Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Integrally private model selection for decision trees

Navoda Senavirathne[a,*], Vicenç Torra[a,b]

[a] School of Informatics, University of Skövde, Högskolevägen, Skövde, Sweden
[b] Maynooth University Hamilton Institute, Eolas Building, North Campus, Maynooth, W23 A5Y6, Co. Kildare, Ireland

ARTICLE INFO

Article history:
Received 15 March 2018
Revised 23 November 2018
Accepted 18 January 2019
Available online 29 January 2019

Keywords:
Integral privacy
Privacy models
Data privacy
Machine learning model space
Privacy preserving machine learning

ABSTRACT

Privacy attacks targeting machine learning models are evolving. One of the primary goals of such attacks is to infer information about the training data used to construct the models. "Integral Privacy" focuses on machine learning and statistical models which explain how we can utilize intruder's uncertainty to provide a privacy guarantee against model comparison attacks.

Through experimental results, we show how the distribution of models can be used to achieve integral privacy. Here, we observe two categories of machine learning models based on their frequency of occurrence in the model space. Then we explain the privacy implications of selecting each of them based on a new attack model and empirical results. Also, we provide recommendations for private model selection based on the accuracy and stability of the models along with the diversity of training data that can be used to generate the models.

1. Introduction

Most of the real world data are "dynamic" and thus subject to be updated on a regular basis. This impacts the conformity of the aggregations and inferences extracted unless they are updated consequently. For example, a machine learning model built on a dynamic data source needs to be updated, so that the model will be in agreement with the data source. Modifications to training data could cause the machine learning models to transform into different ones. An intruder who has access to some auxiliary information can try to infer the cause for such transformation with respect to the set of modifications. Integral privacy (Torra and Navarro-Arribas, 2016) discusses how model transformation could intrinsically bring up disclosure risk to the underlying training data and the set of modifications applied to the training data. The privacy model further discusses desirable characteristics a machine learning model should have in order to avoid such disclosures. The basic idea is that an intruder should not be able to learn about the training data or the set of modifications by comparing machine learning models generated before and after a particular modification.

In this paper, our primary focus is to provide recommendations for machine learning model selection so that the selected models are compliant with the integral privacy. For model selection, predictive accuracy is used as the principal criterion. However, with the increased usage of sensitive data, and the need for collaborative data analysis (i.e., multiparty computations), the degree of "privacy" a model provides over its underlying training data has become an important factor. With the evolution of attacks targeting the machine

[*] Corresponding author.
E-mail addresses: navoda.senavirathne@his.se (N. Senavirathne), vicenc.torra@mu.ie (V. Torra).

https://doi.org/10.1016/j.cose.2019.01.006

learning models, in order to infer information about the training data and their properties, privacy has become an inevitable requirement.

In our work, we explain a potential attack model against integral privacy which we term as model comparison attack. Then we study the "space of models" to understand the relationship between models and their training data, with the intention of exploiting it in order to achieve integral privacy against model comparison. Based on our observations of the model space, we identify two categories of models based on their frequency of occurrence. Then, privacy implications of adopting each type of models are discussed in detail.

We generate an empirical distribution for models and then experimentally show how feasible this approach is. In literature, we have not come across any previous attempts to understand the distribution of models or use it to attain privacy. Most of the existing privacy models use perturbation techniques to gain privacy. And this results in poor model utility (low predictive accuracy) which is undesirable. We provide recommendations for model selection that can be used to minimize this adversarial impact. The recommended models are already (naturally) available in the model space and based on our empirical results we show the accuracy levels of those models also remain high. Hence we propose this as a favourable approach for model selection.

Our experiments are based on decision trees; due to their intuitive representation, it is easy to understand and compare the models when building the model space. CART[1] algorithm (Breiman, 1984) available on R is used for the experiments.

The paper is organized as follows. In Section 2 we review some relevant background knowledge we use in this paper. In Section 3 we introduce model comparison attacks. Section 4 is designed to present our methodology along with the empirical results with reference to the machine learning model space. In Section 5 we focus on recommendations for model selection based on integral privacy, whereas Section 6 is reserved for discussing related work. Finally, we conclude the paper with Section 7 discussing the conclusions, limitations and future work.

2. Background

2.1. *Integral privacy*

Integral privacy (Torra and Navarro-Arribas, 2016) is a privacy model that focuses on machine learning and statistical models and about the inferences we can make from them. The goal is to maximize the uncertainty of the intruders with respect to the original data or modifications of the data once they have access to the models. More specifically the privacy model assumes that the intruder has access to two machine learning models, the algorithm used to generate those models and some background information about the training data. We review the formal definition of integral privacy below.

Consider two data sets X and X'. X is the original data set and X' is the resulting dataset when some modifications μ are applied to X. Let us denote this by $X' = X \oplus \mu$. Using a machine learning algorithm A, on X and X' two models G and G' are generated as $G = A(X)$ and $G' = A(X')$. If the machine learning algorithm A satisfies integral privacy, then the set of modifications $\mathbb{M}$ an intruder can infer from G and G' should be large and in addition $\cap_{m \in \mathbb{M}} m$ is empty. In this work, we only consider record addition and suppressions as possible modifications.

In order to formalize integral privacy, we first need to formalize the set of modifications. Let us consider a reference set P (in our case it corresponds to a set of records). Then let p^+ denote the elements of the set P prefixed with a "+" and p^- denote the elements of the set P prefixed with "-". That is for $p \in P$, $+p$ denotes an addition of p, while $-p$ denotes a deletion. Let $\bar{P} = p^+ \cup p^-$ the set of possible modifications. Using this notation we can define the operation $S_i \oplus \mu$ for any $S_i \subseteq P$ and $\mu \subseteq \bar{P}$ as follows:

$$S_i \oplus \mu = \{p \mid p \in S_i \wedge -p \notin \mu\} \cup \{p \mid +p \in \mu\}$$

Similarly, we can define for any pair $(S_i, S_z) \subseteq P$

$$S_i \ominus S_z = \{-p \mid p \in S_i \wedge p \notin S_z\} \cup \{+p \mid p \in S_z \wedge p \notin S_i\}$$

Then given the assumption that intruder has access to G, G', $S \subseteq X$ (partial knowledge about the data used to build G), and knowledge about the machine learning algorithm A_j the privacy model is about avoiding inferences on μ, X, and X'. Formally, the set of possible modifications μ is the set defined by, $\mathbb{M} = \bigcup_{g \in Gen,\, g \in Gen'} \{g' \ominus g\}$ where,

$$Gen = \{S_G | S \subseteq S_G \subseteq P, A(S_G) = G\}$$

and,

$$Gen' = \{S'_G | S'_G \subseteq P, A(S'_G) = G'\}.$$

k-Anonymous integral privacy is when the set $\mathbb{M}$ contains at least k minimal elements. The rationale of this definition is, by using some background knowledge an intruder should not be able to determine with high confidence, what modifications (μ) are carried out on training data in order to generate a given machine learning model. The larger the size of $\mathbb{M}$, the more difficult is to determine what set of modifications led to a particular model.

2.2. *Approximating the model space*

In order to build integral privacy compliant machine learning models, it is essential to understand the distribution of models in the model space. The first step towards this is to construct the model space for a given dataset. However, the biggest challenge is that the model space is vast and there exists an exponential number of training sets that can be used to generate the models. Due to the complexity of explicitly finding the model space M_C we will try to approximate it by empirically obtaining M_E. To understand the true distribution of the model space, we would have to generate machine learning models for all possible training sets. If we consider the number of observations in a datafile as n (or the population), there

[1] https://CRAN.R-project.org/package=rpart.

exist $2^n - 1$ possible training sets. Given the large size of the space of training sets, exploring the entire space is impossible. To address this issue, we use a sampling method inspired by "subsampling" which can be used to approximate the distribution using smaller subsets of data. The method is especially suited when the original population is either infinite or finite but very large to make any assumptions about the distribution. This idea was first introduced by Politis et al. (1999, 2001). This non-parametric sampling method does not make any prior assumptions about the sampling distribution.

In this section, we review several approaches that can be used to build the model space, including Politis et al.'s subsampling method and "stratified subsampling" method which we use to construct the empirical model space (M_E) for a given dataset in order to understand its probability distribution.

1. Constructing the entire model space (M_C) - This approach extracts all possible subsamples of data records from P and then builds the model space from that. Number of total combinations (samples) can be obtained using unique combinations (nCr) with varying r sizes from 2 - n. E.g. $nC2+nC3+\ldots+nCn-1+nCn$. The minimum subsample size is selected as 2, assuming that there should be at least two different data records which belong to two different classes in order to build a valid decision tree. If this approach is taken the entire model space M_C, will be created. The number of unique models created will be less than the number of different samples because a) Some samples would result into invalid models (e.g., all the selected records belong to the same class) and b) some different samples can result into the same model. As mentioned above this approach is computationally expensive.
2. "Subsampling"- This is used for approximating the sampling distribution based on sample size b, which is smaller than the original dataset size n. In total N subsamples are supposed to be extracted from original dataset without replacement. A data subsample is a subset chosen from the original distribution. Subsample size b is much less than n: $b \ll n$.

   The sampling method we use in this paper is inspired by subsampling. But instead of having a fixed size for b, we vary it randomly. If empirical model space (M_E) is to be constructed by using Politis et al.'s method, deciding on an optimal subsample size (b) will be challenging. Because the selected subsample size should be able to partition a given dataset into blocks, in a way every block of data contribute towards building a valid machine learning model and also it should be able to build the model space (M_E) to have a sufficient representation over the complete model space (M_C).
3. "Stratified subsampling" - We use a subsampling method based on stratified sampling to extract a k number of unique samples with varying sizes. Sampling is stratified with respect to the sizes. We approximately obtain the same number of subsamples for each size between 2 to $|n|-1$. Value k is the size of the empirical sample space and its definition is left to the user. The higher this value, the more representative the empirical model space M_E in relation to M_C. Determining subsample sizes and extraction of subsamples are both random operations. The uniqueness of subsamples is maintained at two stages. First, when generating subsamples data items are drawn without replacement to ensure no duplicate records are included. Then, each subsample is verified to be unique in the empirical sample space. This method is computationally efficient compared to building M_C. Also, it shows the influence of each data record towards constructing a specific model. This helps us to understand the relationship between data and the resulting machine learning models in terms of model fitting. We term this method as "stratified subsampling".

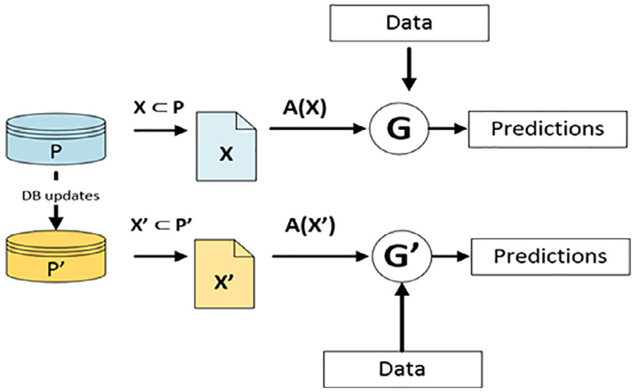


Fig. 1 – Problem notation for repetitively updating machine learning models.

3. Model comparison attacks

In this section, we investigate the risks of not adopting integral privacy for model selection. The privacy risks would be such that, the intruder is able to attain knowledge on the underlying training data of a machine learning model or determine the set of modifications that cause a machine learning model transformation. We have formulated the attack model here along with the intruder's goal and the approach.

3.1. *Framework*

Consider a set of labelled data X, sampled from a database P, which is used to build a machine learning model G using algorithm A. Assume this is set up as a classification problem and A is an algorithm to build decision trees. Eventually, P gets updated into P' as a result of executing erasure requests. This raises the requirement to regenerate the decision trees to match the updated database, P'. For model regeneration, a new data sample X' is obtained from P'. Then a new machine learning model G' is constructed using algorithm A. Note that machine learning models mentioned above are decision trees. Fig. 1 illustrates this scenario. Also, we assume that the intruder has access to the following information about the data and the machine learning models.

- G - Decision tree extracted from the original database ($X \subset P$)

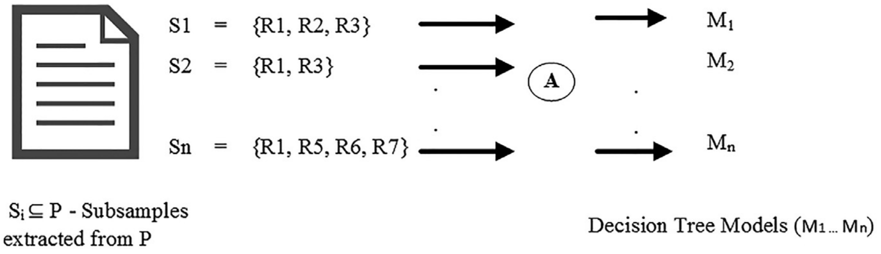

S1 = {R1, R2, R3}
S2 = {R1, R3}
Sn = {R1, R5, R6, R7}
A
M1
M2
Mn
Si ⊆ P - Subsamples extracted from P
Decision Tree Models (M1 ... Mn)

Fig. 2 – An example of generating decision trees models $M_1 \ldots M_n$ from each of the data subsample $S_1, \ldots, S_n \subseteq P$ using machine learning algorithm A.

- G' - Decision tree extracted from the modified database ($X' \subset P'$).
- One time access to database P before any modifications are applied. The point to emphasize here is that the intruder has no knowledge on the exact training data used to build the ML models.
- Knowledge of the machine learning algorithm A.

Also, it is assumed that the input format of data which is used to build the machine learning models (X, X') are the same as in database P.

3.2. *Intruder's goal*

Having access to the above information, intruder's goal is to acquire knowledge on,

- Training dataset X, which is used to build the original machine learning model G,
- Training dataset X' which is used to build the modified machine learning model G',
- Set of modifications μ, carried out on original dataset X, which transformed G to G',

Determining the data records used in the training dataset of a given machine learning model is referred to as "membership inference attacks" in the literature (Shokri et al., 2017). We introduce the term "model comparison attacks" for deriving the set of possible modifications μ, by making use of the respective machine learning models generated before and after the modifications are applied to its training data.

3.3. *Modelling intruder's attack*

Intruder draws blocks of random subsamples $\mathbb{S} = \{S_1, \ldots, S_n\}$ where $S_i \subseteq P$. Each of these subsamples S_i contains a set of records $r_i, \ldots, r_k \in P$. Extracted data subsamples have varying sizes (y) which are randomly decided. Each subsample is a unique "set" of data with respect to the records included (i.e., $S_i \neq S_j \; \forall \; i \neq j$). Then decision trees are trained for each S_i using algorithm A. That is, $M_i = A(S_i)$. This is illustrated in Fig. 2. In this way, the intruder can obtain a subspace M_E of the space of models M_C and an approximation of the probability distribution on the model space for decision trees generated by algorithm A. Representativeness of the model space M_E in relation to M_C, depends on the number of subsamples extracted. For subsampling P, the intruder can use stratified subsampling described in Section 2.2.

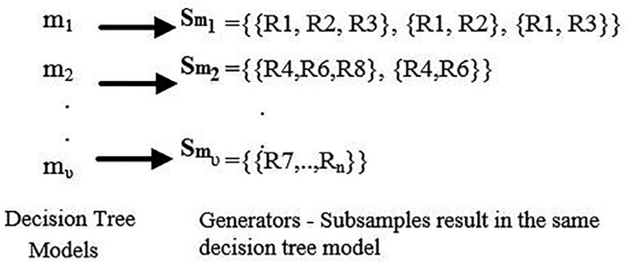

m1
m2
mυ
Sm1 ={{R1, R2, R3}, {R1, R2}, {R1, R3}}
Sm2 ={{R4,R6,R8}, {R4,R6}}
Smυ ={{R7,..,Rn}}
Decision Tree Models
Generators - Subsamples result in the same decision tree model

Fig. 3 – Decision tree model space - An example of decision tree models and their respective generators. $m_1, m_2, \ldots, m_\nu$ refer to the extracted decision rules while generators S_{m_1}, $S_{m_2}, \ldots, S_{m_\nu}$ refer to the different data subsamples that can create the same decision tree.

It is important to note that while all the subsamples (S_i) are different from each other this is not the case for all models (M_i). Therefore, we need to reduce the models into a set of unique models, which corresponds to the empirical model space M_E. To compare the decision trees, we use the decision rules extracted from the decision tree traversing the trees from root to leaf nodes following the same order from left to right. After this comparison we have reduced the collection of models $M_1, \ldots, M_n$ into a set of different models $m_1, \ldots, m_\nu$ with $\nu \leq n$. This set of models is a subset of the model space $M_E = \{m_1, \ldots, m_\nu\} \subseteq M_C$. The association between machine learning models (decision trees) and the set of generators can be formally expressed as $S_{m_i} = \{S_{j \in \mathbb{S}} \mid A(S_j) = m_i\}$. Each of these generators comprise sets of records drawn from P. E.g., $S_1 = \{R_1, R_2, R_3\}$ as mentioned in the Fig. 2. An example of model space is shown in Fig. 3, with a set of unique decision tree models and their respective generators.

The machine learning model space M_E is then used by the attacker to obtain useful information such as the underlying training data or the set of modifications. Below, we discuss three types of attacks an intruder can carry out.

3.3.1. *Membership inference by intersection analysis*

Assume the intruder has built the complete model space M_C, considering all possible sets of generators. Therefore M_C contains G, the original decision tree generated using X, and all the generators of this model. By using this information, the intruder can derive the dominant records that result in a particular decision tree. Dominant records stand for data records, which are a necessity to generate a given decision tree. To derive the set of dominant records we compute the intersection of all the generators, that is, $D(G) = \bigcap_{S_j \in S_{m_i}} S_j$.

In case the intruder has built M_C considering all possible sets of generators from P, the dominant record/s are known with 100% confidence. This implies that $D(G)$ is a subset of the actual training dataset used to generate the given decision tree G. I.e., we have a successful membership attack.

3.3.2. *Membership inference by probabilistic analysis*

Determining the set of dominant records become challenging when the intersection of generators becomes empty or/and the Intruder has built only a subspace of the model space. I.e., $M_E \neq M_C$.

In this case, the intruder can use a probabilistic approach. A probability value can be estimated for each record in P based on the number of times it occurs in the list of generators. E.g., assume there exists three generators, $|S_G| = 3$, that result in a particular decision tree G, and record r_1 appears on two of them. Then we estimate the probability of r_1 generating G by $p(r_1) = 0.66$. Once the probabilities are calculated for all the records available in generators, the items with highest probabilities can be considered as a subset of the training data used to generate a particular decision tree, with reasonable and estimated confidence. Formally, for any record $r_i \in P$ and a given model G, this can be defined as

$$P_G(r_i) = \frac{|\{S \,|\, r_i \in S \wedge S \in S_G\}|}{|S_G|}$$

3.3.3. *Model comparison for detecting the set of modifications*

By using the above approach the intruder can get an idea about the training data set used to generate decision tree models G (decision tree built on X) and G' (decision tree built on modified data X'). The samples that generate G and G' can be used to define the set of modifications. In particular, any transformation from $S_i \in S_G$ to $S_i \in S'_G$ is a possible transformation of the original dataset X to X'. Therefore we can denote that,

$$\mu_E = \{s_i \ominus S_i | s_i \in S_G \textit{ and } s_j \in S_{G'}\}$$

where, $\ominus$ is the operation defined in Section 2.1.

4. Evaluation

This section is focused on testing the privacy risks of "model comparison" by deploying the intruder's approach described in Section 3.3. We gather some empirical results in different settings to show the validity of the concept. Also, we extend the experiments to evaluate the relationship between data and the respective machine learning models using data subsampling.

4.1. *Data*

Four datasets obtained from the UCI machine learning repository (Dua and Karra Taniskidou 2017) are used for the experiments. Stratified sampling is used to partition the training and test datasets after removing any duplicate records. Table 1 shows the datasets and the size of the training and test sets as a percentage of the selected population.

Table 1 – Training and testing datasets as a % of original datafile.

Dataset	Number of records	Train set (%)	Test set (%)
Iris dataset	147	69.4	30.6
Wine dataset	172	81.4	18.6
Glass identification dataset	198	77.3	23.2
Balance scale dataset	179	78.8	21.2

Iris dataset - This is one of the best known datasets used in pattern recognition. Dataset is divided into three different classes and comprises 150 records with 4 attributes.

Wine dataset - The dataset contains a total of 178 records with 13 attributes. It is categorized into 3 different classes. The idea here is to use chemical analysis data of wines to categorize them into 3 classes based on the cultivars.

Glass identification dataset - This dataset contains different glass type classification. 214 records are included in the dataset with 10 different attributes. Glasses are categorized into six different classes based on their oxide content.

Balance scale dataset - Dataset contains 625 records with 4 attributes related to scale balancing. For the testing purpose, we use 180 randomly selected records. There are three classes L, R and B and each includes 49, 49 and 41 records respectively in the training dataset.

4.2. *Experimental setup*

Dataset D is divided into two parts as training set D_t and test set D_h using stratified subsampling to ensure fair representation of all the classes. First, we use the entire training dataset D_t to construct a decision tree (G), which is used as the benchmark model. Then, the stratified subsampling technique is used for extracting k number of unique subsamples from D, with sizes in the interval $[2, |D_t| - 1]$. For each randomly selected subsample $S_1, S_2, \ldots, S_k$, a decision tree is trained and then the decision rules are extracted. Next, the extracted decision rules are compared with each other to figure out the unique decision trees in the empirical distribution and their respective generators (data subsamples that generate the same decision tree). This can also be explained as building the empirical model space M_E for machine learning models (in this case decision trees). As M_E is constructed empirically, it is a subspace of the complete model space M_C.

We consider two scenarios to build the empirical model space M_E. The first approach is when the constructed decision trees are fully over-fitted to its training data and the second approach is when the effect of over-fitting is removed by pruning the decision trees. Each decision tree model obtained from different data subsamples is associated with its frequency of occurrence in M_E. For each dataset, we then build the model space M_E with varying number of subsamples. The size of the subsample space is directly proportional to the size of the original dataset. Therefore, the different subsample sizes we have selected is based on the size of D_t.

For the Iris dataset, we choose 100,000;150,000; and 300,000 subsamples. Wine and Glass Identification datasets are tested for 150,000; and 300,000 number of subsamples. Whereas, the Balance dataset is tested with 100,000; and 150,000 and 300,000 number of subsamples.

The given subsample sizes are selected to approximate the model space. As there is no optimal way of deciding the exact size of the model space with respect to a given dataset, we have used the above values to set the size of the model space. However, when selecting the subsamples sizes the way to proceed is to start with a small number and to gradually increase till the recurrent models start to appear in the approximated model space. The idea is that the model space we generate in each case would be large enough to represent the full model space while containing the recurrent models.

The main criteria for selecting the initial subsample size is the size of the original dataset. A multiple of the original database size can be selected first and then examined for the availability of integrally private models in the generated model space. As mentioned above the subsample size can be improved if the initial model space does not contain the recurrent models with high accuracy. Based on the user's computational resource constraints a maximum size can be set for the number of subsamples. In some cases, the model space built with the maximum number of subsamples might not have the integrally private ML models. The solution to that will be either to modify the original dataset and to re-try or not to select a ML model at the given instance. We have selected a different number of subsamples in the experiments to check how different sizes affect the distribution of empirical model space. Moreover, the sizes were selected to have a large but computationally tractable sample space.

4.3. *Visualizing the empirical model space* (M_E)

Fig. 4 shows the frequency of different machine learning models with and without model over-fitting towards its training data. The same subsamples of data are used in both cases. In each plot, the abscissa represents different models, and the ordinate shows how many subsets of data have resulted in the specific model (i.e., the number of generators). In all the cases illustrated, it is noted that the probability inferred on the model space (M_E) does not follow a uniform distribution, instead it shapes as an exponential distribution. Based on the observations it is prominent that in the model space there exist two extreme cases.

4.3.1. *Frequent machine learning models*

These are the models with a recurrent characteristic. These models appear with a very high frequency, and it implies that they can be built with many different generators. The relationship between a machine learning model and its generators can be explained as 1:N. We can state that the frequent (recurrent) models have high representability over subsamples of training data.

4.3.2. *Infrequent machine learning models*

The other category of machine learning models maintains a 1:1 relationship between the model and the set of generators. This indicates that such a model can only be built with a single, specific generator.

In order to verify the above behaviour of machine learning models, we generated the entire model space M_C for a small toy dataset available on UCI machine learning repository. We used Balloon dataset which only contains 16 instances, thus the number of possible training sets limit to $2^{16} - 1 = 65{,}535$. This experiment confirms the existence of frequent models and infrequent models in the model space.

Also, it is noted that the number of unique machine learning models in the empirical model space (M_E) have reduced as we move from over-fitted models to non over-fitted models (i.e., pruning). This has also caused an increment in the frequency of occurrence for models based on Iris, Balance and Glass-classification datasets. In other words, non over-fitted machine learning models are more generalized towards its training data compared to over-fitted models. Thus, relatively small number of models can be used to represent different subsamples of data. However, in both cases presence of frequent machine learning models were noted.

4.4. *Analysis of frequent (recurrent) and infrequent models*

Based on the model frequency we made a distinction between frequent (recurrent) and infrequent models. As the next step we further analyse the above mentioned model types based on their (a) accuracy (b) sample size and (c) complexity. The intention of the analysis is to show that the probability of choosing an infrequent model is high. For the analysis, we select decision trees with the following characteristics.

- Frequent (recurrent) models - Decision trees with 100 or more generators.
- Infrequent models - Decision trees with a single generator.

Experiments are carried out for all four datasets with differing number of subsamples. And for the comparisons, we have only used non over-fitted trees (i.e., pruned).

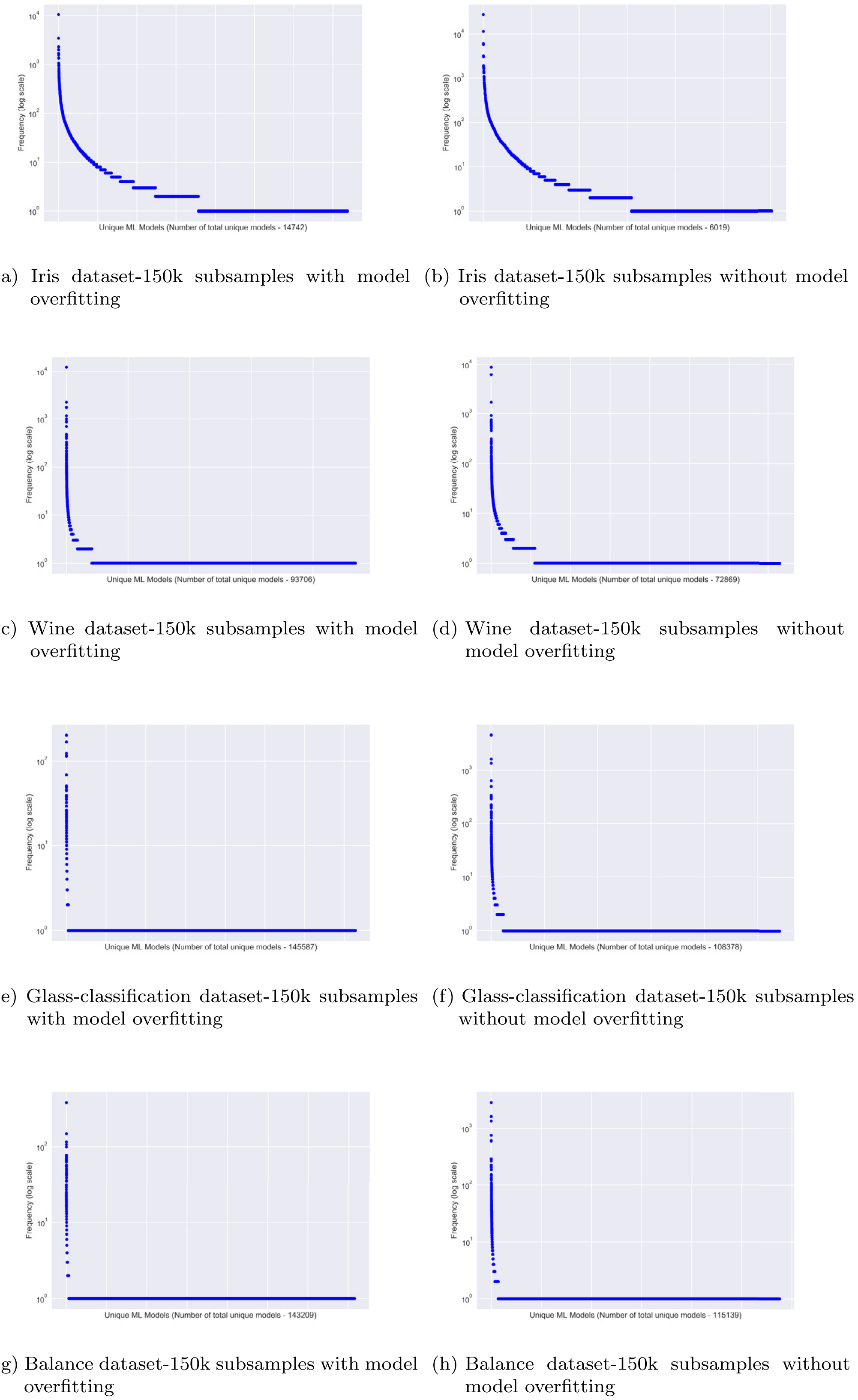
4.4.1. *Accuracy*

A comparison of predictive accuracies for both frequent and infrequent models are shown in Fig. 5. Fig. 5a which refers to the recurrent models shows that Iris and Wine datasets show an accuracy ≈ 0.9, whereas the accuracy levels for Glass and Balance datasets reside in the range of $0.6 - 0.7$ and $0.4 - 0.5$, respectively. Fig. 5a illustrates the infrequent models.

The figures show that the predictive accuracy levels of the two types of machine learning model are similar. If the only focus of model selection is accuracy, it is possible that infrequent models with high accuracy levels could be selected as a result.

4.4.2. *Sample size*

Next, we compare the frequent and infrequent models in terms of the training data subsample sizes (size of the generators); refer Fig. 6. For the frequent models, we obtain the



(a) Iris dataset-150k subsamples with model overfitting

(b) Iris dataset-150k subsamples without model overfitting

(c) Wine dataset-150k subsamples with model overfitting

(d) Wine dataset-150k subsamples without model overfitting

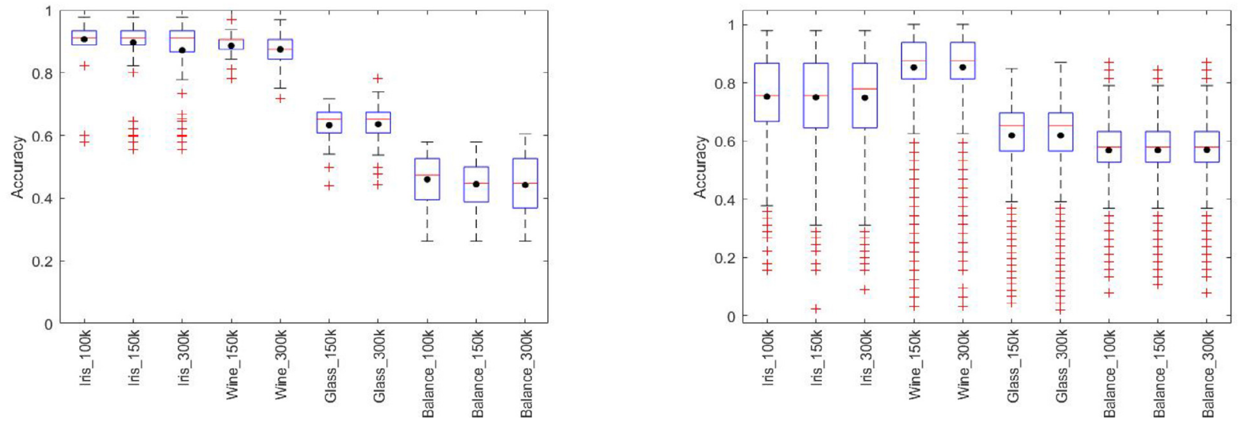
(e) Glass-classification dataset-150k subsamples with model overfitting

(f) Glass-classification dataset-150k subsamples without model overfitting

(g) Balance dataset-150k subsamples with model overfitting

(h) Balance dataset-150k subsamples without model overfitting

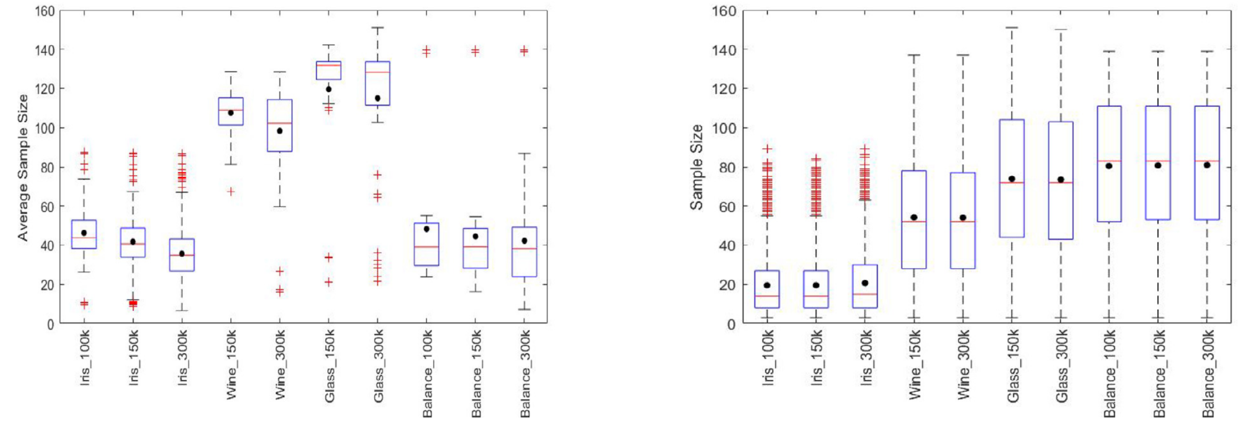
Fig. 4 – Decision tree models and their frequency - Each marker represents a unique decision tree model and its frequency of occurrence in the empirical sample space generated by subsampling D_t. The frequency of a given model is the number of generators (different subsamples) that can generate a given decision tree.



(a) Predictive accuracy for frequent models with more than 100 generators

(b) Predictive accuracy for infrequent models with single generator

Fig. 5 – Predictive accuracy comparison for frequent and infrequent models. The box-plots represent four datasets (Iris, Wine, Glass classification and Balance) with different subsample sizes (e.g., 100 k, 150 k, 300 k). The black marker in each box-plot indicates the mean accuracy value for the specific dataset)



(a) Average size of a data subsample for frequent models with more than 100 generators models

(b) Size of data subsample for infrequent models with single generator

Fig. 6 – Data subsample sizes for frequent and infrequent models - The box-plots represent four datasets (Iris, Wine, Glass classification and Balance) with different training data subsample sizes (e.g., 100 k, 150 k, 300 k). The black marker in each box-plot indicates the mean)

average size of the generators, given $A(S_{i,\ldots,n}) = m_i$, average sample size $= \frac{\sum_{i=1}^{n} |S_{1,\ldots,n}|}{|S_{m_i}|}$, where S_{m_i} is the number of generators. Since infrequent models have only one generator we use the exact size of the generator (subsample size).

The analysis shows that, except for the Balance dataset, the average subsample size of frequent models are larger than infrequent models. For infrequent models, the average subsample sizes are $\sim$ 20 for Iris, 38 for Wine, 48 for Glass and 57 for Balance. This indicates that infrequent models are not always a product of very small subsample sizes. The outliers marked on Fig. 6 b with reference to the Iris dataset, also confirms this. The objective of this experiment is to show that, the infrequent models can also be generated with considerably large subsample sizes. For example in a real world scenario, data deletion can result in an infrequent model, even though the existing training dataset contains a majority of records.

4.4.3. *Complexity*

"Large" decision trees with a high number of nodes could be an indication of over-fitting where the "noise" presented in training data are taken into consideration while building the model. Whereas "small" trees could be very simple models thus they are unable to learn the general concept from given data. However, both of these cases can cause poor predictive

Table 2 – Average complexity of models based on the number of nodes in decision trees.

Dataset	Infrequent models (with single generator)	Frequent models (with more than 100 generators)
Iris 100 k	6	5.09
Iris 150 k	6.1	5.08
Iris 300 k	6.4	4.97
Wine 150 k	7.9	7.7
Wine 300 k	8.04	7.3
Glass 150 k	21.4	9.3
Glass 300 k	21.8	10.93
Balance 100 k	47.2	12.2
Balance 150 k	47.6	9.48
Balance 300 k	48.2	9.49

accuracy. Here, our focus is to understand how model complexity differs between frequent and infrequent models.

The CART algorithm we used for experiments generates binary trees. And the decision trees used here for the complexity observations are pruned.

We derive the complexity of the decision tree models from a very simple measure which is the number of nodes. Table 2 contains a comparison between infrequent and frequent models. Based on the observations it can be seen that the average complexity of infrequent models is higher than the frequent models. Which means that a large tree can be an indication of an infrequent model. For example, following a data deletion if the dataset generates a much larger tree that could be an infrequent model. On the other hand, in terms of addition, this behaviour could also be a result of representing the newly added data.

4.5. *Privacy implications of infrequent and frequent models*

The experimental results explained above show that there is a reasonable chance that a model constructed on a given dataset is an infrequent model. In this case, an intruder who performs a model comparison attack can determine the underlying training data of a given model and also the set of modifications (μ), with high certainty. In other words, infrequent models are not compliant with integral privacy.

In contrast to this, frequent models are privacy friendly. Since multiple numbers of generators can construct the same model, intruder's uncertainty in determining the exact set of training data and modifications (μ) increases. Therefore, in terms of privacy selecting a frequent model is preferable.

4.6. *Accuracy of recurrent models*

Based on the empirical results we recommend to select frequent (recurrent) models over infrequent ones. As the main criterion for model selection is predictive accuracy, in this section we present an accuracy analysis done on the empirical model space (M_E), with the objective of comparing accuracy levels of recurrent models over the others. Mainly we compare recurrent models with the benchmark model which is selected by the machine learning algorithm (decision tree algorithm - CART). The Benchmark model is constructed by using the entire training set D_t. Other models are trained on different data subsamples S_i obtained from empirical training set space which is based on D_t. Fig. 7 depicts the association between frequency of models and their accuracy levels. Here we have illustrated both training and test data accuracies.

By examining the Fig. 7a–d, related to Iris and Wine datasets we can see that the recurrent machine learning models sustain the same or much closer accuracy level as the benchmark model. Recurrent models are clustered around the benchmark model for the above datasets with test and training data accuracy $\approx$ 0.9. The same pattern is observed for both over-fitted and non over-fitted models. The above analysis is carried out for 150,000 subsamples.

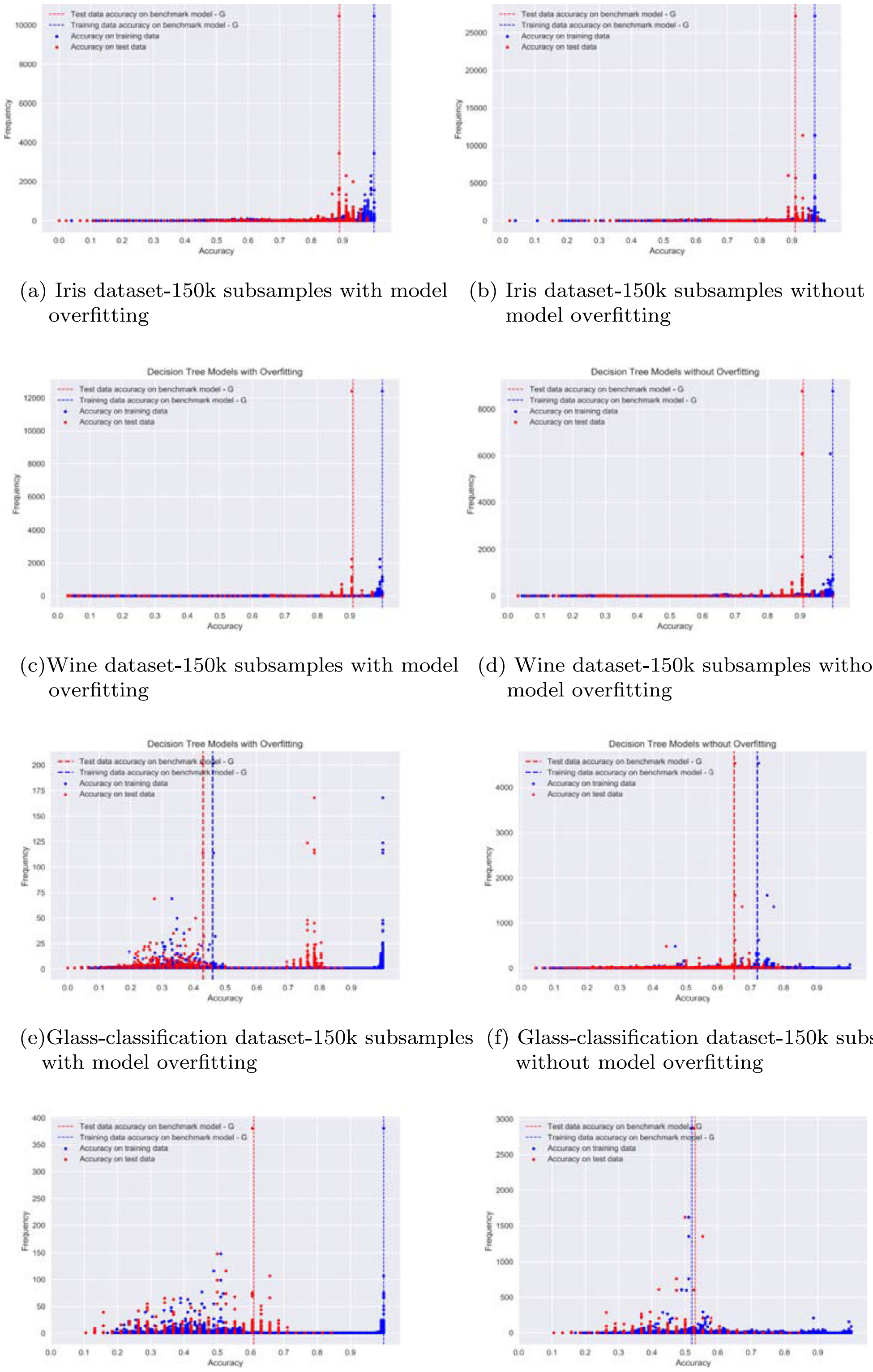
These observations slightly change for Glass-classification and Balance datasets. Fig. 7e shows the over-fitted benchmark model for Glass-classification dataset shows poor accuracy for both training and test data which is in 0.4–0.45 range. This phenomenon can be explained in three ways, (a) the sample complexity of the selected benchmark model is high thus there is a requirement for more training data (b) the selected machine learning algorithm is not capable of modelling the given problem c) poor parameter set-up of the learning algorithm. Fig. 7e shows that there exist many other models (including recurrent models) that report better accuracy compared to the benchmark model. Since our focus is on comparing the predictive accuracy of recurrent models with that of the benchmark model we are not planning to investigate this behaviour further. When the benchmark model is not over-fitted, the accuracy values have increased up to $\approx$ 0.65-0.75 range with respect to both test and training data. This is illustrated in Fig. 7e.

When the decision trees are over-fitted, there exist a significant difference for the Balance dataset between training and test data accuracy levels in relation to the benchmark model. Fig. 7h illustrates the accuracy levels for non over-fitted models. It is noted that the benchmark model accuracies for both train and test data have dropped to $\approx$0.50 compared to the accuracies of the over-fitted version where training data accuracy and test data accuracy are reported to be $\approx$ 1 and $\approx$ 0.60, respectively.

Based on the observations we can see that more often the accuracy of benchmark models and recurrent model are the same or very close to each other. In other words, a recurrent model is a sign that it is more generalized towards its training data thus provides a better accuracy on testing data. Also, as discussed above the recurrent models provide an integral privacy guarantee as the determination of the exact set of generators is difficult when multiple generators exist for a specific model. This shows that considering the rate of recurrence of models is a valid criterion to achieve privacy. Unlike other privacy models, this does not compromise the predictive power of the model to ensure privacy.

4.7. *Intersection and probability analysis for membership inference*

As explained in Section 3, intersection and probability analysis of the set of generators can be used to determine the underlying training data of a given machine learning model,



(a) Iris dataset-150k subsamples with model overfitting

(b) Iris dataset-150k subsamples without model overfitting

(c)Wine dataset-150k subsamples with model overfitting

(d) Wine dataset-150k subsamples without model overfitting

(e)Glass-classification dataset-150k subsamples with model overfitting

(f) Glass-classification dataset-150k subsamples without model overfitting

(g)Balancedataset-150k subsamples with model overfitting

(h) Balance dataset-150k subsamples without model overfitting

Fig. 7 – Training data and test data accuracy of each decision tree model included in the empirical sample space generated by subsampling D_t - Each marker represents a unique decision tree model with respect to its frequency and accuracy. The blue markers show the accuracy of training data and the red markers show the accuracy of test data. Blue and red vertical lines in the plots indicate the training and test data accuracy on the benchmark model, G which is created by using the entire dataset D_t. Other models are trained on data subsamples of D_t.

once the model space is built. Fig. 8 contains box-plot representations of the top 5 recurrent models for each dataset in terms of their generators. Each shows the probability distribution of different data records in a given generator.

As recurrent models noted in a small model space would still be recurrent in a much larger model space, only one subsample size (i.e., 150k) is used for the experiments here.

In the box-plot figures, the points that show a probability of 1 are the elements that can be obtained by applying the intersection operation to all the generators of a given model. In other words, these are dominant records. If we have built the entire model space, we can conclude that these dominant records are a must existence to build a particular model. However, with the empirical model distribution, the records included in the intersection can be considered as dominant records with a high likelihood but not with 100% assurance. The plots also show that higher the frequency of the models, the majority of the records have a high probability. In order to deduce knowledge on training data by analysing the probability distribution of the records, we can introduce a probability range. If a given data record has a probability $\geq$ 0.8 ideally that element can be considered as part of the underlying training set with fair confidence. When considering the Balance dataset, the probabilities of the items are much lower than the others. Since the Balance data set has a high number of records the sample space of training data is large. And therefore, the number of subsamples we have selected could be inadequate for a proper representation of M_C. This could be the reason for the above observation.

5. Integrally private model selection

The empirical results show that the same decision tree (model) can be created by different subsamples of data (generators). Different data subsamples that maximize the information gain (IG) for the same splitting conditions result in the same decision tree. Due to the higher representability of recurrent models, they are less sensitive towards perturbation done to input training data. If we have already deployed a recurrent model, there would be fewer chances that the model needs to be replaced into a completely new one in response to training data modifications (this is in contrast to deploying an infrequent model). This very idea can also be explained in relation to “algorithmic stability”.

Stability of learning algorithms is defined in Turney (1995), as the degree to which it generates repeatable results, given different batches of data generated from the same process. Stability of learning algorithms is more focused on maintaining the robustness of the predictions. Whereas, we are interested in the robustness of the model itself. In other words the extent to which, a model can accommodate the changes introduced to its training data without causing a model transformation or compromising accuracy.

In Sections 4.4 and 4.6 we have shown experimental results to confirm the statements below,

- Infrequent models pose a privacy risk and yet there is a possibility that this kind of models can get picked in model selection.
- Frequent (recurrent) models often have a high accuracy which is almost the same or sometimes even better than the benchmark models.

This implies that opting for a recurrent model would provide a fair trade-off between privacy and predictive accuracy. As explained in “integral privacy”, recurrent models can be used to ensure a privacy gain. The privacy model ensures that, the uncertainty of the intruder is high on the membership of training data and the set of modifications (μ) if a particular machine learning model has many generators. We have described “model comparison attacks” based on this concept and validated it using empirical results.

To recap what we did, consider a machine learning model m_x that has many generators $(S_1, S_2, \ldots, S_k)$. Due to some modification/s (μ) that took place on the training data of m_x, the model has transformed into m_y. m_y has t number of multiple generators $(S'_1, S'_2, \ldots, S'_t)$. In this scenario, even if the intruder has the entire model space (M_C) generated, it would be with a probability of 1/t that he can infer the generator that has been used to build m_y. So, the intruder has uncertainty in determining exactly what generator resulted in the particular model at a given instance. Similarly, when trying to determine the set of modifications (μ) carried out on the training data, it becomes increasingly difficult to narrow it down as both machine learning models have multiple generators with equal chances of being the generator of a particular model. Instead, if m_y is an infrequent model with a single generator, an intruder can exactly determine what training data has used to build the model and also what modifications (μ) on training has caused the model transformation.

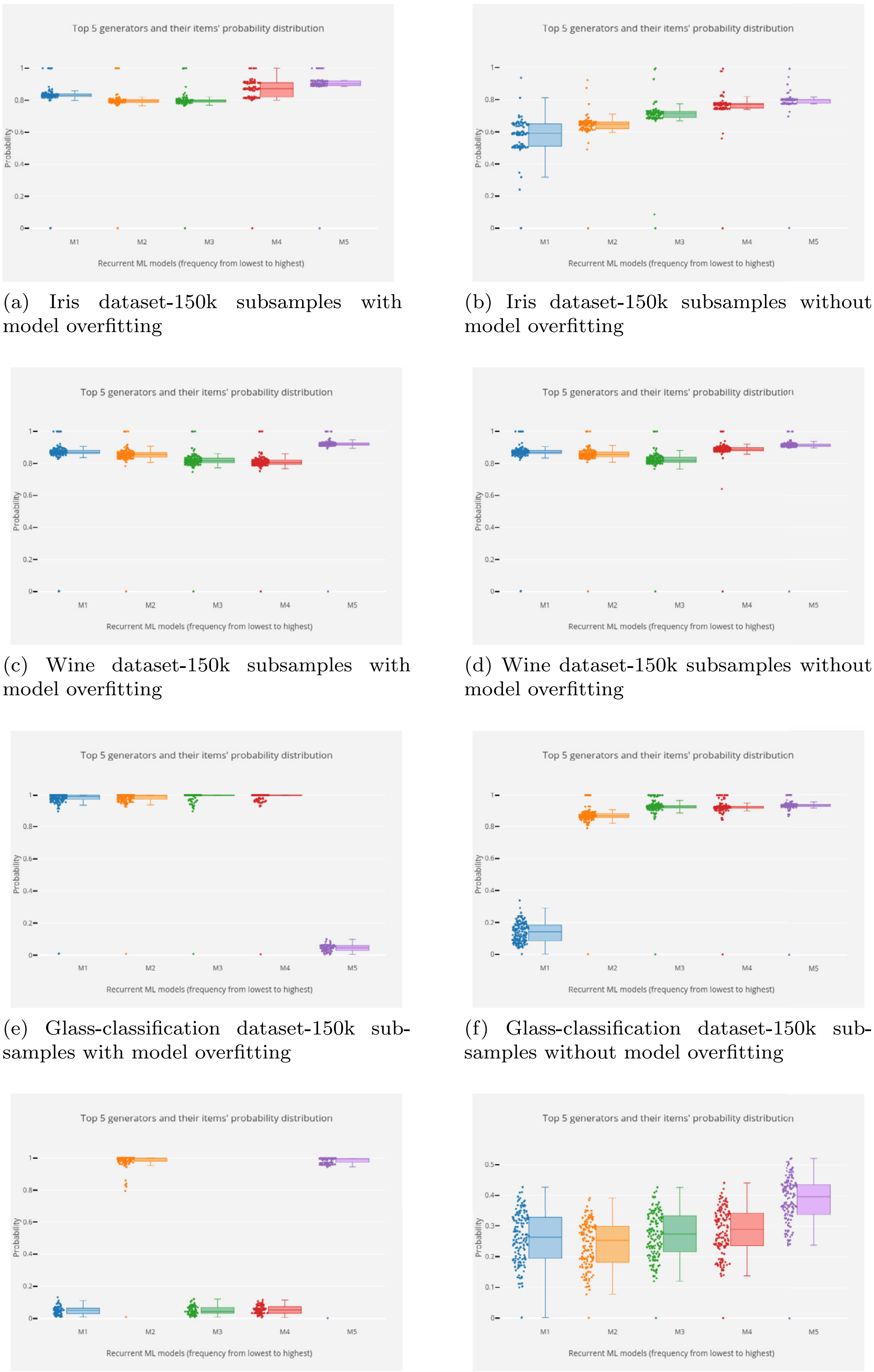
Therefore, it is obvious how selecting a machine learning model with 1 or a few numbers of generators can pose a privacy risk. The empirical results show three classes of machine learning models based on their recurrence rate.

- A single machine learning model with very high recurrence rate that maintains a 1:N relationship with the model and the set of generators.
- A set of machine learning models with average recurrence rate that maintains a 1:n relationship with the model and the set of generators (where $n \ll N$).
- A set of machine learning models with very low recurrence rate. These models are isolated in the model space and maintain a 1:1 relationship (or very low frequency) with the model and the set of generators.

Based on the above mentioned facts we discuss the need to select recurrent models generated with a diversity of generators and that have good accuracy. We close this section with a summary of procedure for integrally private model selection.

5.1. *Recurrence*

First criterion for integrally private model selection is the frequency of a given machine learning model. This can be extracted from the derived empirical distribution. In other words, a machine learning model has to be recurrent. The next step is to define the term “recurrent”. In integral privacy, this is addressed in the definition of k-anonymous integral privacy



(a) Iris dataset-150k subsamples with model overfitting

(b) Iris dataset-150k subsamples without model overfitting

(c) Wine dataset-150k subsamples with model overfitting

(d) Wine dataset-150k subsamples without model overfitting

(e) Glass-classification dataset-150k subsamples with model overfitting

(f) Glass-classification dataset-150k subsamples without model overfitting

(g) Balance dataset-150k subsamples with model overfitting

(h) Balance dataset-150k subsamples without model overfitting

Fig. 8 – Probability of different records included in the generators resulting the top 5 recurrent models - M1,M2,M3,M4 and M5 in the *x*-axis refer to a recurrent machine learning model ordered from lowest to the highest in terms of frequency of the given models.

where the set of modifications M, needs to have at least k-minimal elements. With respect to recurrent models, we introduce a parameter k, which will be the number of generators for a specific machine learning model. k Is adjustable based on the privacy preference of the user. This results in a k-integrally private model.

To find these k-integrally private model we have to sweep the machine learning model space with a significant amount of subsamples. The computation cost of this process is extremely high if the size of the database is large. In practice, we can only obtain an empirical distribution for the model space M_E, as the subsample space grows exponentially with the size of the database. This implies that our most recurrent model can be superseded by another one. In this case, however frequencies of models can only increase when we add additional subsamples into our set of samples. For an actual figure on the sample size, we recommend starting with a sample size which is $\geq$ 100 times the size of the database D_t. If this does not provide enough recurrent models, then users will have to gradually increase the number of subsamples constrained by a given maximum value. The above given value is just to set a direction in selecting a sample size. Based on the user's computational resources this value can be changed.

5.2. *Accuracy*

The selected machine learning model (m_i), with at least k generators, should provide an accuracy level closer or higher than the benchmark model. Our empirical results show that the benchmark model is not always the "best" with respect to the number of generators and the accuracy of the model. In general, we have these four categories.

- Models with high accuracy and a high number of generators.
- Models with high accuracy but with a low number of generators.
- Models with low accuracy but with a high number of generators.
- Models with low accuracy and a low number of generators.

The first category is highly acceptable as they provide the best accuracy and privacy guarantee. The latter is unacceptable as it is poor in both accuracy and privacy. Apart from that, the other two categories are acceptable, because they could be in line with user's privacy requirement. For example, someone with high privacy preference may be ready to compromise accuracy. Thus, ending up selecting a model which has lower accuracy than others but with a higher number of generators. Similarly, someone might select the second category of models, where the number of generators are low but still larger than a given k, if the model accuracy is high.

5.3. *Diversity*

When different models satisfy the requirement of having at least k generators, a distinction can be made based on the distribution of elements in the set of generators. In other words, if we consider all the elements included in the set of generators (derived for a specific model) and the probability of each item in the generator is low, that implies diversity in the set of generators. Higher the diversity among the generators the more stable the model becomes with respect to record removal. This has been discussed in Section 4.7 in relation to Fig. 8. A diverse model would comprise elements with low probabilities and a minimal set of items or no items at all satisfying the intersection operation (i.e., probability = 1).

5.4. *Model selection procedure*

Algorithm 1 summarizes the process of integrally private model selection including, (a) building the empirical model space (M_E), (b) deriving the distribution of models (*DistributionOfModels*) and (c) model selection based on accuracy, recurrence and diversity (*CandidateModelsList*). The users can define threshold values for the accuracy, recurrence and diversity. Based on those parameters there could be more than one eligible ML models to select from. User's preferences on accuracy, recurrence and diversity can be used to determine the final model. Also, with respect to a given set of parameters and a dataset there might not be any integrally private models available in the space of models, thus making the *CandidateModelsList* empty. In this case, either a parameter modification

Algorithm 1: Integrally private model selection procedure. For a given dataset D_t the algorithm either returns a list of integrally private ML models or an empty model list; when there is no integrally private models for the given dataset that matches with the user defined set of parameters.

```
Data: $D_t$: Data set;
      A: Decision tree algorithm;
      n: number of samples;
      $acc_m$ : Accuracy threshold for ML models;
      $k$: frequency threshold for ML models;
      $prob_r$: Maximum item probability in generators;
Result: An integrally private ML model/s
$M_E \leftarrow list()$;$CandidateModelsList \leftarrow list()$;$DistributionOfModels \leftarrow list()$;
for $i = 1$ to $n$ do
    $S_i$ := sample($D_t$) ;
    $m_i$ := A($S_i$)                        ▷ Generate a ML model for given $S_i$
    $M_E$ := add˜($m_i$, $S_i$, accuracy)     ▷ Update the empirical model space
end
for each unique $m_i \in M_E$ do
    $Frequency_i$ := $frequency(m_i)$;
                  ▷ Derive the distribution of models from $M_E$
    $DistributionOfModels$ := add ($M_{E_i}$, $Frequency_i$);
end
for each $m_i \in DistributionOfModels$ do
    if $accuracy(m_i) \geq acc_m \wedge frequency(m_i) \geq k \wedge$ $element\_probability(S_i) \leq prob_r$ then
        $CandidateModelsList$ := add˜($m_i$) ;
    end
end
return $CandidateModelsList$ ;
```

or dataset modification is required to select integrally private ML models. Otherwise, a ML model complies with integral privacy cannot be selected for the specific instance.

In summary, integrally privacy ensures a good trade-off between prediction accuracy and privacy. Based on the above discussion we define a machine learning model to be "Integrally Private" if it is a recurrent model with at least k number of generators, assumes an accuracy level close to the benchmark model and the diversity of items in the set of generators is high.

6. Related work

The goal of privacy preserving data analysis is to ensure that the privacy of individual records in a dataset is protected from adversarial attacks. However, this matter becomes more complex when machine learning comes to play either as the tool for data analysis or as the tool for launching privacy attacks. Machine learning models are vulnerable to privacy attacks in terms of leaking the sensitive information about their training data. Membership inference is one of the most critical privacy attacks against a given machine learning model (Shokri et al., 2017). The attack model explains; given a data record and black-box access to a model, the goal of membership inference is to determine if the record is a part of the model's training dataset or not. Ateniese et al. (2015) present another genre of attacks where meaningful information can be inferred about the training data by constructing a meta-classifier to hack the target models. Statistical and demographic information about training data that the machine learning models have preserved in training can be obtained through this approach.

Existing research on privacy preserving machine learning has its main focus on collaborative learning. That is when multiple parties come together to utilize their information in order to train machine learning models, without sharing sensitive information with each other (Lindell and Pinkas, 2009). The objective here is to minimize information leakage during the training phase. However, this does not provide any protection against the above mentioned attacks targeting the machine learning models. In literature, different data perturbation techniques (i.e., data masking, k-anonymity) and privacy models (i.e., differential privacy) are used to address this issue. The biggest challenge of adopting the above methods is managing the trade-off between privacy and model utility.

Differential privacy (Dwork, 2006) has contributed towards privacy preserving machine learning through different approaches such as the implementation of differentially private machine learning algorithms, output perturbation and objective perturbation to name a few (Ji et al., 2014). The intuition of differential privacy is that the probability of a given outcome is essentially unchanged as a result of modifying a single data instance. The definition itself implies that the distribution of outputs does not depend too much on any data instance. Therefore, theoretically differential privacy provides a strong stability guarantee (Ateniese et al., 2015). However, differential privacy has its focus on neighbouring datasets that differ from each other at most 1 record. But with a dynamic database, where changes take place regularly, maintaining this level of stability can be very costly for model utility. The approach we consider in this paper has its focus on selecting recurrent models, but this does not require the generators of these models to be neighbours in the sense of differential privacy. Integral privacy (Torra and Navarro-Arribas, 2016) defines a degree of privacy based on the number of unique generators for a given model. This can also be explained as the stability of the models.

Stability of learning algorithms or the concept of algorithmic stability has been extensively studied in the machine learning literature, in relation to learnability, generalization (Bousquet and Elisseeff, 2002; Breiman, 1996; Breiman et al., 1996; Elisseeff et al., 2005; Kearns and Ron, 1999; Mukherjee et al., 2003; Shalev-Shwartz et al., 2010; Turney, 1995; Xu and Mannor, 2012) and privacy (Dwork, 2011; Dwork et al., 2014; Wang et al., 2016). A learning algorithm is defined to be stable if small perturbations done on training data do not imply significant changes in the output of the algorithm. To determine the stability of learning algorithms the literature has mainly focused on deriving theoretical bounds for the generalization error based on techniques such as cross-validation (Kale et al., 2011) and leave-one-out error (Elisseeff et al., 2003; Evgeniou et al., 2004; Kearns and Ron, 1999; Mukherjee et al., 2003). Model stability can be explained as a result of strong algorithmic stability. Highly generalized models produce a low generalization error while generalizability can also lead towards multiple generators per model. However, to achieve integral privacy, we are focused on recurrent models so that the selected model remains the same with a high probability despite the modifications done to training data. In the standard definition of algorithmic stability, the "perturbation" is expected to be "small" on both training data and the outcome. But in our approach, we are not concerned about the extent of the perturbation done on training data; large or small. For model stability, the requirement is such that, models remain identical irrespective of the degree of change carried out on training data.

The notion of stability is studied with reference to differential privacy (Dwork, 2011). This has introduced two stability related concepts namely, subsampling stability and perturbation stability.

- Subsampling stability - f is said to be q-subsampling stable on dataset x, if $f(x) = f(x')$ with a probability at least 3/4 when x' is randomly subsampled from x, which includes each entry independently with a probability q.
- Perturbation stability - f is said to be stable on x, if f takes the value $f(x)$ on all neighbours of x (and unstable otherwise).

In this paper, we try to explain recurrent models by working along the lines of "subsampling stability".

We recommend users to select recurrent models, which are naturally available in the model space (with high representability). This ensures a high model utility while providing a privacy assurance against integral privacy attacks. And also in our work, we have shown how these recurrent models can increase intruder's uncertainty with regarding membership inference attacks.

7. Conclusion

The work carried out in this paper is based on the privacy model "Integral Privacy". In our work, we have shown the significance of integral privacy by referring to a newly introduced attack model which is based on machine learning model comparison. Using experimental results, we have shown how the distribution of models can be used as a mitigation strategy against model comparison. Based on our observations of the empirical model space, two categories of machine learning models are identified as frequent (recurrent) and infrequent models. We have explained in detail how frequent models can be used to achieve integral privacy with minimum compromise of model utility. Finally, we have provided recommendations for integrally private model selection.

The method we have used in this paper for constructing the empirical model space can be inefficient for large datasets. Therefore, going forward we can experiment with different resampling techniques to find an efficient method. We have only considered decision trees in this work for building the model space. This can also be tested with other machine learning algorithms. Additionally, we can analyse the similarity of the datasets that results in the same models in order to understand the relationship between data and their models further. As proposed in the original paper (Torra and Navarro-Arribas, 2016), we can carry out the same experiments with different data protection methods (micro-aggregation, differential privacy etc.) to understand its affect towards integral privacy.

Acknowledgments

This work is supported by Vetenskapsrådet project: ”Disclosure risk and transparency in big data privacy” (VR 2016-03346, 2017-2020).

REFERENCES

Ateniese G, Mancini LV, Spognardi A, Villani A, Vitali D, Felici G. Hacking smart machines with smarter ones: how to extract meaningful data from machine learning classifiers. Int J Secur Netw 2015;10(3):137–50.

Bousquet O, Elisseeff A. Stability and generalization. J Mach Learn Res 2002;2:499–526. Mar

Breiman L. In: Behavioral sciences mathematics and statistics. Classification and regression trees. 1st. NewYork: Chapman and Hall; 1984.

Breiman L. Bagging predictors. Mach Learn 1996;24(2):123–40.

Breiman L, et al. Heuristics of instability and stabilization in model selection. Ann Stat 1996;24(6):2350–83.

Dwork C. Differential privacy. In: Bugliesi M, Preneel B, Sassone V, Wegener I, editors. In: Automata, languages and programming. Berlin, Heidelberg: Springer; 2006. p. 1–12.

Dwork C. The promise of differential privacy: a tutorial on algorithmic techniques. In: Proceedings of the annual IEEE symposium on foundations of computer science, FOCS; 2011. p. 1–2. doi:10.1109/FOCS.2011.88.

Dwork C, Feldman V, Hardt M, Pitassi T, Reingold O, Roth A. Preserving Statistical Validity in Adaptive Data Analysis. CoRR 2014;abs/1411.2664 http://arxiv.org/abs/1411.2664.

Elisseeff A, Evgeniou T, Pontil M. Stability of randomized learning algorithms. J Mach Learn Res 2005;6(Jan):55–79.

Elisseeff A, Pontil M, et al. Leave-one-out error and stability of learning algorithms with applications. NATO Sci Ser Sub Ser III Comput Syst Sci 2003;190:111–30.

Evgeniou T, Pontil M, Elisseeff A. Leave one out error, stability, and generalization of voting combinations of classifiers. Mach Learn 2004;55(1):71–97. doi:10.1023/B:MACH.0000019805.88351.60.

Ji Z., Lipton Z.C., Elkan C. Differential privacy and machine learning: a survey and review. arXiv:14127584 2014.

Kale S, Kumar R, Vassilvitskii S. Cross-Validation and Mean-Square Stability. In: Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 7–9, 2011. Proceedings; 2011. p. 487–95 http://conference.itcs.tsinghua.edu.cn/ICS2011/content/papers/31.html.

Kearns M, Ron D. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. Neural Comput 1999;11(6):1427–53.

Dua D, Karra Taniskidou E. UCI Machine Learning Repository; 2017. http://archive.ics.uci.edu/ml.

Lindell Y, Pinkas B. Secure multiparty computation for privacy-preserving data mining. J Priv Confid 2009;1(1):5.

Mukherjee S, Niyogi P, Poggio T, Rifkin R. Statistical learning: well-posedness is necessary and sufficient for consistency of empirical risk minimization. Technical Report 223; 2003.

Politis D.N., Romano J.P., Wolf M. Subsampling in the I.I.D. case; New York, NY: Springer, New York. p. 39–64. doi:10.1007/978-1-4612-1554-7_2.

Politis DN, Romano JP, Wolf M. On the asymptotic theory of subsampling. Stat Sin 2001:1105–24.

Shalev-Shwartz S, Shamir O, Srebro N, Sridharan K. Learnability, stability and uniform convergence. J Mach Learn Res 2010;11(Oct):2635–70.

Shokri R, Stronati M, Song C, Shmatikov V. Membership inference attacks against machine learning models. In: Proceedings of the IEEE symposium on security and privacy (SP), 2017. IEEE; 2017. p. 3–18.

Torra V, Navarro-Arribas G. Integral privacy. In: Foresti S, Persiano G, editors. In: Cryptology and network security. Cham: Springer International Publishing; 2016. p. 661–9.

Turney P. Technical note: bias and the quantification of stability. Mach Learn 1995;20(1):23–33. doi:10.1023/A:1022682001417.

Wang Y-X, Lei J, Fienberg SE. Learning with differential privacy: stability, learnability and the sufficiency and necessity of ERM principle. J Mach Learn Res 2016;17 183:1–183:40.

Xu H, Mannor S. Robustness and generalization. Mach Learn 2012;86(3):391–423. doi:10.1007/s10994-011-5268-1.

Navoda Senavirathne - Received her B.Sc. from Sri Lanka Institute of Information Technology in 2011 specializing in Information Technology. And M.Sc. in Information Security in 2016 from University of Colombo, Sri Lanka. She is currently a Ph.D. candidate at School of Informatics at University of Skövde, Sweden. Her research interests include data privacy, disclosure risk control and privacy-preserving machine learning.

Vicenç Torra - (Barcelona, Catalonia, 1968; B.Sc. 1991, M.Sc. 1992, Ph.D. 1994 in Computer Science) is a Professor at the University of Skövde, Sweden. From 1999 to 2014 he was an Associate Research Professor at the Artificial Intelligence Research Institute (IIIA- CSIC) in Catalonia. He has been visiting researcher at the University of Tsukuba (Japan). His fields of interests are approximate reasoning, information fusion, and data privacy. His research has been published in specialized journals and major conferences. He has written a few books including ”Modeling Decisions: Information Fusion and Aggregation Operators” (Springer, 2007) with Yasuo Narukawa, and ”Data Privacy” (Springer, 2017). He is the founder and editor of the ”Transactions on Data Privacy”.