.git`.

3.1. *Membership inference attack (MIA).*

As illustrated in **Fig. 1**, MIA assumes that a target machine learning model is trained on a set of labeled samples from a certain population. The adversary utilizes the output of the target model of a given sample to infer the membership of the sample (i.e., the given sample was included in the training dataset of the target model). Formally, let $f_{target}()$ be the target model trained on a private dataset D^{train}_{target} which contains labeled samples $(\mathbf{x}, \mathbf{y})$. The output of the target model is a probability vector $\mathbf{y} = f_{target}(\mathbf{x})$ whose size is the number of classes. Let $f_{shadow}()$ be the shadow model trained on a dataset D^{train}_{shadow}, that is generated by the attacker to mimic the target model $f_{target}()$ (i.e. take similar input and output of the target model). We use the same assumption as in the pioneering work,[11] that the shadow dataset is disjoint from the private target dataset used to train the target model (i.e., $D^{train}_{shadow} \cap D^{train}_{target} = \emptyset$). Let $f_{attack}()$ be the attack model. Its input $\mathbf{x}_{attack}$ is composed of a predicted probability vector and a true label, where the distribution of predicted probability vectors heavily depends on the true label. Since the goal of the attack is membership inference, the attack model is a binary classifier, in which the output 1 indicates that the target record is in the training dataset, and 0 otherwise.

To construct the MIA model, a shadow training technique is often applied to generate the ground truth of membership inference. One or multiple shadow models are built to imitate the target model. In this study, we consider the white-box setting, where the adversary has the full knowledge of the target model including its hyperparameters and network structure. This white-box threat setting reflects the observations that researchers often share their full models and accidentally white-box representations of models may fall into the hands of an adversary via means such as a security breach.

3.2. *Differential privacy (DP)*

DP describes the statistics of groups while withholding individuals' information within the dataset.[16] Informally, DP ensures that the outcome of any data analysis on two databases

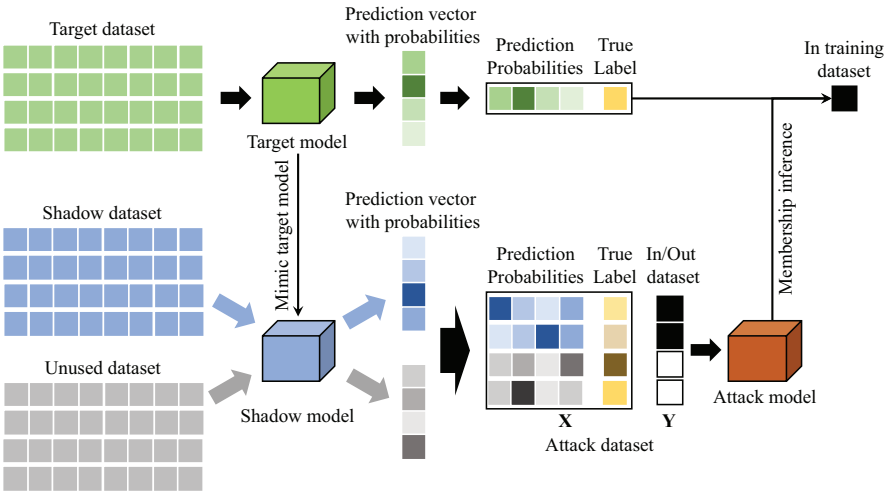

Fig. 1. **An illustration of the membership inference attack.** A record in the target dataset is fed into the target model and outputs a predicted probability vector. The shadow dataset and unused dataset are either simulated or selected from publicly available datasets that have the same distribution as the target dataset. A shadow model is built on the shadow and unused datasets to mimic the target model. The attack dataset is composed of the probability vectors and true labels. The attack model performs a binary classification (in/out) to determine whether a data record is included in the training dataset (in) or not (out).

differing in a single record does not vary much. Formally, a randomized algorithm $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ is (ε, δ)-differentially private if for all subsets of $\mathcal{S} \subseteq \mathcal{R}$ and for all database inputs $d, d' \in \mathcal{D}$ such that $\|d - d'\|_1 \leq 1$ satisfied with $\Pr[\mathcal{M}(d) \in \mathcal{S}] \leq e^{\varepsilon}\Pr[\mathcal{M}(d') \in \mathcal{S}] + \delta$. Here, $\|d - d'\|_1$ requires that the number of records that differ between d and d' is at most 1. The parameter ε is called the *privacy budget* and a lower ε indicates stronger privacy protection. The parameter δ controls the probability that ε-differential privacy is violated. A lower δ value signifies greater confidence of differential privacy. If $\delta = 0$, we say $\mathcal{M}$ is ε-differentially private, and simplify $(\varepsilon, 0)$-differential privacy as ε-differential privacy. A rule of thumb for setting δ is that it is smaller than the inverse of the training data size (i.e. $1/\|d\|$).[25]

4. Experimental Setup

4.1. *Dataset*

We evaluate the effectiveness of DP against MIA on a widely-used yeast genomic dataset.[39] We choose this yeast dataset because it provides an ideal scenario for evaluating the power and privacy of phenotype prediction with well-controlled genetic background and phenotype quantifications, without worries about complex genetic background and the hard-to-defined phenotypes in humans. We extract and filter missing values of the original genotypes[39] and organize them into a matrix that contains genotypes of 28,820 genetic variants or features (with values of 1 and 2 representing the allele comes from a laboratory strain or a vineyard strain respectively) from 4,390 individuals. Similar to any typical human genomic data, the yeast data is high dimensional where the feature size (28,820) is much larger than the sample size (4,390). We also obtain phenotypes or labels of these 4,390 individuals for 20 traits,[39] where

we pick the trait of copper sulfate as our target phenotype in this study. This trait represents the growth of yeast by measuring the normalized colony radius at a 48-hour endpoint in agar plates with different concentrations of copper sulfate.[39] Since MIA is mainly launched on classification models, we binarize the quantitative phenotype values as 1 if they are larger than the mean value and 0 otherwise.

4.2. *Implementation of target models*

For the target models of MIA, we implement a Lasso model[26,40] as an example of sparse learning models, and a CNN model[27,41,42] as an example of deep learning model, that are widely-used in analyzing high-dimensional genomics data.

Lasso is a regression analysis method that performs variable selection with a regularization term using ℓ_1 norm.[26] Lasso minimizes the residual sum of squares subject to the sum of the absolute value of the coefficients being less than a constant. The general objective of Lasso is $\min_{\beta} \frac{1}{2}\|y - \mathbf{X}\beta\|_2^2 + \lambda\|\beta\|_1$, where $\mathbf{X}$ is the feature matrix, β is the coefficient vector, and y is the label vector. λ is the coefficient of ℓ_1 norm which controls the model sparsity. Lasso uses an ℓ_1 norm regularization to shrink the parameters of the majority of features to zero which are trivial, and those variants corresponding to non-zero terms are selected as the identified important features. We set λ to be 0 (without model sparsity) and 0.001352 (with model sparsity selected using the glmnet package in R[43]).

CNN has shown its capability to capture local patterns in genomic data.[27] For demonstration, the CNN model in this study includes one CNN layer, followed by a dense layer as an output layer. To improve model robustness, the ℓ_1 norm is applied to all layers to shrink small weights to zero. We utilize a grid search with 5-fold cross validation to find the optimized hyperparameters. In particular, we use two different learning rates (0.01 and 0.001) and two micro batch sizes (50% and 100% of batch size). Regarding ℓ_2 norm clipping which determines the maximum amounts of ℓ_2 norm clipped to cumulative gradient across all network parameters from each microbatch, we use four unique ℓ_2 norm clipping values (0.6, 1.0, 1.4, and 1.8 respectively). For CNN models, we use two different kernel sizes (5 and 9), and two different numbers of kernels (8 and 16). Furthermore, we set the values of λ as 0 (without model sparsity) and 0.001352 (with model sparsity chosen using glmnet[43]).

4.3. *Implementation of DP*

We implement DP on both Lasso and CNN models with and without ℓ_1 norm respectively, using a Python library called TensorFlow-privacy.[44] DP is implemented in these models by adding a standard Gaussian noise on each gradient of the SGD optimizer. The major process for training a model with parameters θ by minimizing the empirical loss function $L(\theta)$ with differentially private SGD, is summarized as the following: at each step of computing the SGD: 1) compute the gradient $\nabla_\theta L(\theta, x_i)$ for a random subset of examples; 2) clip the ℓ_2 norm of each gradient; 3) compute the average of gradients; 4) add some noise in order to protect privacy; 5) take a step in the opposite direction of this average noisy gradient; 6) in addition to outputting the model, compute the privacy loss of the mechanism based on the information maintained by the privacy accountant.

In the DP implementation, the privacy budget is determined by a function that takes multiple hyperparameters as the input. These hyperparameters include the number of epochs, batch size and noise multiplier. The noise multiplier controls the amount of noises added in each training batch. In general, adding more noise leads to better privacy and lower utility. The hyperparameters used in this study are: two epoch sizes (50 and 100), two batch sizes (8 and 16) and five noise multipliers (0.4, 0.6, 0.8, 1.0, 1.2). We set the value of the parameter δ as the inverse of training dataset size (i.e. $\delta = 0.00066489$).[25]

4.4. *Implementation of MIA*

To train differentially private machine learning models and perform MIA, we split the whole dataset into two disjoint subsets, one as the private target dataset and the other one as the public shadow dataset.[11] We randomly split the public shadow dataset, with 80% used for model training and 20% used to generate the ground truth of the attack model. We focus on a white-box model attack, where the target model's architecture and weights are accessible, to evalute how much privacy will be leaked in the worst case. Hence, the shadow model has the same architecture and hyperparameters as the target model. We use an open-source library of MIA[45] to conduct MIA attacks on the Lasso and CNN models. We build one shadow model on the shadow dataset to mimic the target model, and generate the ground truth to train the attack model. The attack dataset is constructed by concatenating the probability vector output from the shadow model and true labels. If a sample is used to train the shadow model, the corresponding concatenated input for the attack dataset is labeled 'in', and 'out' otherwise. For the attack model, we build a random forest with 10 estimators and a max depth of 2. Each MIA attack is randomly repeated 5 times.

4.5. *Evaluation metrics*

Our evaluation metrics include: (1) the mean accuracy of 5-fold cross validation of the target model on the private target dataset, and (2) the mean of MIA accuracy of 5 MIA attacks. The accuracy of the target model on the training (testing, resp.) data is measured as the precision (i.e., the fraction of classification results that are correct) of the prediction results on the training (testing, resp.) data. We follow the pioneering work[11] and use the *attack accuracy* to measure MIA performance. All samples in the target dataset are fed into the attack model.

5. Results

5.1. *Vulnerability of target model against MIA without DP protection*

We investigate the vulnerability of Lasso and CNN models against MIA for predicting the target phenotype without any DP protection. **Table 1** shows the accuracy of the two target models without DP and attack accuracy of MIA on these models. When the models are not sparse ($\lambda = 0$), Lasso and CNN achieves a similar accuracy on the target dataset (0.7910 vs. 0.7894). The attack accuracy of MIA on Lasso and CNN with no sparsity is 0.5728 and 0.5726 respectively, which is better than random guess (0.5) and on a par with MIA accuracy reported in other areas.[11] The high dimensionality of genomic data makes MIA on genomic

data much harder than other types of datasets, since shadow models hardly mimic the target model on a high dimensional dataset. Nonetheless, with such a MIA accuracy, the adversary still has a chance to infer the membership in a genomic dataset. After introducing model sparsity by adding an ℓ_1 norm ($\lambda = 0.001352$) to coefficients (in Lasso) or weights (in CNN), the target accuracy of both models is slightly improved and their attack accuracy is reduced.

Table 1. **Model performance against MIA (without DP).**

Methods	Target model		Attack model	
	Accuracy	Std.	Accuracy	Std.
Lasso ($\lambda = 0$)	0.7910	0.0123	0.5728	0.0071
Lasso ($\lambda = 0.001352$)	0.7963	0.0157	0.5631	0.0042
CNN ($\lambda = 0$)	0.7894	0.0199	0.5726	0.0059
CNN ($\lambda = 0.001352$)	0.7936	0.0225	0.5628	0.0050

5.2. *Impact of privacy budget on the target model accuracy*

In order to evaluate the impact of DP on the accuracy of the target model, we conduct a grid search to find different privacy budgets and quantitatively investigate the impact of privacy budget. As summarized in **Fig. 2(a)**, we observe that the fitting curve between the privacy budget and the target accuracy can be represented as a log-like curve. The performance of all target models rapidly deteriorates as the privacy budget becomes smaller. When the privacy budget is large, both non-sparse Lasso ($\lambda = 0$) and non-sparse CNN ($\lambda = 0$) models achieve similar target accuracy. Compared with non-sparse models, the target accuracy of sparse Lasso ($\lambda = 0.001352$) and sparse CNN ($\lambda = 0.001352$) models, is downgraded by DP to a more extent even when the privacy budget is large. This is because sparse models only keep coefficients or weights which are higher than λ, and shrink those coefficients or weights that are smaller than λ to 0. Therefore, adding a noise to those large weights will have a more significant impact on the accuracy of the target model.

5.3. *Effectiveness of DP against MIA*

To assess the effectiveness of DP against MIA, we conduct MIA on the target models with different DP budgets. Our results (**Fig. 2(b)**) show that, for Lasso models, the fitting curve between the privacy budget and the target accuracy can be represented as a log-like curve. For CNN, we notice that the curve of attack accuracy is different from that of Lasso, since the attack accuracy becomes unstable when the epsilon is smaller than 10. However, CNN with DP still can provide strong privacy protection. In both Lasso and CNN models, we observe that DP can defend against MIA effectively by perturbing the prediction vector output from the target model, so that the adversary cannot easily infer the membership from such noisy predictions.

According to results in **Fig. 2**, we choose the turning point with a maximum curvature in the log curve as a trade-off between privacy budget and model accuracy. As the privacy

budget becomes tight, the target accuracy is rapidly dropped after this turning point, while the target model with DP can still provide sufficient protection against MIA. Based on this observation, we choose the privacy budget of 10 that best addresses the trade-off between privacy and target accuracy in this study.

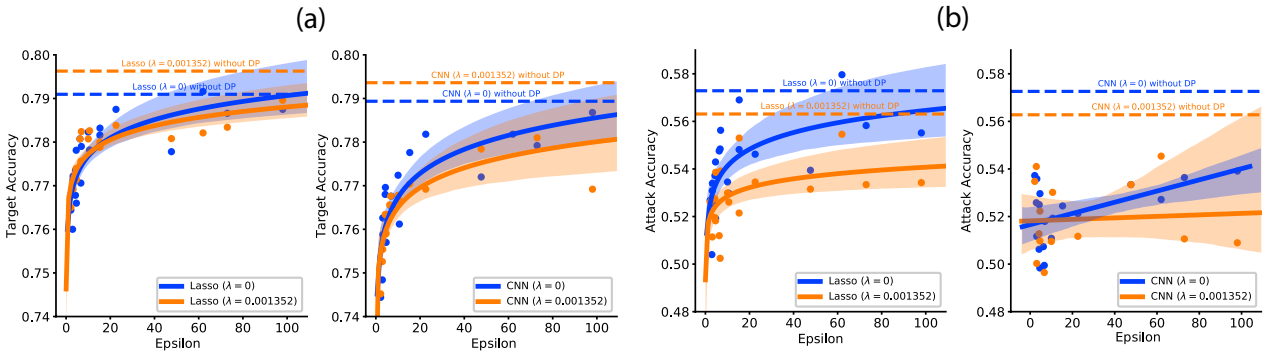

Fig. 2. **Accuracy values of the (a) target model and (b) attack model respectively under various privacy budgets (5-fold cross validation).** Curves indicate the fitted regression lines; shadow areas represent the 95% confidence intervals for corresponding regressions. Horizontal dotted lines represent model performances without DP.

5.4. *Effect of model sparsity*

We investigate the effect of model sparsity by adding an ℓ_1 norm to model coefficients or weights. Due to the large hyperparameter searching space, we only use the value of $\lambda = 0.001352$ for both Lasso and CNN, chosen using the glmnet package.[43] Our results (**Table 1**) show that adding sparsity to a model can improve the accuracy of the target model and reduce the attack accuracy of MIA when DP is not deployed. This is because that on the high-dimensional dataset, a Lasso or CNN model with no sparsity (i.e. $\lambda = 0$) can overfit the training data. However, by introducing model sparsity, the overfitting of the model is reduced, leading to better accuracy of the target model.

We further explore the impact of model sparsity on the accuracy of the target model when DP is deployed. We observe that sparse models with DP have slightly worse model accuracy compared with those non-spare models with DP (**Fig. 2(a)**). This is because each weight in a sparse model is important to prediction results; and any perturbation to these weights can significantly impact model accuracy. We also find that when the privacy budget is smaller than the trade-off (e.g. $\varepsilon < 10$ in our results), the accuracy of the target model is relatively insensitive to model sparsity compared with larger privacy budgets (i.e., $\varepsilon > 10$). Next, we evaluate the impact of model sparsity on the defense power of DP against MIA. As shown in **Fig. 2(b)**, sparse models provide better privacy protection compared with those models without sparsity, given the same DP budget ε.

6. Conclusion

We investigate the vulnerability of trained machine learning models for phenotype prediction on genomic data against a new type of privacy attack named membership inference attack (MIA), and evaluate the effectiveness of using differential privacy (DP) as a defense mechanism against MIA. We find the MIA can successfully infer if a particular individual is included in the training dataset for both Lasso and CNN models, and DP can defend against MIA on genomic data effectively with a cost of reducing accuracy of the target model. We also evaluate the trade-off between privacy protection against MIA and the prediction accuracy of the target model. Moreover, we observe that introducing sparsity into the target model can further defend against MIA in addition to implementing the DP strategy.

Using yeast genomic data as a demonstration, our study provides a novel computational framework that allows for investigating not only the privacy leakage induced from MIA attacks on machine learning models, but also the efficiency of classical defending mechanisms like DP against these new attacks. Nonetheless, there are several limitations of our current study. We are limited to white-box setting where hyperparameters and model architectures are accessible to an adversary in this study. In the future, we will also evaluate black-box access where the adversary simply uses the target model as a black-box for query without any inside information of the model. We will comprehensively explore the relationship between privacy budget and model accuracy, under various combinations of model hyperparameters space and phenotypes. We will apply the framework to analyze large-scale human genomic data where privacy is of a realistic concern. We will investigate whether DP gives unequal privacy benefits to genomes from minority groups compared with those from majority groups. We will investigate other factors (e.g., the number of classes) and conventional genomic analysis (e.g. associations studies, risk prediction) to assess the attack power of MIA and the effectiveness of appropriate defense mechanisms.

References

1. J. C. Lee, D. Biasci, R. Roberts, R. B. Gearry, J. C. Mansfield, T. Ahmad, N. J. Prescott, J. Satsangi, D. C. Wilson, L. Jostins *et al.*, Genome-wide association study identifies distinct genetic contributions to prognosis and susceptibility in crohn's disease, *Nature genetics* **49**, p. 262 (2017).
2. S. Sanchez-Roige, P. Fontanillas, S. L. Elson, A. Pandit, E. M. Schmidt, J. R. Foerster, G. R. Abecasis, J. C. Gray, H. de Wit, L. K. Davis *et al.*, Genome-wide association study of delay discounting in 23,217 adult research participants of european ancestry, *Nature neuroscience* **21**, 16 (2018).
3. R. B. Ness, J. P. Committee *et al.*, Influence of the hipaa privacy rule on health research, *Jama* **298**, 2164 (2007).
4. M. D. Mailman, M. Feolo, Y. Jin, M. Kimura, K. Tryka, R. Bagoutdinov, L. Hao, A. Kiang, J. Paschall, L. Phan *et al.*, The ncbi dbgap database of genotypes and phenotypes, *Nature genetics* **39**, p. 1181 (2007).
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, *arXiv preprint arXiv:1806.01246* (2018).
6. 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 genetics* **4**, p. e1000167 (2008).
7. C. Uhlerop, A. Slavković and S. E. Fienberg, Privacy-preserving data sharing for genome-wide association studies, *The Journal of privacy and confidentiality* **5**, p. 137 (2013).
8. X. Shi and X. Wu, An overview of human genetic privacy, *Annals of the New York Academy of Sciences* **1387**, 61 (2017).
9. D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. C. Courville, Y. Bengio and S. Lacoste-Julien, A closer look at memorization in deep networks, *ArXiv* **abs/1706.05394** (2017).
10. C. Song, T. Ristenpart and V. Shmatikov, Machine learning models that remember too much, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (2017).
11. R. Shokri, M. Stronati, C. Song and V. Shmatikov, Membership inference attacks against machine learning models, in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017.
12. 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, *arXiv preprint arXiv:1802.04889* (2018).
13. S. Truex, L. Liu, M. Gursoy, L. Yu and W. Wei, Demystifying membership inference attacks in machine learning as a service, *IEEE Transactions on Services Computing* , 1 (2019).
14. J. H. Cheon, A. Kim, M. Kim and Y. Song, Homomorphic encryption for arithmetic of approximate numbers, in *International Conference on the Theory and Application of Cryptology and Information Security*, 2017.
15. J. Xu and F. Wang, Federated learning for healthcare informatics, *arXiv preprint arXiv:1911.06270* (2019).
16. C. Dwork, A. Roth *et al.*, The algorithmic foundations of differential privacy, *Foundations and Trends® in Theoretical Computer Science* **9**, 211 (2014).
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 *2019 IEEE Symposium on Security and Privacy (SP)*, 2019.
18. L. Melis, C. Song, E. De Cristofaro and V. Shmatikov, Exploiting unintended feature leakage in collaborative learning, in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019.
19. Y. Wang, J. Wen, X. Wu and X. Shi, Infringement of individual privacy via mining differentially private gwas statistics, in *International Conference on Big Data Computing and Communications*, 2016.
20. Y. Wang, C. Si and X. Wu, Regression model fitting under differential privacy and model inversion attack, in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.
21. M. Nasr, R. Shokri and A. Houmansadr, Machine learning with membership privacy using adversarial regularization, in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018.
22. 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 *In Proceedings of the 2019 Network and Distributed System Security Symposium (NDSS)*, 2019.
23. J. Jia, A. Salem, M. Backes, Y. Zhang and N. Z. Gong, Memguard: Defending against black-box membership inference attacks via adversarial examples, *arXiv preprint arXiv:1909.10594* (2019).
24. N. Phan, Y. Wang, X. Wu and D. Dou, Differential privacy preservation for deep auto-encoders: an application of human behavior prediction., in *AAAI*, 2016.
25. M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar and L. Zhang, Deep learning with differential privacy, in *Proceedings of the 2016 ACM SIGSAC Conference on Com-*

puter and Communications Security, 2016.
26. R. F. Barber, M. Reimherr, T. Schill *et al.*, The function-on-scalar lasso with applications to longitudinal gwas, *Electronic Journal of Statistics* **11**, 1351 (2017).
27. Z. J and T. OG., Predicting effects of noncoding variants with deep learning-based sequence model., *Nat Methods.* **12**, 931 (2015).
28. S. Truex, L. Liu, M. E. Gursoy, L. Yu and W. Wei, Demystifying membership inference attacks in machine learning as a service, *IEEE Transactions on Services Computing* (2019).
29. B. Hilprecht, M. Härterich and D. Bernau, Monte carlo and reconstruction membership inference attacks against generative models, *Proceedings on Privacy Enhancing Technologies* **2019**, 232 (2019).
30. D. Chen, N. Yu, Y. Zhang and M. Fritz, Gan-leaks: A taxonomy of membership inference attacks against gans, *arXiv preprint arXiv:1909.03935* (2019).
31. K. S. Liu, B. Li and J. Gao, Performing co-membership attacks against deep generative models, *arXiv preprint arXiv:1805.09898* (2018).
32. L. Song, R. Shokri and P. Mittal, Membership inference attacks against adversarially robust deep learning models, in *2019 IEEE Security and Privacy Workshops (SPW)*, 2019.
33. J. Hayes, L. Melis, G. Danezis and E. De Cristofaro, Logan: Membership inference attacks against generative models, *Proceedings on Privacy Enhancing Technologies* **2019**, 133 (2019).
34. A. Johnson and V. Shmatikov, Privacy-preserving data exploration in genome-wide association studies, in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013.
35. F. McSherry and K. Talwar, Mechanism design via differential privacy, in *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, 2007.
36. K. Chaudhuri and C. Monteleoni, Privacy-preserving logistic regression, in *Advances in Neural Information Processing Systems*, 2009.
37. A. Patil and S. Singh, Differential private random forest, in *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2014.
38. R. Shokri and V. Shmatikov, Privacy-preserving deep learning, in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015.
39. J. S. Bloom, I. Kotenko, M. J. Sadhu, S. Treusch, F. W. Albert and L. Kruglyak, Genetic interactions contribute less than additive effects to quantitative trait variation in yeast, *Nature Communications* **6**, p. 8712 (2015).
40. J. Chen and C. Nodzak, Statistical and machine learning methods for eqtl analysis, in *eQTL Analysis*, (Springer, 2020) pp. 87–104.
41. J. Chen and X. Shi, A sparse convolutional predictor with denoising autoencoders for phenotype prediction, in *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, 2019.
42. J. Chen and X. Shi, Sparse convolutional denoising autoencoders for genotype imputation, *Genes* **10**, p. 652 (2019).
43. G.-X. Yuan, C.-H. Ho and C.-J. Lin, An improved glmnet for l1-regularized logistic regression, *The Journal of Machine Learning Research* **13**, 1999 (2012).
44. A. Galen, C. Steve and P. Nicolas, Tensorflow privacy: Library for training machine learning models with privacy for training data `https://github.com/tensorflow/privacy`, (2019), Accessed: 2020-01-30.
45. K. Bogdan and Y. Mohammad, Mia: A library for running membership inference attacks against ml models `https://github.com/spring-epfl/mia`, (2019), Accessed: 2020-01-30.