CIS 6930: Trustworthy Machine Learning
Final Report: Machine Unlearning for Randomized Decision Trees

Ashwin Rai
(Point of Contact)
raiashwin@ufl.edu

Vaibhav Kulkarni
kulkarniv@ufl.edu

Zubin Arya
z.arya@ufl.edu

December 10, 2021

Abstract

Computational feasibility is an important aspect while updating the model for retraining when user requests digital footprint from the dataset to be removed. It is obvious that it is not feasible to retrain the model from scratch for large datasets if every time selected data points are purged. To make model update computationally feasible, machine unlearning comes to the rescue. Since there has been limited scope of work done for machine unlearning on tree based models, we came up with study of machine unlearning on data removal-enabled forest (DaRE) models which is a variant of the random forest and focused on three quantifying metrics to evaluate the efficacy of the machine unlearning with respect to complete retraining which are data deletion threshold, space overhead and the time overhead.

Contents

1 Introduction

There is a rising need for users to safeguard their footprints by forcing search engines to remove their links about them from the past. The *Right to be forgotten* law which is General Data Protection Regulation(GDPR) and California Consumer Privacy Act(CCPA) addresses these issues and strives to enforce security and maintain the privacy of both individuals and community by and large.

Large organizations that consume and process public data are obligated to respect these laws and maintain the privacy and security of the users. In most cases, the organizations train machine learning models on these large data sets, and the problem arises when a part of the training data set decides to move out of the system and requests to erase their data, data lineage, any traces of their data along with its influence or contribution in training the machine learning model. Organizations around the world use diverse strategies to implement machine unlearning. A naive approach is to retrain a model from scratch, but it becomes computationally and resourcefully expensive in the case of large datasets.

In this project, we have chosen variant of randomized decision tree machine learning model which is DaRE and implemented machine unlearning without retraining the whole model. In the subsequent sections, we will evaluate computational feasibility and robustness of the machine unlearning using DaRE model against complete retraining. For evaluation, we will use quantifying metrics which are unlearning time computation, space overhead and effect of tuning hyper-parameters. In addition, we will test the membership inference attacks on the deleted instances to validate the claim of guaranteed unsuccessful membership inference attacks on deleted instances.

2 Background and Related Work

During the process of the work survey, we studied numerous pieces of literature, and we observed multiple approaches and strategies being used to achieve machine unlearning. For this project, we decided to work on the specific machine learning model type, the randomized decision tree. We referred to the paper "Humans Forget, Machines Remember" [3] to clearly understand and learn the legal background behind the Right to Be Forgotten. The technical papers that are strongly related to unlearning of Randomised decision tree were "Brophy and Lowd, Machine Unlearning for Random Forests" [2] and "Schelter et al. HedgeCut: Maintaining Randomised Trees for Low-Latency Machine Unlearning" [4] which emphasize on ensembling of small randomized decision trees to achieve low latency machine unlearning for classification type problems. We also referred to the literature "DART: Data Addition and Removal Trees" [1] which proposes a variant of Random Forests that supports the addition or removal of data. We believe "HedgeCut" and "data removal-enabled forests" (DaRE) are some of the efficient unlearning techniques developed for tree-based models.

Based on our findings from the above cited research papers especially, we were motivated to investigate further into the machine unlearning for DaRE models implemented in this paper [2] with focus on three key aspects. First is the operating limits and robustness of the unlearning solution by finding the threshold up to which data points can be removed while retaining the accuracy of the model in comparison to the completely retrained model. Secondly, to test the feasibility of the batch deletion through successive single instance deletions and validate claims of membership inference attacks.

3 Approach

We analyzed and studied the frameworks DARE and DART, thereby refactoring the strategy to build a model with support for single instance deletion unlearning and batch unlearning through sequential deletion.

For the computation and building of the model, we have used the local system running on Ubuntu 20.2 with Intel i5-10600K processor,16 GB memory and NVIDIA 3070ti Graphics.

We measure the speedup of the generated unlearned model in comparison with the completely retrained model. Speedup is the ratio of time for generating the model (unlearning vs. retraining). We evaluate the

delta between the unlearned model and retrained model accuracy on the test data. We choose random data points to be deleted uniformly. We aim to achieve speedup by a factor of two on average and have delta error within 5% with respect to the unlearned vs. retrained model.

3.1 Dataset(s)

We plan to work with the following privacy-sensitive datasets: Census Income Dataset[1] which predicts whether the annual income of an individual exceeds 50k USD, based on census data. Bank Marketing Dataset[2] which predicts whether the client will subscribe a term deposit or not and App Behavior Dataset[3] which predicts new customer who is interested in buying the product or not.

3.2 Plan of Action

- We started with the technical analysis of the paper referenced Jonathan Brophy and Lowd Daniel on Machine unlearning for random forests [2]. We built our own version of DaRE and implemented in python and built its relevant dependencies.
- In addition to the datasets used in the experimentation, we chose our own new dataset. App Behavior Dataset which was not used in the original experimentation, to validate the claims.
- Through experimentation, we determined the optimal size of the dataset (threshold) which can be unlearned for which unlearning time is less than the retraining time, which was not originally covered in the scope of the paper by Jonathan Brophy and Lowd Daniel [2].
- We performed time and space overhead analysis and tuned the hyperparameters (d_{rmax} and k) to improve the speedup and potentially extended the threshold for optimal deletion size.
- In addition, we validated the claim *deletions in random forest are exact, so Membership Inference attack is guranteed to be unsuccessful* by implementing Membership Inference attack on DaRE model which was a part of discussion and left for future scope.

3.3 Experiments and Validations

- After successfully implementing the DaRE, we performed the unlearning experiments for three data sizes, ie. **small**, **medium** and **large** for each dataset used for experimental purpose.
- Originally the DaRE model was overfitted on **app behavior analysis**, we tuned the parameters to avoid the overfitting.
- We faced some **challenges** in experimenting with the **app behaviour analysis** after data preprocessing using **columntransformer** with kmeansdiscretizer, we obtained large number of columns with one hot encoding which required high compute time for model training. We resolved this issue by performing data visualization followed by feature selection which successfully reduced the computation time.
- We tuned the following hyperparameters which are the maximum depth of each tree d_{max} (greedy) and d_{rmax} (random) , the number of trees in the forest T and the number of thresholds considered per attribute for greedy nodes k.

[1]Census Income Dataset Link: https://archive.ics.uci.edu/ml/datasets/adult
[2]Bank Marketing Dataset Link: https://archive.ics.uci.edu/ml/datasets/bank+marketing
[3]App Behavior Dataset Link: www.kaggle.com/hkhamnakhalid/customers-to-subscription-through-app-behavior

- From the previous work done by Brophy and Daniel and from our own experiments we identified that increasing d_{rmax} (random) and reducing k, can improve the speedup of the DaRE at the cost of small loss in the predictive performance for small number of unlearning samples (>1%) and for large data deletions, we keep tuning the hyper parameters to produce similar results and to obtain optimal values of d_{rmax} and k respectively.
- Space Overhead Computation: To test the impact of space consumption, we performed the space overhead analysis by comparing our version of DaRE with the ScikitLearn random forest.
- Implementation of Membership Inference Attack: We performed two attacks which are Shokri attack and Loss attack 2. We took into account all three datasets and trained the samples. We deleted some samples used in the training set from the each of the datasets and evaluated above chosen Membership Inference attacks on the deleted samples to check whether they were able to predict whether these samples were a part of the training set or not.

			Experiment 1- Data Size 10		Experiment 2- Data Size 5000		Experiment 3- Data Size 16000	
		Scores	Unlearning	Complete Retraining	Unlearning	Complete Retraining	Unlearning	Complete Retraining
Census Income	Train	Accuracy	94.82	94.83	93.39	93.37	89.99	89.9
		Precision	91.82	91.8	89.3	89.26	82.55	82.02
		Recall	86.17	86.21	82.42	82.38	74.12	74.43
	Test	Accuracy	84.0125	83.98	84.0875	84.0625	84.0125	83.93
		Precision	68.16	68.04	68.85	68.79	68.47	67.9
		Recall	58.98	59.03	57.96	57.91	58.23	58.87

Figure 1: Census Income Dataset: Unlearning vs Retraining results

			Experiment 1- Data Size 10		Experiment 2- Data Size 5000		Experiment 3- Data Size 16000	
		Scores	Unlearning	Complete Retraining	Unlearning	Complete Retraining	Unlearning	Complete Retraining
Bank Marketing	Train	Accuracy	99.375	99.375	97.8	97.9	94.83	94.82
		Precision	99.54	99.51	93.4	93.52	80.32	80.24
		Recall	94.94	94.97	87.6	87.63	72.13	72.21
	Test	Accuracy	90.71	90.6875	90.675	90.66	90.6875	90.675
		Precision	53.18	53.048	53.01	52.83	53.08	52.98
		Recall	41.69	41.18	40.54	41.82	40.8	40.92

Figure 2: Bank Marketing Dataset: Unlearning vs Retraining results

			Experiment 1- Data Size 10		Experiment 2- Data Size 6000		Experiment 3- Data Size 20000	
		Scores	Unlearning	Complete Retraining	Unlearning	Complete Retraining	Unlearning	Complete Retraining
App Behavior	Train	Accuracy	97.79	97.79	94.255	94.265	85.63	85.57
		Precision	98.105	98.14	94.213	94.272	85.16	85.14
		Recall	97.44	97.4	94.22	94.17	86.07	85.95
	Test	Accuracy	72.21	72.05	72.38	72.52	72.63	72.58
		Precision	71.6	71.4	71.74	71.89	72.015	71.99
		Recall	73.23	73.21	73.47	73.59	73.65	73.53

Figure 3: App Behaviour Dataset: Unlearning vs Retraining results

Dataset	Experiment 1- Unlearning size: Low		Experiment 2- Unlearning size: Medium		Experiment 3- Unlearning size: Large	
	Unlearning Time	Retraining Time	Unlearning Time	Retraining Time	Unlearning Time	Retraining Time
Bank Marketing	212.38 ms	1680.86 ms	15471.33 ms	13944.96 ms	43398.93 ms	9039.49 ms
Census Income	774.02 ms	29889.70 ms	37588.25 ms	25869.86 ms	107543.95 ms	16109.82 ms
App Behavior	1080.62 ms	78488.24 ms	31576.01 ms	43283.96 ms	35932.19 ms	17381.61 ms

Figure 4: Speedup Time for Unlearning vs Retraining in (ms)

S/N	Untuned hyperparameters		
	Data Points Deleted	Retraining Time	Unlearning Time
1	50	21.46 sec	3.35 sec
2	55	21.42 sec	3.63 sec
3	60	21.37 sec	4.08 sec
4	65	21.34 sec	4.22 sec
5	70	21.32 sec	4.37 sec
6	75	21.28 sec	4.44 sec

(a) Parameter Untuned

S/N	Tuned hyperparameters		
	Data Points Deleted	Retraining Time	Unlearning Time
1	50	20.9 sec	3.092 sec
2	55	20.75 sec	3.37 sec
3	60	20.72 sec	3.74 sec
4	65	20.65 sec	3.91 sec
5	70	20.61 sec	4.06 sec
6	75	20.63 sec	4.18 sec

(b) Parameter Tuned

Figure 5: App Behaviour Dataset: Observed Speedup in unlearning post tuning parameters

	Dataset: Bank Marketing: Memory usage (in megabytes) for the training data, G-DARE RF, and a SKLearn RF trained using the same values of T and dmax as G-DARE RF.					
DataSetSize	Structure Memory	Decision Stats Memory	Leaf Stats Memory	Total Memory (DaRE)	ScikitLearn RF MemUsage	Memory Overhead Factor
10	0.07	0.59	0.06	0.72	0.24	3x
110	0.68	6.89	0.53	8.1	1.12	7.23x
210	1.25	13.35	1	15.6	2.08	7.5x
310	1.83	19.84	1.47	23.14	2.95	7.84x
410	2.46	27.15	1.96	31.57	3.97	7.95x
510	3.07	34.34	2.44	39.85	4.96	8.03x
610	3.7	42.3	2.92	48.92	6.22	7.86x
710	4.19	48.23	3.37	55.79	7	7.97x
810	4.79	55.63	3.85	64.27	8.07	7.96x
910	5.35	62.72	4.31	72.38	8.97	8.07x

Figure 6: Bank Marketing Dataset: Memory usage (MB) comparison of Total Memory(DaRE) vs Standard Random Forest by ScikitLearn

	Dataset: Census Income: Memory usage (in megabytes) for the training data, G-DARE RF, and a SKLearn RF trained using the same values of T and dmax as G-DARE RF.					
DataSetSize	Structure Memory	Decision Stats Memory	Leaf Stats Memory	Total Memory (DaRE)	ScikitLearn RF MemUsage	Memory Overhead Factor
10	0.15	1.6	0.08	1.83	0.36	5.08x
110	1.16	14.77	0.67	16.6	2.13	7.79x
210	2.1	28.33	1.25	31.68	3.67	8.63x
310	3.06	42.47	1.83	47.36	5.24	9.04x
410	3.88	54.42	2.37	60.67	6.64	9.14x
510	4.56	64.33	2.86	71.75	7.83	9.16x
610	5.4	77.31	3.41	86.12	9.2	9.36x
710	6.3	91.17	3.97	101.44	10.7	9.48x
810	7.27	106.31	4.55	118.13	12.24	9.65x
910	8.19	121.01	5.12	134.32	13.74	9.78x

Figure 7: Census Income Dataset: Memory usage (MB) comparison of Total Memory(DaRE) vs Standard Random Forest by ScikitLearn

	Dataset: App Behavior: Memory usage (in megabytes) for the training data, G-DARE RF, and a SKLearn RF trained using the same values of T and dmax as G-DARE RF.					
DataSetSize	Structure Memory	Decision Stats Memory	Leaf Stats Memory	Total Memory (DaRE)	ScikitLearn RF MemUsage	Memory Overhead Factor
10	0.13	1.19	0.07	1.39	0.34	4.09x
110	1.45	15.18	0.76	17.39	2.19	7.94x
210	2.79	29.97	1.44	34.2	4.15	8.24x
310	4.28	46.7	2.17	53.15	6.2	8.57x
410	5.43	59.34	2.81	67.58	8.05	8.4x
510	6.66	73.43	3.47	83.56	9.85	8.48x
610	8.05	89.16	4.17	101.38	11.92	8.51x
710	9.25	103.02	4.82	117.09	13.75	8.52x
810	10.43	116.8	5.46	132.69	15.6	8.51x
910	11.78	132.92	6.15	150.85	17.55	8.6x

Figure 8: App Behaviour Dataset: Memory usage (MB) comparison of Total Memory(DaRE) vs Standard Random Forest by ScikitLearn

DataSet	Shokri Attack Accuracy (%)	Loss Attack 2 Accuracy (%)
Bank Marketing	40.8	26.2
Census Income	36.45	29.15
App Behaviour	36.25	30.2

Figure 9: Membership Inference Attack: Accuracy Comparison of Shokri and Threshold Loss Attack on three datasets

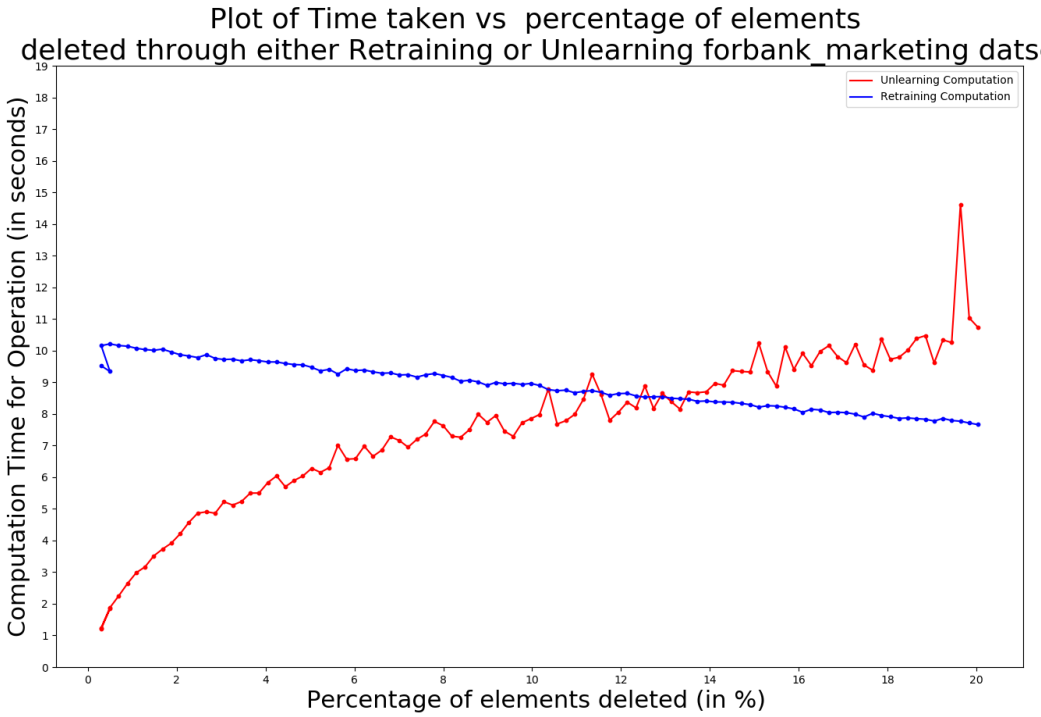

Figure 10: Bank Marketing Dataset: Computation time as a function of percentage of elements deleted - for retraining vs unlearning

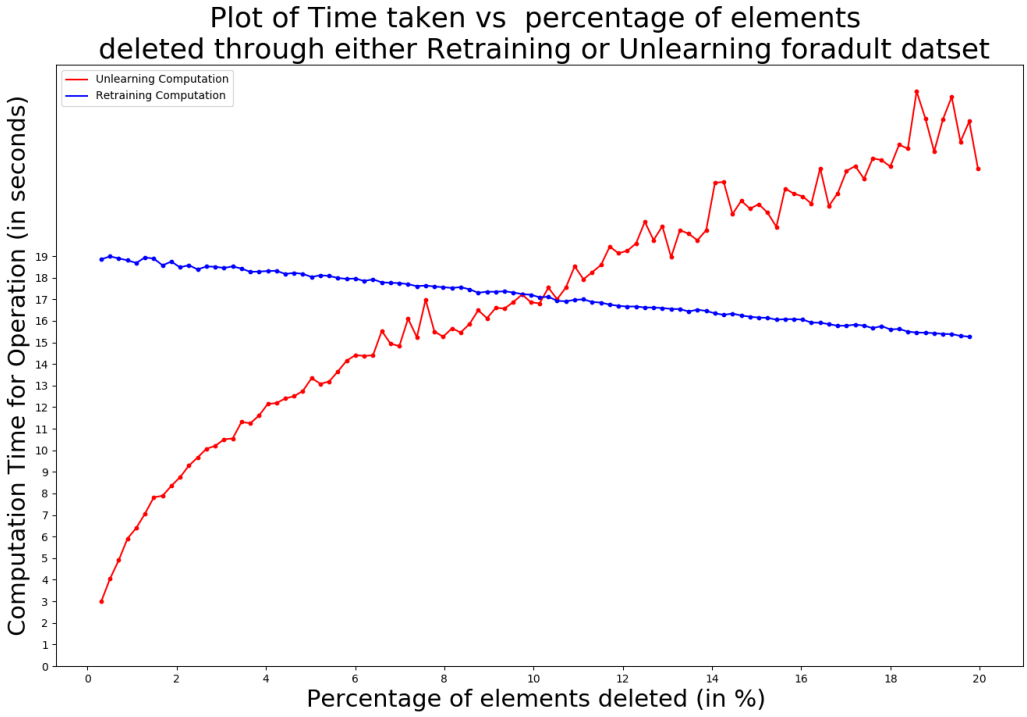

Figure 11: Census Income Dataset: Computation time as a function of percentage of elements deleted - for retraining vs unlearning

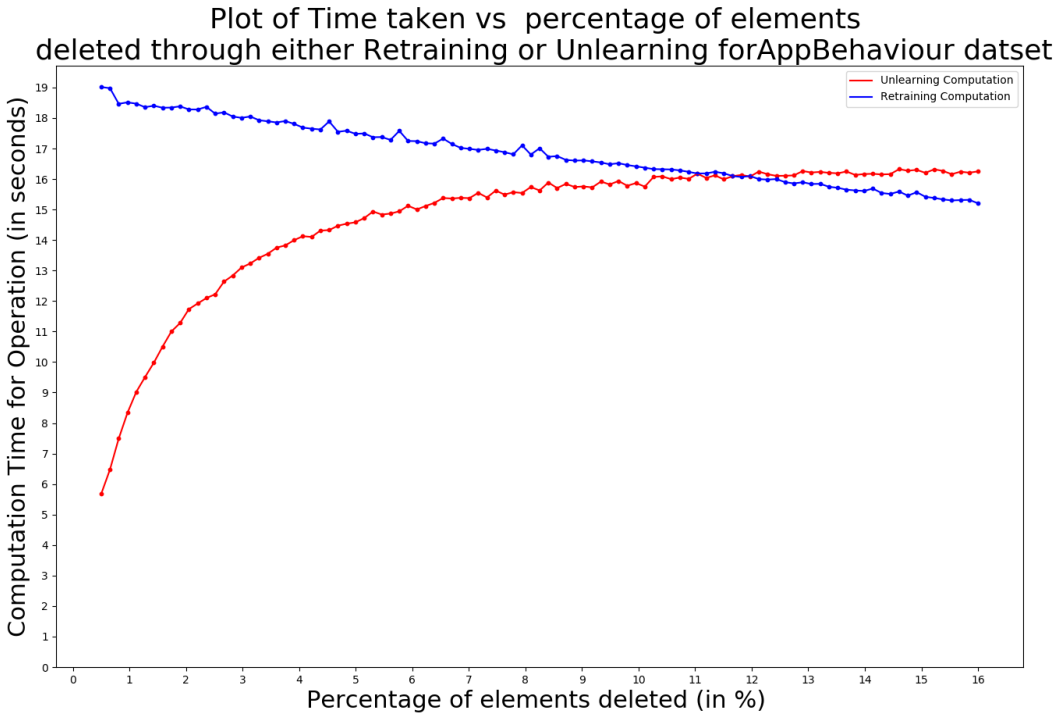

Figure 12: App Behaviour Dataset: Computation time as a function of percentage of elements deleted - for retraining vs unlearning

4 Results

- Post unlearning we computed confusion matrix followed by precision, recall and accuracy scores. We compared the performance of unlearned model vs. completely retrained model for three datasets to produce the following results in the tables 1, 2 and 3 .
- We find that the generalization error does not vary significantly and the model is able to unlearn data in 3-4 orders of magnitude faster than retraining from scratch while sacrificing less than 1% in terms of predictive performance for which results are reported in 4.
- We observed as we increase the number of indices to be removed the speedup is less significant as compared to retraining the left portion and its better to actually retrain rather than unlearn which roughly happens when unlearning around 11% of dataset as evident from the analysis of computation time with varying percentage of elements deleted for all three datasets referenced in 10, 11 and 12.
- Unlearning Time Speedup Analysis: We observed the improve in speedup for small data unlearning 5, but did not see the same results being reproduced as we increase the unlearning size shown.
- Space Computation Analysis: Our experimental findings match with the theoretical claims as we observed an overhead in space consumption by a factor of 3-10 when compared to the simple random forest by the scikit learn for which the results are 6, 7 and 8.
- Membership Inference Attack Analysis: We formulated the attack accuracy results shown in 9 and found that the attack accuracy results for Shokri and Loss Attack 2 are in the range (26%-40%) for all the three datasets. This complements the failure of the membership inference attacks to predict the deleted samples in or out of the training set. Considering the data deletions as exact in the DaRE models, membership inference attacks failed and hence we validated the claim by the researchers that *data deletions in DaRE models are exact, membership inference attacks are guaranteed to be unsuccessful for instances deleted from the model.*

5 Conclusions & Future work

We have successfully implemented DaRE, tested its robustness and operating limits of unlearning by performing unlearning tests for varying deletion size (small, medium, large). We performed time and space analysis to understand DaRE and tuned hyper-parameters to avoid overfitting and improve the speedup in order to achieve threshold for optimal deletion greater than 11%. After the hyper-parameter tuning, we noticed speedup in the unlearning for smaller data deletions but this difference does not translate to the large data deletions. We find that the claim holds true regarding membership inference attacks and membership inference attacks are unsuccessful on DaRE model under black box settings.

Future Work: We will try to bring down space overhead which will enhance the performance of DARE in 11+% data unlearning cases. In addition to that, we want to implement and test differential-private random forest models, but the problem lies in the large privacy budget and generalization error due to which they often suffer from poor predictive performance.

References

[1] Jonathan Brophy and Lowd Daniel. Dart: Data addition and removal trees. *CoRR abs/2009.05567*, 2020.

[2] Jonathan Brophy and Lowd Daniel. Machine unlearning for random forests. *ICML*, 2021.

[3] Villaronga Eduard Fosch, Kieseberg Peter, and Tiffany Li. Humans forget, machines remember: artificial intelligence and the right to be forgotten. *Computer Law and Security Review*, 2018.

[4] Sebastian Schelter, Grafberger Stefan, and Dunning Ted. Hedgecut: Maintaining randomised trees for low-latency machine unlearning. *ACM SIGMOD*, March, 2021.