Research Square

Preprints are preliminary reports that have not undergone peer review. They should not be considered conclusive, used to inform clinical practice, or referenced by the media as validated information.

Integrally Private Model Selection for Deep Neural Networks

Ayush K. Varshney (ayushkv@cs.umu.se)
Umeå University https://orcid.org/0000-0002-8073-6784

Vicenç Torra
Umeå University https://orcid.org/0000-0002-3525-0435

Research Article

Keywords: Data privacy, Integral privacy, Deep neural networks, Privacy-preserving ML

Posted Date: May 18th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2944008/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations:
Competing interests: The authors declare no competing interests.

Integrally Private Model Selection for Deep Neural Networks

Ayush K. Varshney*[1] and Vicenç Torra[1]

Department of Computing Sciences
Umeå University, Umeå 90740, Sweden
ayushkv@cs.umu.se, vtorra@cs.umu.se

Abstract. Deep neural networks (DNNs) are one of the most widely used machine learning algorithm. In the literature, most of the privacy related work to DNNs focus on adding perturbations to avoid attacks in the output which can lead to significant utility loss. Large number of weights and biases in DNNs can result in a unique model for each set of training data. In this case, an adversary can perform model comparison attacks which lead to the disclosure of the training data. In our work, we first introduce the model comparison attack for DNNs which accounts for the permutation of nodes in a layer. To overcome this, we introduce a relaxed notion of integral privacy called ϵ-integral privacy. We further provide a methodology for recommending ϵ-Integrally private models. We use a data-centric approach to generate subsamples which have the same class-distribution as the original data. We have experimented with 6 datasets of varied sizes (10k to 7 million instances) and our experimental results show that our recommended private models achieve benchmark comparable utility. We also achieve benchmark comparable test accuracy for 4 different DNN architectures. The results from our methodology show superiority under comparison with three different levels of differential privacy.

Keywords: Data privacy · Integral privacy · Deep neural networks · Privacy-preserving ML.

1 Introduction

In today's world, Artificial Intelligence (AI) plays a crucial role in our day-to-day life. AI techniques are widely used in object recognition, speech recognition, medical imaging, robotics and many other fields. AI approaches and Machine Learning (ML) in particular are very data hungry [1]. They tend to improve with the quality and quantity of data. The data often include sensitive and personal information which must be guarded to ensure security/privacy of each individual or organization. Several guidelines exists such as Europe's General Data Protection Regulation (GDPR), to regulate the use of data in ML. GDPR requires that the analysis to be made should use the minimum amount of data and must

* This work was partially supported by the Wallenberg Al, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

be privacy-preserving. There exists several data masking and privacy-preserving models such as k-anonymity [2], differential privacy [3], integral privacy [4], etc. which try to protect privacy of individuals and organization from any adversaries. Adversaries aim to gain sensitive information about individual or a group of individuals making inferences from ML models.

Data masking is used to modify sensitive information so that a record can not be uniquely identified. K-anonymity is one of the most used data masking methods. A database satisfies k-anonymity if for each record there are k-1 other indistinguishable records. This can be implemented using clustering (replacing k similar records with their mean or with their generalization). In the recent years, much attention has been given to differential privacy (DP) and its variants (see [5] for more details). Differential privacy is satisfied if the outputs of a query on neighbouring datasets are similar i.e. addition or removal of one record should not affect the outcome of the query. Differential privacy depends on a parameter ϵ that establishes the level of this similarity. Theoretically, DP offers sound privacy-preserving models but it has practical limitations such as the amount of noise for small ϵ (high privacy) can be very high. Therefore, high sensitivity queries require high amount of noise. However, in case of multiple queries as the privacy budget is limited, high amount of noise is also required. High noise leads to a loss of utility for ML models. In our approach, we have considered Integral Privacy as an alternative to DP to achieve high utility privacy-preserving machine learning.

Integral Privacy models [4] are the data-driven models that appear recurrently with different training data sets. This makes inferences on sensitive information harder for an intruder. Formally, the set of integrally private models are the set of recurrent models, i.e. generated by different datasets for the same problem. This approach has practical limitations, as in general, we rarely have a huge number of different datasets. The first practical approach for Integral private model selection was given for decision trees [6], where instead of having an available set of datasets, the authors have used sampling approaches to build the model space and eventually suggesting models which are integrally private. The authors expanded the idea with integral privacy guarantees for linear regression. This is given in [7]. In [8], authors have shown how maximal c-consensus meets (see [9] for further details) can be used in the context of integral privacy to find datasets which can produce the same models. The work presented in [6] generates or approximates the model space for a given dataset. A stratified subsampling approach is used to approximate the model space for small datasets ($\approx$ 200 instances). The authors approximate the model space using 100k, 150k and 300k subsamples from each datasets. This can be time consuming and 100-300k subsamples may not be enough to approximate the model space for real-world big datasets. Overall, the approach is computationally expensive.

Deep Neural Networks is one of the most successful machine learning paradigms for several computer vision tasks such as image classification [10], object detection [11], video classification [12], and many other areas. However, DNNs are known to be highly dependent on the input data. In the last few years, interest

in adversarial DNN examples has grown [13]. DNNs are assumed to work well with large datasets. They have large number of weights and biases which can result in very few generators (unique in many of the cases) for each model. In other words, generation or discovery of recurrent models in DNNs is difficult.

Considering these challenges in mind, we introduce a relaxed variant of integral privacy called 'ϵ-Integral Privacy' where models in the ϵ range are considered a perturbated version of each other and, thus, they are considered ϵ-integrally private. We also propose a model selection strategy for choosing ϵ-integrally private models for Deep Neural Networks (DNNs). Our algorithm recommends the mean of the top recurrent models as the private model. We distribute the data in disjoint subsamples having same class-distribution as the original dataset. We find that large enough disjoint subsets having same class-distribution as the original dataset leads to the generation of the models which are utmost ϵ-different, with utility comparable to the benchmark model. This way we do not need to generate 100-300k sub samples. Our approach also supports the data-centric approach [14]. We are able to generate benchmark comparable models with samples sizes 1/100th of the original dataset. There hasn't been much work in the literature which discusses about using smaller datasets for training DNNs. The work in [15] improves the quality of data by eliminating the invalid instances, our approach is focused on maintaining the class-distribution of the data.

In this paper, we have also extended the potential model comparison attack [6] for DNNs. In this type of attack, an intruder gets access to the training data by comparing the models learned by the intruder obtained from original data and the model obtained from a modified dataset. In case of DNNs, the attack becomes tricky as any permutation of the similar set of nodes at any given layer l results in the same learning. We incorporate this to extend the model comparison attack on DNNs.

We have arbitrarily chosen a 3-hidden layered DNN for 6 datasets with varied sizes. Our experimental results show that large enough disjoint sets lead to the generation of ϵ-integral private models with benchmark comparable utility and loss. We get benchmark metrics by training and testing on our chosen DNN on 70-30 split for each data. We have also compared ϵ-integral private models with high DP (differential privacy) model, moderate DP model and low DP model; we found integrally private models have better utility in many cases and have significant improvement in terms of loss for most of the datasets.

This paper is organized as follows. In Section 2 we introduce the model comparison attack for DNNs; In Section 3 we introduce the notion of ϵ-integral privacy and present the algorithm for private model selection procedure for DNNs; In Section 4 we present the experimental analysis to support our claim and in Section 5 we present our conclusion and directions for future work.

2 Model comparison attack for DNNs

In this section, we describe our model comparison attack for deep neural networks. Deep neural networks are machine learning models which were created to

learn like the human mind. The underlying architecture of DNNs consists of the perceptron (or commonly known as neuron) which receives an array of inputs and transform them into output signal(s). DNNs learns from data by putting together a list of layers. Each layer is responsible for learning some relationship or functionality in the input. Each layer is a collection of neurons that learns to detect patterns in the input. Each neuron in the DNNs can be considered as a logistic regression. DNNs are the extension of artificial neural network with two or more hidden layers. In each neuron, the weighted sum of the input with a bias term is computed which is then transformed using an activation function, which is then passed on to the next layer of the DNNs. Nodes at layer l receive input from the nodes at layer $l-1$, which means each neuron has $|l-1|+1$ ($+1$ for bias) number of parameters to be tuned in training. Final weights and biases of each neuron highly depends on their initialization.

2.1 Framework

In this section, we propose our framework. Let X be the training set from the original dataset D, $\mathcal{G}$ be the model generated on X. In our work, we have considered DNNs as learning algorithm. Let us denote an initial architecture and weight by $Arch$ and let A be the algorithm.

We assume the intruder has some background knowledge $S^* \in D$. They are the records that are known to be used to train the model. The intruder also has access to the model. That to $\mathcal{G}$ which was learned from the training set X on the initial architecture $Arch$. That is, $\mathcal{G} = DNN(Arch, X)$. With this information, the intruder aims to gain knowledge on the training set and do membership inference attacks

The intruder essentially can perform the model comparison attack once they can generate the model space associated to S^*. The intruder can perform comparison with the models in model space and his knowledge of G. After comparison, if there is a single generator for the model, the intruder gets complete access to the training set and their inferences. If there are more than one generator for the model, an intruder can do membership inference attack for dominant records by finding the intersection between the generators.

2.2 Intruders Approach

The intruder has some background information S^*. Then, they can draw a block of subsamples $S = \{S_1, S_2, ..., S_n\}$ where $S_i \subseteq S^*$ to generate the (approximated) model space. Each subsample is a set of instances from S^* which are used to generate a DNN (see Fig. 1). Generation of the complete model space can be computationally expensive but can be approximated using sampling approaches.

Comparison of two DNNs for model comparison attack is a difficult task because we need to deal with a combinatorial problem. We need to align neurons in each layer. Observe that layers in both DNNs must contain the same neurons i.e. for two DNNs to be the same they must have equal layers; and for two layers to be equal, neurons in one layer must be some permutation of the neurons in the other layer. Given r neurons, we will have r! possible permutations.

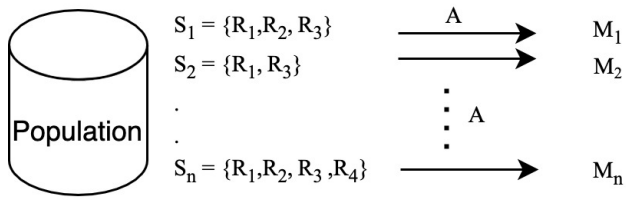


Fig. 1: Demonstration of model generation using algorithm A for subsamples $S_1, S_2, ..., S_n$.

Each model in the generated model space can be compared with the original model G. In case of DNNs, each model has one or very few generators due to the high number of parameters of the model. Therefore, after the comparison attack, the intruder may be able to uniquely identify the training set used to generate the model. When there are more than one generator for a model G, an intruder can check for membership inference by finding the dominant records from the intersection of the generators for the model.

2.3 Integral Privacy

This privacy model [4] aims to protect the disclosure of training data and inferences from a model comparison attack. Let A be an algorithm to compute model G from a given population of samples P. The model G is integrally private if it can be generated by enough number of samples from the population. Let S^* be the background information available to the intruder, then $Gen^*(G, S^*) = \{S' \setminus S^* \mid S^* \subseteq S' \subseteq P, A(S') = G\}$ is the possible set of generators for the model G. K-anonymous integral privacy holds when there are at least k disjoint generators in the set $Gen^*(G, S^*)$. Disjoint generators are required to avoid membership inference attacks. Formal definition for Integral privacy is as follows.

Integral privacy. Let P be the set of samples or a dataset. For model $G \in \mathcal{G}$ generated by algorithm A on samples $S \subseteq P$, let $Gen^*(G, S^*)$ represent the set of all generators of G which are consistent with the background knowledge S^*. Then, the model G is said to be k-anonymous integrally private if $Gen^*(G, S^*)$ contains at least k sets of generators and

$$\bigcap_{S \in Gen^*(G,S^*)} S = \emptyset \tag{1}$$

3 ϵ-Integrally private model selection for DNNs

To construct the complete model space is computationally intractable for large sets. Consider an example of a dataset with 5000 instances. Considering all possible datasets to produce all possible models of the model space (say M_c)

corresponds to producing 2^{5000} generators and the corresponding models. The alternative to M_c is to construct an approximation of the model space (M_e) using sampling. This approach was used in previous works [6] [7]. Nevertheless, even in this case the number of generators and their corresponding models can be high and computationally expensive. In case of bigger datasets say with 5 million instances, the process of building an approximation of a model space will be very costly. In our approach, we have focused on reducing the huge computational requirement to recommend relaxed integrally private deep neural network models.

Let us consider the problem of finding the set of different models of the model space. First, let us recall that each neuron at layer l in DNNs receive inputs from all the neurons in layer $l-1$, which in turn require weights and bias for the neuron. The weights and biases in DNNs can take any value between -1 and +1. Even for a small DNN there can be a unique generator for each model or only very few models will have more than one generator. Our initial studies on DNNs confirms this even when we round-off weights to 3 digits. It is worth mentioning here that initialization of DNNs also affects the number of generators. More concretely, we may not get the same generators on differently initialized models. This makes achieving integral privacy difficult.

Because of this in our approach, we have adopted the relaxed version of integral privacy which we call 'ϵ-Integral privacy' in which models utmost ϵ different from each other are considered. In case of DNNs, two models are utmost ϵ different if and only if the difference between weights for same connection between neurons is at most ϵ. In case of DNNs, two models are utmost ϵ different if and only if the difference between weights for the same connections between neurons is always less than ϵ I.e. if $G1,\ G2$ represent the weights for two DNNs then $||G_1 - G_2|| \leq \epsilon$, where $||G_1 - G_2||$ represent the difference between every same connection between neurons for both DNNs. Now, let $Gen^*(G, S^*, \epsilon)$ denote the set of possible pairwise disjoint generators for the models which are utmost ϵ different than G (generators that are consistent with the background knowledge S^*), then k-anonymous ϵ-Integral privacy holds if $Gen^*(G, S^*, \epsilon)$ has at least k elements and their intersection is empty. A more formal definition follows.

ϵ-Integral privacy: Let P be the set of samples or datasets. For a model $G \in \mathcal{G}$ generated by algorithm A on samples $S \subseteq P$, let $Gen^*(G, S^*, \epsilon)$ represent the set of all generators of G which are consistent with the background knowledge S^* and are utmost ϵ different. Then, the model G is said to be k-anonymous ϵ-Integrally private if $Gen^*(G, S^*, \epsilon)$ contains at least k elements and

$$\bigcap_{S \in Gen^*(G, S^*, \epsilon)} S = \emptyset \tag{2}$$

Now, we will focus on the private model selection procedure for DNNs. Our approach to generate subsampling is data centric. We choose a subsample of size N with same class-distribution as the original dataset D. We denote these subsamples by $S_1, S_2, ..., S_n$ (here $n = \lfloor |D|/N \rfloor$). With this, we also satisfy there is no intersection between subsamples i.e. $S_1 \cap S_2 \ \cap ... \cap \ S_n \ = \emptyset$. This condition

Algorithm 1 Integrally private model selection procedure for Deep Neural Networks for a given perturbed dataset D'. The algorithm return top 5 integrally private models with their accuracies

```
Inputs: D - Perturbed Dataset
N - Size of subsamples
ϵ - Privacy parameter
A - Algorithm to generate DNNs
Output: returns a list of integrally private models with their accuracies
Algorithm:
S = Generate_subsample(D, N)                ▷ Generate n subsamples of size N
ModelList = [[]]
for S_i in S do
    M_i ← A(S_i)
    present = False
    for each m_j ∈ ModelList do
        if compare_model(m_j, M_i) ≤ ϵ then
            ModelList[j].append(M_i)
            present = True
            break
        end if
    end for

    if present == False then
        ModelList.append(list(M_i))
    end if
end for
chosen_models = choseXModels(ModelList)
                                            ▷ Chose top X recurring models
meanModels = A(mean(chosen_models))         ▷ Compute mean models
statistics = computeMetrics(meanModels)     ▷ Statistics of mean models
return meanModels, statistics
```

is important to avoid membership inference attack from the intersection analysis between generators.

Now, we propose our algorithm for choosing integrally private models for DNNs. Its flowchart is given in Fig. 2. The algorithm is as follows for a given dataset D. First, we generate n subsamples each of size N having the same class-distribution as the original. Second, we compute models and cluster them so that each cluster has models that are utmost ϵ different from each other. Finally, we can choose a cluster of models which are recurring in nature and has high utility. In our methodology, we chose the mean of all the models in the cluster as our recommended model. I.e. we generate a new model whose weights are the mean of the weights of all the ϵ-integrally private models. This is our recommended model.

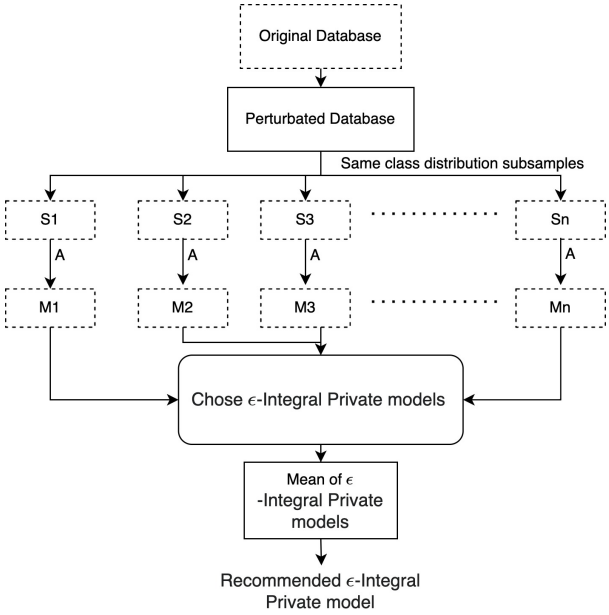


Fig. 2: Flowchart of the proposed methodology to recommend an ϵ-integral private model.

Algorithm 1 formalizes this approach. In the algorithm we have a dataset D, Algorithm A, privacy parameter ϵ and size of each subsample N as inputs. We initialize an empty list of lists and append models which are utmost ϵ distant apart from the first one. For our results we can either chose the top recurring model or X most frequent models (for more ambiguity) which is done in function choseXModels(). Our recommended model is the mean of the models in the cluster. For X $\epsilon-$ranged models, we recommend X mean models and their statistics as the output of our proposed algorithm.

4 Experimental Results

In this section, we present our experimental results for our proposed methodology. Our approach is valid for both numerical/categorical data and for classification problems with an arbitrary number of classes. Table 1 shows the details of the datasets we have considered for our experiments namely Adult, Susy, ai4i and HepMass from UCI repository [16]; and Churn_Modelling, Diabetes [17]. Of these datasets, Churn_Modelling and Adult have categorical data and Diabetes is a multi-class problem. We have considered small datasets ($\approx$ 10-50K instances), medium dataset ($\approx$ 250K instances) and large datasets ($\approx$ 5-7 million instances) for our experimental study. Table 1 also shows the size of the subsamples. The size is chosen so that there are enough subsamples to find integrally private models.

Dataset	# instances	# attribute	Data type	# classes	subsample size
Adult	48842	14	Categorical Integer	2	1000
Susy	5000000	18	Real	2	10000
ai4i	10000	14	Real	2	500
HepMass	7000000	28	Real	2	10000
Churn Modelling	10000	21	Categorical Real	2	500
Diabetes	254000	21	Real	3	5000

Table 1: Details of the used Datasets

To compare the performace of our approach and 2 benchmark, we have used an architecture of 5-layered DNN with 3-hidden layers with 5-10-5 neurons. As we explain later, we have considered other architectures as well. Then, we have taken $\epsilon = 0.05$ for all the datasets, other values could be used depending on the application requirements.

The results of our methodology have been compared with results with a differential private solution [18] and the benchmark results. Benchmark results are obtained by training the model with 70-30 train-test split of original dataset. Now, let us look at the number of generated models from randomly chosen subsamples of the size given in Table 1. In case of the adult dataset, the total possible models which can be considered for integral privacy are 47, similarly for ai4i dataset we have 19, for susy dataset we have 498, for hepmass dataset we have 698, for churn modelling dataset we have 18, and for diabetes dataset we have 49 models to be considered for integral privacy.

Fig. 3 shows the training f1 score of top 5 (for ai4i and Churn Modelling datasets there are 2 and 3 generators only) recurring models along with the training score of the benchmark model in black solid line and three level of differential privacy(DP): high privacy ($\epsilon \approx 0.1$, represented by ◆), moderate privacy ($\epsilon \approx 0.5$, represented by $\cdot-$) and low privacy ($\epsilon \approx 1.0$, represented by •). In general, higher DP privacy (low ϵ, ◆) leads to lower training score and higher training loss. In the plots, the f1 scores of all the models are in the light shade, and the dark solid line represents the mean of the ϵ ranged integral private models. Observe from Fig. 3a and 3b, we achieve better training score than the benchmark training scores while from Fig. 3c, 3d, 3e and 3f we can observe benchmark comparable results. It can be seen from Fig. 3a, 3c and 3d, integrally private models have better training score than all three variants of differentially private models on the other hand Fig. 3b, 3e and 3f, the training utility of integrally private model is comparable with the differentially private models. We get similar results for the training loss as shown in Fig. 4. We have denoted the loss of each model in the lighter shade solid line, their mean loss in dark solid line, the benchmark model loss with solid black line and three level of differential privacy: high privacy with ◆, moderate privacy with $\cdot-$ and low privacy with •. It can be seen that the loss for integrally private models is comparable with

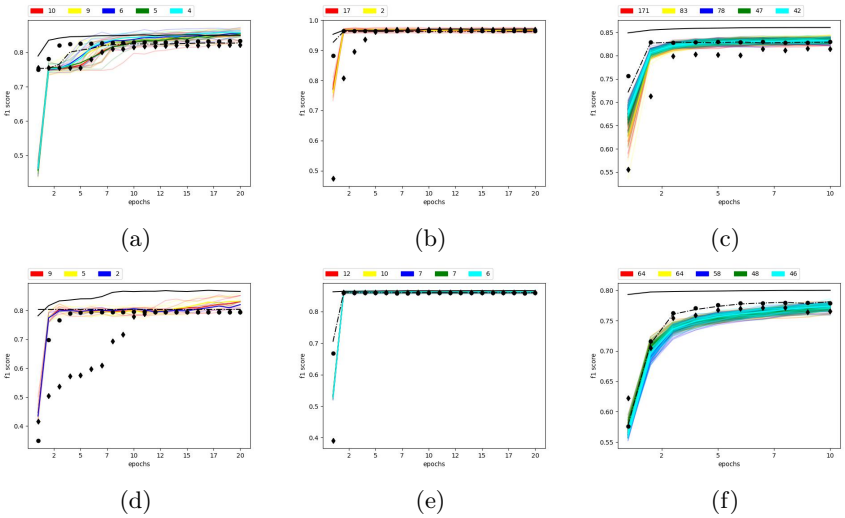


Fig. 3: f1 score of top 5 ϵ-recurring models over training data for (a) Adult (b) ai4i (c) HepMass (d) Churn Modelling (e) Diabetes (f) Susy Datasets

the benchmark model loss. We can observe from Fig. 4b, 4c and 4d, integrally private models have significant improvement in terms of training loss from DP variants while Fig 4a, 4e shows some improvement from DP variants in contrast to Fig. 4f where low, moderate DP privacy has improvement in training loss from integrally private models.

The concept of data-centric AI simply suggests that good quality of data can lead to good models. In our approach, we have only used 0.15% to 2% of the original data, but with the same class-distribution, to train our model (see table 1 for subsample size). We got surprising result when we compared their performance on test data i.e. 30% of the original data. Fig. 5 shows the result on the test data, lighter shade circles represent the test result for each model while dark solid colored circle represents their mean value. From Fig. 5, we can say that our ϵ-integrally private models achieve benchmark comparable f1 score on much bigger test datasets (15 to 200 times).

Our recommended model is the mean of all the models in the ϵ-integral private range. The result in Fig 5 motivated us to compare performance of the aggregated ϵ-integrally private models with the original training and testing datasets. Fig. 6 shows the comparison of f1 score on training data (in solid color circles) and test data (in hollow circles) with benchmark training score (in solid line) and benchmark test score (in dashed line). Our recommended models have benchmark comparable f1 score on all the datasets.

Table 2 shows the recurrence of the recommended model with the test accuracy on much bigger test sets. We have considered 4 different architectures: DNN-1 has 3-hidden layers (with 5-10-5 neurons respectively) architecture; DNN-2

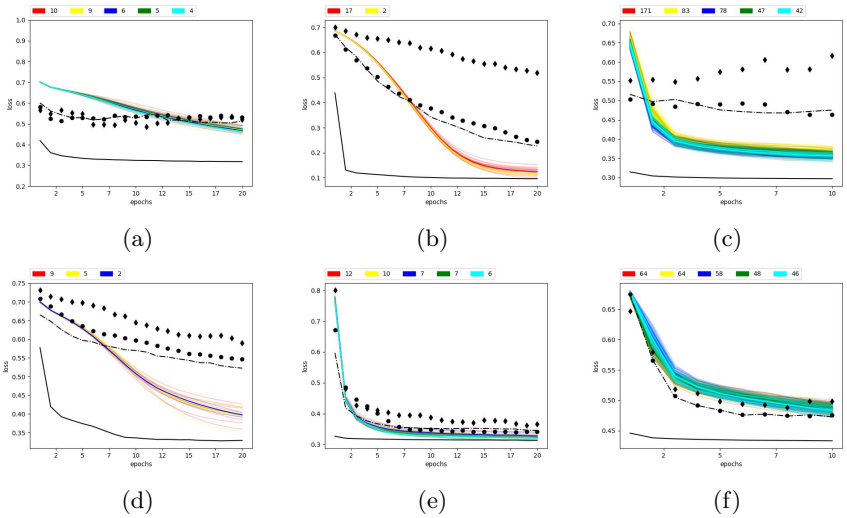


Fig. 4: Training loss of top 5 ϵ-recurring models for (a) Adult (b) ai4i (c) HepMass (d) Churn Modelling (e) Diabetes (f) Susy Datasets

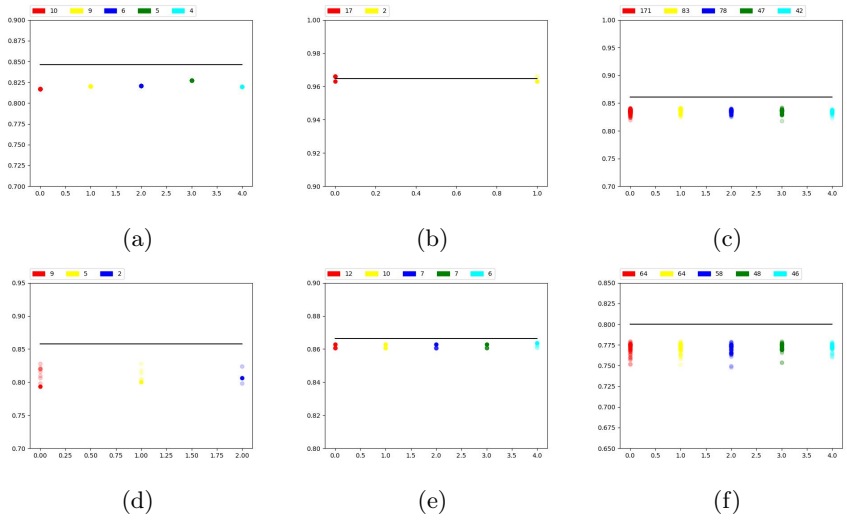


Fig. 5: f1 score of top 5 ϵ-recurring models on bigger test data for (a) Adult (b) ai4i (c) HepMass (d) Churn Modelling (e) Diabetes (f) Susy

has 1- hidden layer (with 1024 neurons) architecture; DNN-3 has 3-hidden layers (with 10-20-10 neurons respective) architecture; and DNN-4 has 5-hidden layers (with 5-10-20-10-5 neurons respectively) architecture. Table 2 shows that

Dataset	DNN-1		DNN-2		DNN-3		DNN-4	
	recurrence	test_acc	recurrence	test_acc	recurrence	test_acc	recurrence	test_acc
Adult	10	0.8387	89	0.7797	16	0.8286	36	0.8284
Susy	64	0.7758	366	0.7917	8	0.7636	6	0.7882
ai4i	17	0.9647	19	0.9723	12	0.9683	10	0.9747
HepMass	171	0.8325	562	0.8344	68	0.8325	51	0.8336
Churn Modelling	9	0.8145	13	0.8520	10	0.7927	10	0.7870
Diabetes	12	0.8627	21	0.8596	13	0.8634	5	0.8596

Table 2: Different architectures and their f1 score on 30% test dataset.

the proposed methodology produces benchmark comparable results for different DNN architectures as well.

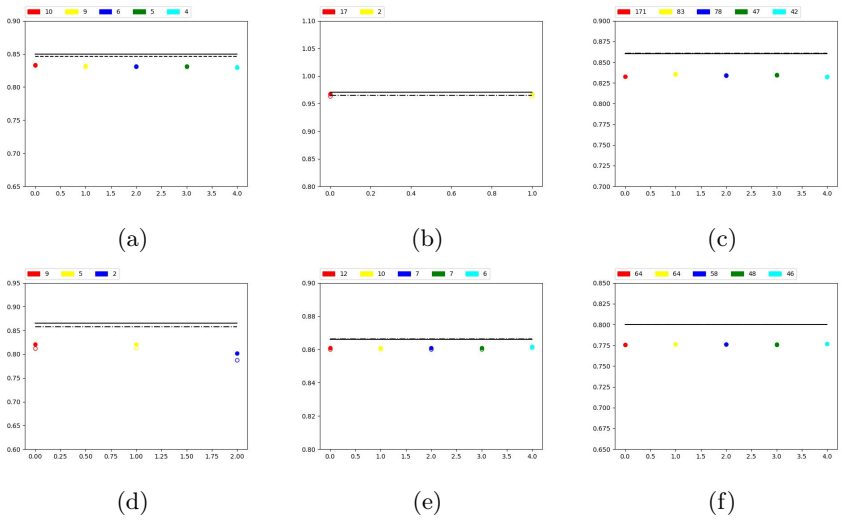


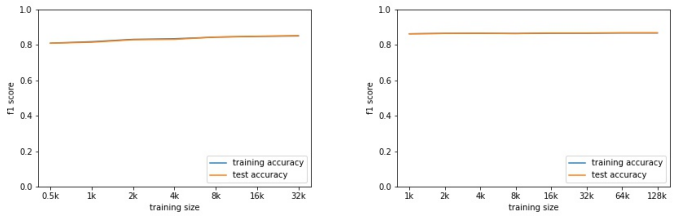
Fig. 6: f1 score on train and test data for mean of the ϵ-recurring models for (a) Adult (b) ai4i (c) HepMass (d) Churn Modelling (e) Diabetes (f) Susy

4.1 Discussion

In summary, our results with varied sized, multi-class and categorical datasets suggest that we can achieve ϵ-integral privacy with good utility (comparable to benchmark utility) from the list of the recommended models depending on the value of k (number of models in ϵ range) with no additional computational cost.

The good results of our approach can essentially be linked to the data centric AI approach where we train our model for smaller datasets with the same class-distribution as the original dataset and get good results. We further explored

the impact of subsample size and compared their performance on separate 70-30 training data and testing data on moderately sized adult and diabetes datasets. Our results from Fig. 7 shows that the f1 score for both training and testing data is non-decreasing but it is neither increasing significantly with respect to the increase in subsample size. Our results are in line with [19] which highlights that one can generate arbitrarily similar model of finite floating point weights from two (or more) non-overlapping dataset. The paper [19] also suggest that we can get good results on smaller datasets as well, which aligns with the results in Fig. 7.



(a) (b)

Fig. 7: f1 score of various subsample sizes on (a) Adult (b) Diabetes datasets

For our proposed methodology, we must chose subsamples size (N) very carefully. The choice for N must be large enough to generate the model with good utility at the same time it should be able to generate sufficient number of disjoint subsamples. Probably approximately correct (PAC) [20] can suggest an estimate for the choice of the parameter N. A model G is said to be PAC learnable with respect to loss l if and only if the difference between the loss for the learned model G and true (best possible) model $\bar{G}$ is at most ϵ with probability at least $1-\delta$ i.e. $P[G_l - \bar{G}_l \leq \epsilon] \geq 1-\delta$. With this the minimum number of samples required for a PAC learnable model is bounded by $O([VC(G) + ln(1/\delta)]/\epsilon^2)$ [21] where $VC(G)$ is the Vapnik–Chervonenkis dimension of the model G. Quantifying the VC-dimension for complex models like deep neural network is still an open problem [22]. Therefore, in the literature scientists follow the rule-of-thumbs: (1) The VC dimension of DNNs is considered equal to the number of weights in DNNs [23] and then (2) the minimum number of samples required to learn the DNN is established as 10 times the VC dimension [24]. Considering this, i.e., a sample size of 10-times the VC-dimension (number of weights) should provide a PAC learnable model. For datasets ai4i, and Churn_Modeling the number of weights are 172 and 197, respectively, and hence the minimum subsample size is estimated as 1720 and 1970 for PAC learnability. This results in very few disjoint subsamples (5 for both datasets) which may not be enough to find integrally private models. This suggests a trade-off between model complexity (number of

weights) and its learning ability for integral privacy. Further study in this area is required to investigate the impact of this trade-off for integral privacy.

4.2 Limitations:

Based on a critical analysis of our approach and the results obtained, we can underline the following limitations of our approach:

1. Our methodolgy may not be suitable in the presence of outliers as the outliers disturbs the distribution of the dataset.
2. Selection of private models on very small datasets with our proposed methodology is not feasible.
3. High model complexity may result in less number of models in ϵ-range.

5 Conclusion and Future work

In this paper, we have first extended the model comparison attack to deep neural networks. We have also introduced the concept of ϵ-integral privacy which is then used to recommend integrally private models for deep neural networks. Our results show that we are able to achieve ϵ-integrally private models without any significant utility loss (improvement of utility in some cases). Our results also highlights that small data of good quality can result in a well trained model.

For our proposed methodology, we have arbitrarily chosen the size of the subsamples; the privacy parameter ϵ and the DNNs architecture. Tuning of these areas may yield interesting results. Another interesting direction is to use a data-enhancement approach to remove outliers as done in [15]. Federated Learning takes advantage of data distributed across multiple users, where learning takes place locally. Our methodology can be seen as independent and identically distributed (IID) ϵ-integral private model selection in federated learning for a single pass. Our work can further be extended into non-IID settings of federated learning.

References

[1] Z. Obermeyer and E. J. Emanuel, "Predicting the future—big data, machine learning, and clinical medicine," *The New England journal of medicine*, vol. 375, no. 13, p. 1216, 2016.

[2] P. Samarati, "Protecting respondents identities in microdata release," *IEEE transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010–1027, 2001.

[3] C. Dwork, "Differential privacy," in *Automata, Languages and Programming*, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–12.

[4] V. Torra and G. Navarro-Arribas, "Integral privacy," in *International Conference on Cryptology and Network Security*, Springer, 2016, pp. 661–669.

[5] Z. Ji, Z. C. Lipton, and C. Elkan, "Differential privacy and machine learning: A survey and review," *arXiv preprint arXiv:1412.7584*, 2014.

[6] N. Senavirathne and V. Torra, "Integrally private model selection for decision trees," *computers & security*, vol. 83, pp. 167–181, 2019.

[7] N. Senavirathne and V. Torra, "Approximating robust linear regression with an integral privacy guarantee," in *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, IEEE, 2018, pp. 1–10.

[8] V. Torra, G. Navarro-Arribas, and E. Galván, "Explaining recurrent machine learning models: Integral privacy revisited," in *International Conference on Privacy in Statistical Databases*, Springer, 2020, pp. 62–73.

[9] V. Torra and N. Senavirathne, "Maximal c consensus meets," *Information Fusion*, vol. 51, pp. 58–66, 2019.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[12] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang, "Exploiting feature and class relationships in video categorization with regularized deep neural networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 2, pp. 352–364, 2017.

[13] C. Oh, A. Xompero, and A. Cavallaro, "Visual adversarial attacks and defenses," in *Advanced Methods and Deep Learning in Computer Vision*, Elsevier, 2022, pp. 511–543.

[14] A. Ng, "Mlops: From model-centric to data-centric ai," *Online unter https://www. deeplearning. ai/wp-content/uploads/2021/06/MLOps-From-Model-centric-to-Data-centricAI. pdf [Zugriffam09. 09.2021] Search in*, 2021.

[15] M. Motamedi, N. Sakharnykh, and T. Kaldewey, "A data-centric approach for training deep neural networks with less data," *arXiv preprint arXiv:2110.03613*, 2021.

[16] D. Dua and C. Graff, *UCI machine learning repository*, 2017. [Online]. Available: `http://archive.ics.uci.edu/ml`.

[17] C. for Disease Control, Prevention, *et al.*, *National diabetes statistics report, 2017. atlanta, ga: Centers for disease control and prevention; 2017*, 2015.

[18] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*, Springer, 2006, pp. 265–284.

[19] A. Thudi, H. Jia, I. Shumailov, and N. Papernot, "On the necessity of auditable algorithmic definitions for machine unlearning," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 4007–4022.

[20] V. N. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," in *Measures of complexity*, Springer, 2015, pp. 11–30.

[21] R. Vershynin, *High-dimensional probability: An introduction with applications in data science*. Cambridge university press, 2018, vol. 47.

[22] M. Anthony and P. Bartlett, *Neural network learning: Theoretical foundations*. cambridge university press, 1999.

[23] Y. S. Abu-Mostafa, "Hints," *Neural computation*, vol. 7, no. 4, pp. 639–671, 1995.

[24] E. Baum and D. Haussler, "What size net gives valid generalization?" *Advances in neural information processing systems*, vol. 1, 1988.