Membership Inference Attack in Random Forests

Fatemeh Akbarian and Amir Aminifar *

Department of Electrical and Information Technology, Lund University, Sweden

Abstract. Machine Learning (ML) offers many opportunities, but its reliance on personal data raises privacy concerns. One such example is the Membership Inference Attack (MIA), which aims to determine whether a specific data point was part of a model's training dataset. In this paper, we investigate this attack on Random Forests (RFs) and propose a method to quantify their vulnerability to MIA. We also demonstrate that in collaborative setups like federated learning, a client with access to the model and partial training dataset can establish MIA against other clients' training data. The effectiveness of our method is validated through experiments.

1 Introduction

The remarkable capabilities of Machine Learning (ML) models come with significant privacy concerns, as they are often trained on extensive personal data [1,2]. One particularly concerning type of privacy is the Membership Inference Attack (MIA) [3], which aims to determine whether a specific data sample was part of a model's training dataset. Such attacks pose a serious threat to individual privacy, potentially exposing sensitive information about users whose data was used to train the model. As ML continues to permeate various aspects of our lives, addressing these privacy challenges becomes paramount to maintaining public trust and protecting individual rights in the age of data-driven decision-making.

These privacy concerns and MIA have garnered significant attention. Shadow training was introduced in [3], a technique that creates a proxy for the target model's behavior. This approach enables adversaries to train their attack model without directly accessing the target Model's training data. However, this method requires knowledge of the training data distribution and the use of multiple shadow models. In [4], these assumptions of the shadow training technique are relaxed and a MIA method using an unsupervised binary classification is proposed. In [5] and [6], the relationship between overfitting and privacy leakage is discussed. In [7], a new method for performing MIAs on ML models by exploiting differences in prediction sensitivities between training and non-training data is proposed. More recently, in [8], an optimization-based reconstruction attack is introduced that can nearly completely reconstruct the training dataset of a Random Forest (RF) model, demonstrating a significant privacy vulnerability in widely used ensemble methods. However, exact reconstruction using this approach is generally possible only in datasets with binary features.

*This research has been partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP), Swedish Research Council (VR), and European Union (EU).

This paper demonstrates that an adversary with access to an RF model and partial training data can conduct MIA against the remaining training data. This vulnerability is particularly relevant in collaborative machine learning scenarios, such as federated learning, where multiple participants build a joint model using their individual datasets. We show that a participant can potentially execute MIA against other participants' training datasets using its own data and the joint model. Therefore, our proposed method determines whether a single sample belongs to an RF model's training set, requiring only access to the model and a portion of the training data. At the same time, from a model provider's perspective, our method can evaluate and quantify the model's vulnerability to MIA against the protected portion of the training data. We detail our approach in the next section.

2 Method

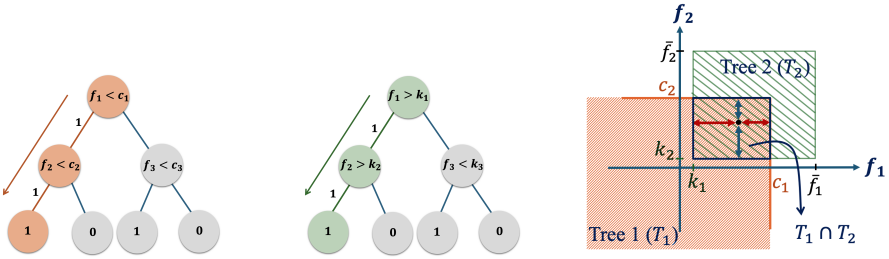
In this paper, we investigate the MIA attack on RF models. RF is an ensemble learning method that constructs multiple decision trees for tasks like classification and regression, with the final output determined by majority voting among trees. To address the MIA problem, we investigate the margins of samples from the decision boundaries in each tree.

Our approach systematically traces each sample through every tree in the RF model. We begin by selecting numerical features and initializing their ranges using global minimum and maximum values from the dataset. We update the range for each feature by intersecting the conditions encountered in the nodes along the sample's path. Subsequently, we compute the intersection of these defined ranges across all trees, establishing a final range for each feature per sample. Figure 1 illustrates this process with a simplified example. The orange part in Figure 1a represent the sample's path through Tree 1 ($T1$), with the resulting feature ranges depicted by the orange area in Figure 1c. Considering an RF model with two trees, the green area in Figure 1c represents the feature ranges defined by the second tree, i.e., Tree 2 ($T2$), for the same sample in Figure 1b. The intersection of these two trees' ranges, shown as the $T_1 \cap T_2$ area in Figure 1c, represents the final feature range for the sample. $\overline{f}_1$ and $\overline{f}_2$ in Figure 1c are the global maximum values for f_1 and f_2, respectively.

The final intersected range effectively delineates the decision boundaries for each sample's classification within the RF model. For each numerical feature, we compute two critical distances: the absolute difference between the feature's actual value and both the lower and upper bounds of this final range. Formally, for each numerical feature f of sample s, we calculate:

$$\begin{aligned} \underline{d}_s^f &= \min\left(\left|s^f - r_{\min}(f,s)\right|, \left|s^f - r_{\max}(f,s)\right|\right), \\ \overline{d}_s^f &= \max\left(\left|s^f - r_{\min}(f,s)\right|, \left|s^f - r_{\max}(f,s)\right|\right), \end{aligned} \tag{1}$$

where s^f is the value of feature f for sample s, and $r_{min}(f,s)$ and $r_{max}(f,s)$ are respectively the lower and upper bounds of the final range obtained for feature f for sample s from all trees. In Figure 1c, red and blue arrows within the $T_1 \cap T_2$



(a) Path of the sample through Tree 1.

(b) Path of the sample through Tree 2.

(c) Feature range intersection in the RF trees.

Fig. 1: Illustration of our proposed method using a simplified example

area indicate $\underline{d}_s^f$ and $\overline{d}_s^f$ for features f_1 and f_2 in the example, respectively. Consequently, for a dataset with m numerical features, we generate a new feature space of dimensionality $2 \cdot m$. We generate this new dataset by combining the test set with an equally sized random subset of the training set. We assign a binary target variable: 1 for samples originating from the training set and 0 for those from the test set. Subsequently, we employ this augmented dataset to train a new ML model for membership inference, tasked with classifying samples as either part of training or not. To clearly distinguish between the two models discussed in this paper, we refer to the main RF model being analyzed as "Target Model" and the ML model inferring data membership as "Inference Model".

RFs adopt the bootstrapping technique to avoid overfitting, which may need to be considered. Bootstrapping involves randomly sampling the original dataset with replacement to create unique subsets for each decision tree, enhancing diversity and reducing overfitting. The size of these bootstrap samples can be controlled. We explore the relationship between the Target Model's bootstrap sample size and Inference Model performance in Section 3.

3 Evaluation

3.1 Experiment Setup

To evaluate our proposed method, we employ the RF model developed by [9] for predicting colorectal cancer survival as the Target Model. This model, trained on a comprehensive dataset of 31,916 patient records from Hospital Based Cancer Registries of Sao Paulo, achieves approximately 77% accuracy in predicting cancer-specific survival using 25 patient features. Among the 25 features, our analysis focuses exclusively on numerical features: age, CONSDIAG (days between consultation and diagnosis), TRATCONS (days between consultation and treatment), and DIAGTRAT (days between diagnosis and treatment), excluding binary and categorical features. Twenty-five percent of the dataset is designated as the test set, while the remaining portion is used for training the model.

3.2 Experiment Results

After training the main RF model, we will develop an Inference Model and assess its accuracy in detecting training data. First, to ensure fair evaluation, we randomly select an equal-sized subset from the larger training data to match the test set size. By combining this subset with the test set, we create a balanced dataset and apply our proposed algorithm to generate a new dataset with $2 \cdot m$ features. Since the main data used to train the Target Model has 4 numerical features, the new dataset will have 8 features and a binary label (1 for training data, 0 for testing data). We then train the Inference Model on 70% of this new dataset to classify training and testing instances, as outlined in Section 2 and use the remaining 30% to test the inference model. We utilize an RF model with 100 trees as the Inference Model for this classification. We assess this model's accuracy in distinguishing training from testing data, comparing its performance with and without bootstrapping in the Target Model's training process.

Figure 2 illustrates the Inference Model's accuracy and standard deviation under various bootstrapping conditions. For robust evaluation, we repeated each scenario 10 times, generating a new dataset and training a new Inference Model each time. We examined six scenarios for the Target Model: bootstrapping with max samples of 20%, 40%, 60%, 80%, and 100%, and no bootstrapping (None) using the full dataset for each decision tree. Results show that with bootstrapping, accuracy increases as the percentage of data used per tree grows. The highest accuracy (mean value around 75.2%) is achieved without bootstrapping, where each tree accesses the entire dataset. This is because, by bootstrapping and using only a subset of data in training the Target Model, a sample may not be used to train a certain tree. Conversely, using more samples per tree is likely to produce more accurate feature ranges. Hence, for our task of distinguishing training from test data, larger subsets or using the entire dataset (no bootstrapping) in the Target Model yield higher accuracy in the Inference Model.

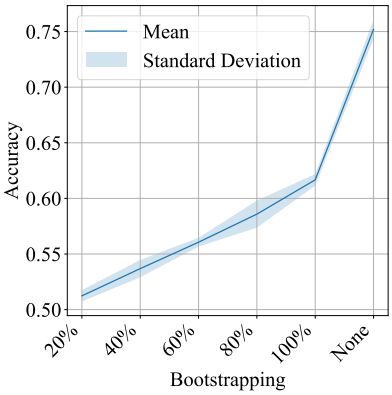


Fig. 2: Inference Model's Accuracy

Note that 100% bootstrapping is different from the scenario without bootstrapping (None). In 100% bootstrapping, each tree is trained on a dataset of the same size as the original but created by random sampling with replacement from the original dataset, potentially including duplicate samples while omitting others. In contrast, without bootstrapping, the entire original dataset is used for each tree. This comprehensive use of data in the no-bootstrapping case contributes to the higher accuracy of the Inference Model.

Figure 3 presents the confusion matrix of the Inference Model for the scenario without bootstrapping, showing 75.5% accuracy with high true positive (86.2%) and true negative rates (64.9%). Notably, the true positive rate exceeds the true

negative rate, indicating that the Inference Model classifies training data better than test data. This bias aligns with our objective of identifying data used in training, as it minimizes the risk of overlooking training data.

In the next phase of the experiment, we evaluate how training and test datasets can be distinguished using the new dataset generated by our proposed method. Here, we consider the Target Model without bootstrapping. To ensure a fair evaluation, we randomly select an equal-sized subset from the larger training data to serve as our training set, matching the test set size (7979 samples). Additionally, since our goal is to determine whether the data has been used to train the model, we extract another equally sized subset as a reference set from the remaining training data. In the collaborative/federated learning setting, for instance, this could be the data each client has locally access to. For each sample in these three datasets, we measure $\underline{d}_s^f$ for all the selected numerical features, meaning that for each sample, we will have four different values. Then, for each sample, we calculate the geometric mean [10] of these four values and call it D_s. We generate B_{ref}, B_{train}, and B_{test} sets, which include D_s for all samples in the reference, training, and test datasets, respectively. We calculate the geometric mean over all D_s in each of these sets as $\overline{B}_{ref}$, $\overline{B}_{train}$, and $\overline{B}_{test}$. Then, we normalize $\overline{B}_{train}$ and $\overline{B}_{test}$ with respect to $\overline{B}_{ref}$: $\hat{B}_{train} = \frac{\overline{B}_{train}}{\overline{B}_{ref}}$ and $\hat{B}_{test} = \frac{\overline{B}_{test}}{\overline{B}_{ref}}$. We consider two different scenarios: one with a constant tree number 100 and varying max depths: $[5, 10, 15, 20, 30, 40, 50]$ for the Target Model, and another with a constant max depth of 8 and varying tree numbers: $[10, 50, 100, 300, 500]$ for the Target Model. The results are compared within each set of parameters.

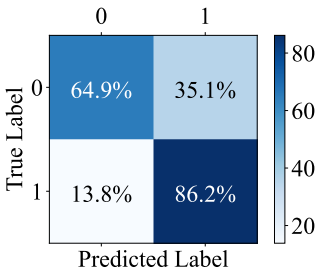
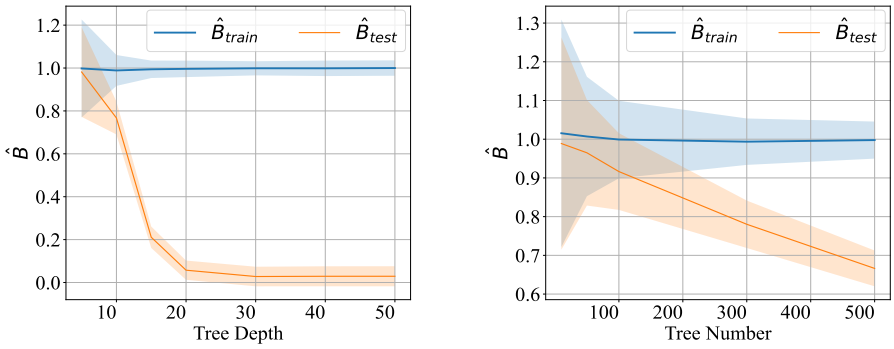


Fig. 3: Inference model's confusion matrix

Figure 4 illustrates the normalized geometric mean and standard deviation of D_s for both datasets. Figures 4a and 4b, respectively, demonstrate that increasing the depth and number of trees in the Target Model enhances the distinction between $\hat{B}_{train}$ and $\hat{B}_{test}$. This improvement occurs because we focus solely on numerical features. With greater tree depth and more trees in the RF model, these selected features are more likely to be used in defining tree node conditions. Consequently, we obtain more accurate ranges for these features, potentially enhancing the distinction between $\hat{B}_{train}$ and $\hat{B}_{test}$. The normalization process, using a randomly chosen reference subset from the training set ($\overline{B}_{ref}$), results in $\hat{B}_{train}$ converging to 1, confirming that training and reference subsets share similar characteristics, while the test dataset differs from them.

4 Conclusion

This paper proposes an approach to quantify RF models' vulnerability to MIA. We demonstrate that if an RF model and partial training data are public, the adversary can leverage this to conduct MIA against the remaining data. Our



(a) $\hat{B}$ according to tree depth

(b) $\hat{B}$ according to tree number

Fig. 4: Normalized geometric mean and standard deviation of D_s

method trains a new model to detect if a single sample was used in training the target Model. We performed several experiments to validate the efficiency of our approach, highlighting the potential security risks in RF models and emphasizing the need for robust privacy measures, particularly in the context of federated learning.

References

[1] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, and Z. Lin, When machine learning meets privacy: A survey and outlook, *ACM Computing Surveys (CSUR)*, 2021.

[2] Pascual, D. Pascual, A. Amirshahi, A. Aminifar, D. Atienza, P. Ryvlin, and R. Wattenhofer, EpilepsyGAN: Synthetic epileptic brain activities with privacy preservation, *IEEE Transactions on Biomedical Engineering*, 2020.

[3] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, Membership inference attacks against machine learning models, *proceedings of the 38th IEEE Symposium on Security and Privacy* (SP 2017), IEEE, pages 3-18, MAY 22-24, SAN JOSE, CA, USA.

[4] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, ML-leaks: Model and data independent membership inference attacks and defenses on machine learning models, *Network and Distributed Systems Security (NDSS) Symposium 2019*, San Diego, CA, USA.

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

[6] S. Yeom, I. Giacomelli, M. Fredrikson, Matt and S. Jha, Privacy risk in machine learning: Analyzing the connection to overfitting, *proceedings of the 31st computer security foundations symposium* ((CSF) 2018), IEEE, 268-282, July 9-12, Oxford, UK.

[7] Liu, Lan, et al. Membership inference attacks against machine learning models via prediction sensitivity, *IEEE Transactions on Dependable and Secure Computing*, 2022.

[8] J. Ferry, R. Fukasawa, T. Pascal, and T. Vidal, Trained Random Forests Completely Reveal your Dataset, *proceedings of the 41st International Conference on Machine Learning,* (ICML 2024), PMLR 235, July 21-27, Vienna, Austria.

[9] L. Buk Cardoso and V. Cunha Parro and S. Verzinhasse Peres and M. Curado and G. Fernandes and V. Wünsch Filho and T. Natasha Toporcov, Machine learning for predicting survival of colorectal cancer patients, *Scientific reports*, 2023.

[10] P.J. Fleming and J.J. Wallace, How not to lie with statistics: the correct way to summarize benchmark results, *Communications of the ACM*, 1986.