Machine Unlearning for Random Forest via Method of Images

Hang Zhang and Kai Ming Ting(✉)

State Key Laboratory for Novel Software Technology & School of Artificial Intelligence,
Nanjing University, Nanjing, 210023 China
zhanghang@lamda.nju.edu.cn, tingkm@nju.edu.cn

Abstract. Privacy law can now demand specific training samples, if requested from concerned parties, to be deleted from a trained model. Random forest, an effective and widely used machine learning algorithm, has been the model of study for various machine unlearning techniques. The current unlearning techniques of random forest involve additional processing before model training, so that fast unlearning of some samples can be achieved. However, no algorithm can achieve the unlearning of a trained random forest. This paper proposes a novel algorithm for unlearning a trained random forest. The algorithm employs the method of images to generate image samples of the samples that need to be forgotten and trains a small number of additional decision trees on these image samples. The proposed method, called MUMI, enables efficient unlearning of samples from a trained random forest. Our theorems and experiments show that MUMI achieves fast unlearning in a trained random forest with virtually no loss of model performance.

Keywords: Machine unlearning · Random forest · Method of images.

1 Introduction

In recent years, the rapid development of artificial intelligence has brought many conveniences to the lives of humans [40,9,33]. However, technology has always been a double-edged sword. While artificial intelligence improves humans' lives, it also poses challenges to the privacy and security of people [8,26]. For this reason, institutions such as GDPR (General Data Protection Regulation) have enacted laws to protect users' rights to delete their data [34,32,20,23]. After a user's request to delete their data, their data must not only be deleted from the database but also from the machine learning models that have used these data for training. The task of erasing the influence on a trained model, of the samples that have previously been used to train it, is called machine unlearning [2,23].

To comply with legal requirements and address practical needs, many algorithms have been proposed for unlearning different machine learning models, e.g., decision tree [41], logistic regression [21], Markov chain Monte Carlo [10], or other specific models [1,24,28,13,15]. And some machine unlearning algorithms can be used to unlearn different kinds of models [5,2,12,37,25,29,7,14,16,17,27,39,30].

Random forest is widely used as a model with superior performance and high interpretability [3]. However, since it selects discrete dimensions and thresholds for the tree split each time, it is discontinuous and cannot use gradient information for machine unlearning like logistic regression [21] and neural networks [19]. Random forest ensembles multiple decision trees to improve prediction accuracy and robustness. Each decision tree selects the optimal split attribute and threshold based on the Gini index and entropy. In the process of unlearning, the optimal attribute and threshold may change after deleting some points, and calculating the new optimal attribute and threshold on the entire subtree is complicated. This presents challenges for the unlearning of random forest.

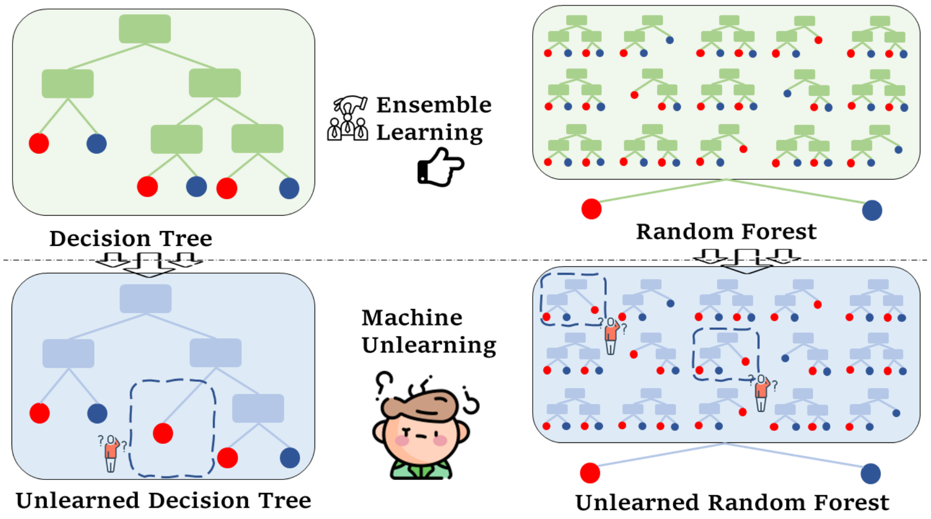

Fig. 1. An illustration of machine unlearning for random forest.

An illustration of machine unlearning for random forest is shown in Fig. 1. Random forest is obtained by ensemble learning of decision trees, each tree is trained on the randomly selected subset of the entire data. When we delete some data, some trees in the random forest need to be revised.

Existing unlearning methods for random forest attempt to pay more cost before training so that unlearning can be achieved at a small cost when required. The DaRE tree [4] constructs random forest by employing randomness at most nodes near the root, where attributes and thresholds are sampled randomly. In contrast, only a few layers close to the leaf nodes are optimized using greedy methods, guided by criterion such as the Gini index or mutual information. This hybrid approach ensures that the tree remains computationally efficient while maintaining accuracy. When samples need to be removed or 'forgotten', only a few layers (subtrees) near the leaf nodes require retraining. This selective retraining reduces the computational overhead and enables rapid unlearning within the random forest, making DaRE a highly efficient solution for scenarios where data removal is necessary. HedgeCut [36] learns an ensemble of randomized decision trees with randomly chosen splits, it classifing nodes into robust and non-robust nodes based on whether they are easily affected by data deletion. For robust nodes, when data is deleted, only the node statistics need to be modified. For non-robust nodes, variant seed trees are prepared in advance, and when data is deleted, variants are used to replace them. Existing methods can only modify the training process before training so as to quickly unlearn the points that need to be deleted, however, **no existing method can achieve fast unlearning of the already trained random forest model currently.**

In this paper, we introduce Machine Unlearning based on Method of Images (MUMI), a novel approach that leverages the method of images [22] to enable unlearning in a trained random forest. Specifically, we generate image samples of the samples that need to be forgotten and construct additional decision trees based on these image samples. By ensemble these additional trees with the already trained random forest, MUMI effectively achieves the unlearning of the target samples while maintaining the classification accuracy.

We summarize our contributions as follows:

1. Introducing the method of images into the machine unlearning task for the first time.
2. Proposing the first machine unlearning algorithm MUMI, based on the method of images, to unlearn a trained random forest.
3. Demonstrating the efficiency and effectiveness of MUMI through experiments on real-world datasets.

2 Related Work

Decision trees [38] are a class of tree-structured models that facilitate binary predictions through a hierarchical decision-making process. Each leaf node represents a final prediction, while each internal node functions as a decision point, associated with a specific attribute and a threshold value. Each decision node partitions the data into branches based on a selected attribute and its threshold. For a given test point $x \in X$, its prediction is determined by traversing the tree from the root node, following the branches that comply with the attribute values, until reaching a leaf node, where the prediction is derived from the leaf's class label. A decision tree is constructed recursively by selecting an attribute and threshold at the root node that optimizes a chosen empirical split criterion. Two commonly used criteria are the Gini index [11] and entropy [35].

Random forest extends decision trees by creating an ensemble of multiple trees to improve prediction accuracy and robustness. The ensemble predicts the average value of its constituent trees. To introduce diversity among the trees, two sources of randomness are employed. First, each tree is trained on a bootstrap sample of the original data, which allows some instances to be excluded or repeated.

Second, at each decision node, only a random subset of attributes is considered for splitting, rather than all attributes.

The DaRE [4] tree leverages randomness and caching mechanisms to enhance the efficiency of data removal. In the upper levels of the DaRE tree, random nodes are employed. These nodes uniformly select split attributes and thresholds randomly. As a result, they rarely require updates, since their behavior is largely independent of the underlying data distribution. At the lower levels, splits are determined through a greedy optimization process, targeting criteria such as the Gini index or mutual information. This approach ensures that splits are made in a way that maximizes the purity or information gain of the resulting partitions. To further optimize performance, the DaRE tree caches statistics at each node and stores training data at each leaf. This design allows for selective updates: when data is removed, only the necessary subtrees need to be adjusted, rather than the entire tree. The DaRE tree can effectively trade-off between prediction accuracy and update efficiency.

HedgeCut [36] trains an ensemble of randomized decision trees, where each tree is built using random splits on randomly selected attributes. It continuously manages this ensemble even when some training samples are removed. In some trees, certain splits are non-robust, meaning that the decision to split could change after data removal. To handle this, HedgeCut updates the statistics of leaf labels and maintains alternative subtree structures. If the removal of data would have caused the model to choose a different split, these alternative subtrees are activated to ensure the model remains consistent.

In essence, both DaRE and HedgeCut require additional procedures prior to training the random forest model. These additional steps alter the conventional training process. It can be argued that they do not truly facilitate the unlearning of a random forest in its original form. Instead, they introduce a modified version of the random forest that is more amenable to the unlearning process. Given the inherent structure and training process of a random forest, it appears that there is no feasible method to induce machine unlearning in a trained random forest without fundamentally altering its training process.

3 Method of Images for Unlearning Random Forest

3.1 Method of Images

The method of images is a powerful and elegant technique in classical physics, particularly in the study of electrostatics and gravitational fields [22]. It is based on the principle of superposition and the uniqueness theorem, which states that if a solution to Laplace's equation satisfies the boundary conditions, it is the only possible solution.

In electrostatics, the method of image is often used to solve problems involving conductors with complex geometries or boundaries. For example, when dealing with a point charge q in $(0, 0, a)$ near an infinite conducting plane as shown in Fig. 2, instead of solving the complicated boundary value problem directly, we

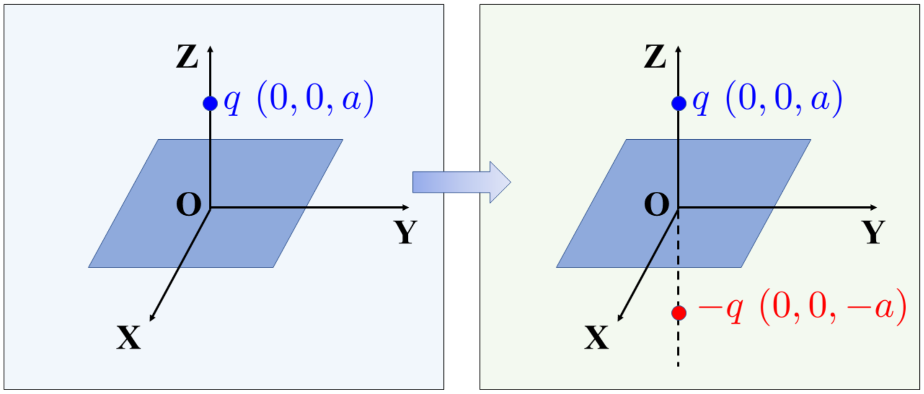

Fig. 2. An example of the method of images.

can introduce an image charge $-q$ on the opposite side of the plane $(0,0,-a)$. This image charge is chosen such that its magnitude and position ensure that the potential on the conducting surface is zero, as required by the boundary conditions. By doing so, the problem is simplified to finding the potential due to the (opposite) charges of the two points, which can be easily calculated using Coulomb's law.

3.2 Machine Unlearning based on Method of Images (MUMI)

The beauty of the method of images lies in its ability to transform a seemingly intractable problem into a much simpler one. After training a machine learning model $A(\mathcal{D})$ using data $\mathcal{D}$, a data subset $\mathcal{D}_f$ is required to be deleted from $A(\mathcal{D})$, and the machine unlearning uses algorithm U to obtain a model $U(A(\mathcal{D}),\mathcal{D},\mathcal{D}_f)$, which is equivalent to the model $A(\mathcal{D}\setminus\mathcal{D}_f)$ trained from $\mathcal{D}\setminus\mathcal{D}_f$. However, due to the complexity of the model (such as random forest), it is difficult to obtain $U(A(\mathcal{D}),\mathcal{D},\mathcal{D}_f)$ without retraining, because the influence of deleting $\mathcal{D}_f$ from $\mathcal{D}$ on the model is difficult to measure. Inspired by the method of images, deleting sample $\mathcal{D}_f=\{X_f,Y_f\}$ from the model $A(\mathcal{D})$, for the purpose of machine unlearning, is equivalent to adding image sample set $\mathcal{D}_f^I=\{X_f,-Y_f\}$ to model $A(\mathcal{D})$ to obtain a new model $\Lambda(A(\mathcal{D}),\mathcal{D}_f)$, in order to offset the information contained in $\mathcal{D}_f$.

An illustration of employing the method of images is shown in Fig. 3. We typically use the orange curve to represent the influence of data $\mathcal{D}$ on model $A(\mathcal{D})$. When we are required to delete a data point $\mathcal{D}_f=\{x,y\}$ (where x is the training sample and y is the corresponding label) from model $A(\mathcal{D})$, directly computing the new influence curve becomes challenging. To address this issue, we can employ a method analogous to the 'Method of Images' in physics. Specifically, we introduce an 'image' data point $(x,-y)$ into the dataset. By considering

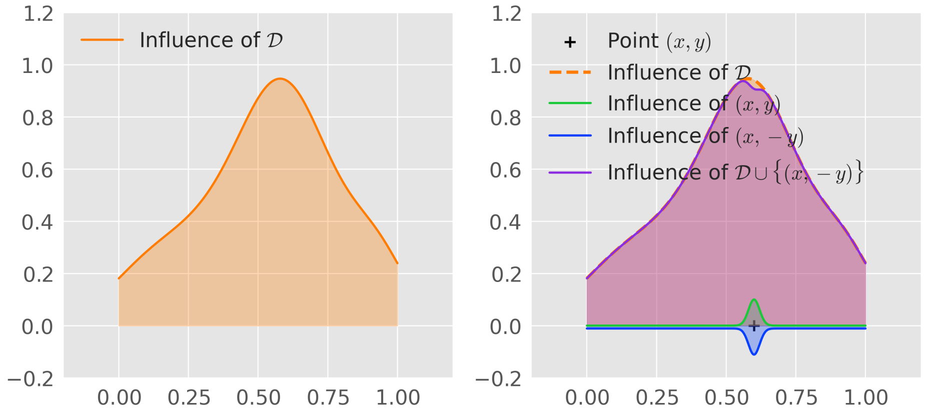

Fig. 3. An illustration of the method of images in machine unlearning. The influence of data $\mathcal{D}$ on model $A(\mathcal{D})$ is shown on the left. The influence of data $\mathcal{D} \setminus \{x, y\}$ is approximated by the method of images is shown on the right.

the influence curve of the augmented dataset $\mathcal{D} \cup \{x, -y\}$, we can approximate the influence curve of the reduced dataset $\mathcal{D} \setminus \{x, y\}$. This approach effectively leverages the symmetry of the influence function, allowing us to bypass the complexity of directly recomputing the influence curve after deletion.

Algorithm 1 MUMI: $\Lambda(A(\mathcal{D}), \mathcal{D}_f)$

```
Input: Model $A(\mathcal{D})$, Data to delete $\mathcal{D}_f = \{X_f, Y_f\}$
Output: Model $\Lambda(A(\mathcal{D}), \mathcal{D}_f)$
Generate image data: $\mathcal{D}_f^I = \{X_f, -Y_f\}$.
Train image decision trees $A(\mathcal{D}_f^I)$ on the image data set $\mathcal{D}_f^I$.
Ensemble the original model $A(\mathcal{D})$ and the image model $A(\mathcal{D}_f^I)$ to obtain the
    unlearned model: $\Lambda(A(\mathcal{D}), \mathcal{D}_f) \leftarrow \text{Ensemble}(A(\mathcal{D}), A(\mathcal{D}_f^I))$.
return $\Lambda(A(\mathcal{D}), \mathcal{D}_f)$
```

For the unlearning of random forest, we propose a **M**achine **U**nlearning algorithm based on the **M**ethod of **I**mages (MUMI) as shown in Algorithm 1. Given the model $A(\mathcal{D})$ and the data $\mathcal{D}_f$ that need to be deleted, MUMI first generates the image data $\mathcal{D}_f^I = \{X_f, -Y_f\}$, and then train the decision trees $A(\mathcal{D}_f^I)$ on the image data set $\mathcal{D}_f^I$. Finally, MUMI ensembles the original model (original decision trees) $A(\mathcal{D})$ and the image model (image decision trees) $A(\mathcal{D}_f^I)$ to obtain $\Lambda(A(\mathcal{D}), \mathcal{D}_f)$. By employing the method of images, MUMI achieves data unlearning with significantly reduced computational cost. Specifically, it only requires training on the much smaller image dataset $\mathcal{D}_f^I \ll \mathcal{D}_r(\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f)$. This approach eliminates the need to retrain on the remaining dataset $\mathcal{D}_r$, thereby substantially decreasing the computational resources and time required for the machine unlearning.

In the example illustrated in Fig. 2, the boundary condition is defined by the electric potential of the conductor being zero. Satisfying this boundary condition is a crucial requirement for the method of images to be valid. Similarly, in the context of MUMI, we utilize the performance of the unlearned model on datasets X_r and X_f as the boundary condition. Intuitively, we aim to make the unlearned model to perform well on X_r while deliberately underperforming on X_f. This strategy is based on the assumption that a retrained model (a model that has completely forgotten X_f) would naturally exhibit strong performance on X_r, as it retains the relevant information from X_r, and weaker performance on X_f, since it no longer has access to the information from X_f [7].

Theorem 1. *Let E_r and E_{T+k} denote the generalization errors of the model obtained through retraining and MUMI, respectively. Then:*

$$|E_{T+k} - E_r| \ll E_r.$$

Theorem 1 shows that the generalization error of the model unlearned by MUMI is close to the generalization error of the retrained model. This guarantees that the model has good performance after unlearning.

Theorem 2. *Let $f_{RF}^{(T+k)}(x)$ and y_f denote the predicted label and the true label of x, respectively. The expectation error of the model unlearned by MUMI on $\mathcal{D}_f$ is:*

$$\mathbb{E}_{\{x,y_f\}\in\mathcal{D}_f}|f_{RF}^{(T+k)}(x) - y_f| = \frac{2k}{T+k},$$

where k is the number of image decision trees, T is the number of initial decision trees.

Theorem 2 shows that MUMI can effectively forget $\mathcal{D}_f$. In particular, when $k = T$, the expected error on $\mathcal{D}_f$ is 1. However, in practice, it is not necessary to set k too large, because the expected error on D_f is not necessarily better when it is smaller [14]. As long as it closely approximates the performance of the retrained model, it is deemed sufficient.

The proofs of Theorem 1 and Theorem 2 are provided in the supplementary materials.

4 Experimental Evaluation

Experimental Aims: We experimentally evaluate whether MUMI has the following three capabilities:

1. Model effectiveness: whether the model maintains good classification accuracy after unlearning.

2. Unlearning efficacy: whether the data is unlearned from the model.
3. Unlearning efficiency: whether the data can be unlearned efficiently.

Datasets: We conduct our experiments on 13 publicly available datasets used in the previous paper [4] that represent problems well-suited for tree-based models. We use the code provided in DaRE [4] to split the data into training and test sets (X_t). For each dataset, we generate one-hot encodings for any categorical variable and leave all numeric and binary variables as is.

Table 1. Summary of the datasets used in the paper.

datasets	n	%pos.	# cat.	# num.	# attr-hot	Met.
Surgical	14,635	25.2	17	7	90	Acc.
Vaccine	21,365	46.6	36	0	185	Acc.
Adult	48,842	23.9	8	5	107	Acc.
Bank Mktg.	41,188	11.3	10	10	63	AUC
Flight Delays	100,000	19	6	2	648	AUC
Diabetes	101,766	46.1	36	7	258	Acc.
No-Show	110,527	20.2	2	15	98	AUC
Olympics	206,165	14.6	8	3	1004	AUC
Census	299,285	6.2	30	6	408	AUC
Credit Card	284,807	0.2	0	29	29	AP
CTR	1,000,000	2.9	0	13	13	Acc.
Synthetic	1,000,000	50	0	40	40	Acc.
Higgs	11,000,000	53	0	28	28	Acc.

[*] n: the number of points, %pos.: positive label percentage.
[⋆] # cat.: the number of categorical attributes, # num.: the number of numeric attributes, # attr-hot: the number of one-hot attributes.
[†] Met.: predictive performance metric.

Evaluation Metrics: To account for the varying degrees of label imbalance in the datasets, we evaluate the predictive performance of models using different metrics based on the proportion of positive labels. Specifically, we use:

1. Average Precision (AP) [42] for datasets with a positive label percentage of less than 1%.
2. Area Under the ROC Curve (AUC) [18] for datasets with a positive label percentage between 1% and 20%.
3. Accuracy (Acc.) for datasets with a positive label percentage exceeding 20%.

This approach allows us to select the most appropriate metric for each dataset, ensuring a fair and meaningful assessment of model performance across different levels of class imbalance.

A summary of the datasets is shown in Table 1, and the details about the datasets are available in supplementary materials.

Comparison algorithms: we compare MUMI [1] with the Retrain (retrain a model on $\mathcal{D}_r$), DaRE [2] and HedgeCut [3].

More details about experimental settings, such as the random forest model used and the parameter settings, are given in the supplementary materials.

4.1 Model Effectiveness and Unlearning Efficacy

For each dataset, we report the scores of the unlearned model on the X_t (model effectiveness) and X_f (unlearning efficacy). We evaluate the performance of MUMI when unlearning 10, 100, 1,000, and 0.1%$\times n$ samples, with the results presented in Tables 2, 3, 4, and 5, respectively.

Table 2. The results of unlearning 10 samples.

Dataset	X_t (↑)				X_f (↓)			
	Retrain	DaRE	HedgeCut	MUMI	Retrain	DaRE	HedgeCut	MUMI
Surgical	0.932	0.917	0.724	**0.939**	0.800	0.700	**0.600**	0.700
Vaccine	0.921	0.902	0.783	**0.922**	**0.800**	0.900	**0.800**	**0.800**
Adult	**0.885**	0.858	0.835	**0.885**	**0.900**	**0.900**	**0.900**	**0.900**
Bank Mktg.	0.990	0.990	0.786	**0.991**	1.000	1.000	**0.000**	1.000
Flight Delays	0.800	**0.836**	0.722	0.796	0.875	**0.375**	0.438	0.875
Diabetes	0.713	0.667	0.605	**0.714**	0.800	0.800	**0.700**	0.800
No-Show	0.828	**0.865**	0.503	0.826	0.556	**0.444**	**0.444**	0.556
Olympics	0.811	**0.856**	0.525	0.805	1.000	1.000	**0.438**	1.000
Census	**0.962**	0.945	0.500	**0.962**	1.000	1.000	**0.000**	1.000
Credit Card	**0.993**	0.958	N/A	**0.993**	**1.000**	**1.000**	N/A	**1.000**
CTR	0.873	0.696	N/A	**0.875**	0.556	0.778	N/A	**0.444**
Synthetic	0.941	0.842	0.845	**0.942**	1.000	0.800	**0.600**	1.000
Higgs	**0.744**	0.678	N/A	0.743	**0.700**	**0.700**	N/A	0.743
Average Score	**0.876**	0.847	0.683	**0.876**	0.845	0.800	**0.492**	0.832
Average $\lvert s - s_R \rvert$	-	0.044	0.195	**0.002**	-	0.088	0.381	**0.018**

* N/A means an error occurred during the run or the result could not be output within 12 hours.

⋆ s is the scores of algorithms, and the s_R is the scores of Retrain.

In each of 10, 100, 1,000, or 0.1% $\times n$ samples for unlearning, MUMI consistently outperforms other algorithms on X_t and achieves the highest score (see the second row from the bottom 'Average Score' in the table). In contrast, both DaRE and HedgeCut exhibits lower scores compared to retraining. This is attributed to the fact that HedgeCut employs extremely randomized trees, whereas DaRE only optimizes split attributes and thresholds in the layers close to the leaf nodes.

[1] MUMI: `https://anonymous.4open.science/r/MUMI-8A0A`

[2] DaRE: `https://github.com/jjbrophy47/dare_rf`

[3] HedgeCut: `https://github.com/schelterlabs/hedgecut`

Table 3. The results of unlearning 100 samples.

Dataset	X_t (↑)				X_f (↓)			
	Retrain	DaRE	HedgeCut	MUMI	Retrain	DaRE	HedgeCut	MUMI
Surgical	0.933	0.917	0.726	**0.960**	0.830	0.770	**0.740**	0.830
Vaccine	0.923	0.902	0.785	**0.935**	0.800	0.800	**0.780**	**0.780**
Adult	0.885	0.857	0.833	**0.886**	0.840	0.830	**0.800**	0.830
Bank Mktg.	0.991	0.990	0.787	**0.992**	0.960	0.944	**0.489**	0.917
Flight Delays	**0.799**	0.836	0.722	0.796	0.737	0.742	**0.497**	0.682
Diabetes	0.712	0.667	0.610	**0.714**	0.730	0.640	**0.630**	0.730
No-Show	**0.830**	0.864	0.503	0.829	0.626	0.631	**0.494**	0.592
Olympics	0.811	**0.856**	0.527	0.803	0.796	0.813	**0.489**	0.728
Census	**0.963**	0.945	0.500	0.962	0.966	0.933	**0.495**	0.963
Credit Card	**0.994**	0.959	N/A	0.992	**1.000**	**1.000**	N/A	**1.000**
CTR	0.873	0.696	N/A	**0.876**	0.525	0.596	N/A	**0.253**
Synthetic	0.941	0.842	0.841	**0.942**	0.950	**0.800**	0.820	0.950
Higgs	**0.744**	0.678	N/A	0.743	0.790	**0.720**	N/A	0.760
Average Score	0.877	0.847	0.683	**0.879**	0.812	0.786	**0.623**	0.770
Average $\|s - s_R\|$	-	0.044	0.195	**0.004**	-	**0.038**	0.200	**0.038**

Table 4. The results of unlearning 1000 samples.

Dataset	X_t (↑)				X_f (↓)			
	Retrain	DaRE	HedgeCut	MUMI	Retrain	DaRE	HedgeCut	MUMI
Surgical	0.934	0.924	0.722	**0.982**	0.834	0.806	0.761	**0.742**
Vaccine	0.924	0.904	0.785	**0.956**	0.765	0.762	0.761	**0.721**
Adult	0.884	0.857	0.833	**0.893**	0.866	0.850	**0.833**	0.851
Bank Mktg.	0.991	0.991	0.784	**0.994**	0.929	0.921	**0.613**	0.911
Flight Delays	0.796	**0.837**	0.724	0.797	0.694	0.702	**0.528**	0.683
Diabetes	0.712	0.668	0.609	**0.718**	0.641	0.629	**0.597**	0.617
No-Show	0.827	**0.864**	0.515	0.836	0.690	0.691	**0.499**	0.642
Olympics	0.807	**0.856**	0.527	0.804	0.784	0.799	**0.512**	0.753
Census	**0.963**	0.945	0.500	**0.963**	0.965	0.955	**0.499**	0.966
Credit Card	**0.996**	0.956	N/A	0.993	**1.000**	**1.000**	N/A	**1.000**
CTR	0.874	0.696	N/A	**0.877**	0.681	0.655	N/A	**0.621**
Synthetic	0.941	0.842	0.834	**0.944**	0.931	0.840	**0.837**	0.927
Higgs	0.744	0.678	N/A	**0.745**	0.755	0.709	N/A	**0.662**
Average Score	0.876	0.847	0.683	**0.885**	0.810	0.794	**0.644**	0.777
Average $\|s - s_R\|$	-	0.045	0.195	**0.009**	-	**0.019**	0.166	0.032

Table 5. The results of unlearning 0.1%×n samples.

Dataset	X_t (↑)				X_f (↓)			
	Retrain	DaRE	HedgeCut	MUMI	Retrain	DaRE	HedgeCut	MUMI
Surgical	0.931	0.917	0.722	**0.951**	0.818	0.727	**0.667**	0.818
Vaccine	0.922	0.902	0.787	**0.929**	**0.714**	**0.714**	0.810	**0.714**
Adult	0.885	0.857	0.833	**0.888**	0.844	0.844	**0.788**	0.844
Bank Mktg.	**0.991**	0.990	0.776	0.990	0.955	0.893	**0.448**	0.964
Flight Delays	0.796	**0.836**	0.724	0.796	0.711	0.704	**0.469**	0.637
Diabetes	**0.713**	0.667	0.610	0.711	0.741	0.642	**0.605**	0.716
No-Show	0.828	**0.864**	0.504	0.825	0.636	0.652	**0.493**	0.583
Olympics	0.803	**0.856**	0.527	0.804	0.818	0.871	**0.493**	0.777
Census	**0.963**	0.945	0.500	0.962	0.966	0.947	**0.497**	0.964
Credit Card	**0.995**	0.960	N/A	0.993	1.000	1.000	N/A	1.000
CTR	0.873	0.696	N/A	**0.879**	0.712	0.682	N/A	0.659
Synthetic	0.941	0.842	0.840	**0.944**	0.933	0.838	**0.833**	0.929
Higgs	0.744	0.678	N/A	**0.746**	0.736	0.677	N/A	**0.662**
Average Score	0.876	0.847	0.682	**0.878**	0.814	0.784	**0.610**	0.790
Average $\|s - s_R\|$	-	0.045	0.195	**0.003**	-	0.038	0.222	**0.024**

This represents the trade-off they have made in order to expedite the machine unlearning process. Only MUMI performs machine unlearning on a standard random forest. Moreover, the average absolute difference score of MUMI on X_t is the closest to Retrain (see the last row 'Average $|s - s_R|$' in the table).

With the exception of unlearning 1,000 points, the performance of MUMI on X_f is the closest to that of Retrain. Meanwhile, when unlearning 10, 100, 1,000, and 0.1% ×n samples, the average scores of MUMI on X_f is lower than those of the Retrain model, indicating that MUMI truly achieves the unlearning of $\mathcal{D}_f$.

While DaRE outperforms MUMI on a few specific datasets, such as when unlearning 10 samples from the *No-Show* dataset, this discrepancy is due to the inherent poor performance of their model (as mentioned earlier, they do not employ the standard random forest model, and inject more randomness into the model). And it is crucial to reiterate that neither DaRE nor HedgeCut can truly achieve data unlearning from a trained random forest model. Only the proposed MUMI is capable of effectively removing data from a trained random forest, thereby fulfilling the requirements of machine unlearning.

4.2 Unlearning Efficiency

Let $\mathcal{T}(\cdot)$ be the time complexity of random forest, the time complexity of DaRE and HedgeCut are $\mathcal{T}(|D_r|)$, while the time complexity of MUMI is $\mathcal{T}(|D_f|)$. To evaluate the unlearning efficiency of MUMI, we report the time required for different models to unlearn 10, 100, 1000, and 0.1% ×n samples in Tables 6 and 7.

Table 6. Time (seconds) of unlearning 10 and 100 samples.

Dataset	10 (↓)				100 (↓)			
	Retrain	DaRE	HedgeCut	MUMI	Retrain	DaRE	HedgeCut	MUMI
Surgical	0.727	**0.041**	0.062	0.439	0.716	**0.270**	0.531	0.371
Vaccine	0.839	0.050	**0.040**	0.177	0.831	**0.357**	0.391	0.361
Adult	0.943	**0.033**	0.041	0.108	0.874	**0.209**	3.134	0.221
Bank Mktg.	0.879	**0.026**	0.037	0.157	0.871	**0.210**	0.383	0.253
Flight Delays	6.763	0.127	**0.046**	0.090	6.448	0.925	0.419	**0.119**
Diabetes	2.990	0.071	**0.039**	0.114	2.761	0.532	0.389	**0.150**
No-Show	1.692	**0.075**	0.367	0.097	1.696	0.437	0.427	**0.133**
Olympics	16.682	0.354	**0.044**	0.092	15.643	2.055	0.380	**0.105**
Census	11.378	0.330	0.389	**0.093**	11.478	1.045	0.402	**0.100**
Credit Card	9.451	16.753	N/A	**0.103**	9.772	23.817	N/A	**0.096**
CTR	17.409	2.009	N/A	**0.101**	16.024	4.478	N/A	**0.113**
Synthetic	33.961	0.488	**0.068**	0.099	36.580	5.563	0.561	**0.108**
Higgs	390.824	0.728	N/A	**0.126**	356.949	5.777	N/A	**0.138**
Average Time	38.041	1.622	**0.113**	0.138	35.434	3.514	0.702	**0.174**

Table 7. Time (seconds) of unlearning 1000 and 0.1%×n samples.

Dataset	1000 (↓)				0.1%×n (↓)			
	Retrain	DaRE	HedgeCut	MUMI	Retrain	DaRE	HedgeCut	MUMI
Surgical	0.724	2.131	11.940	**0.498**	0.731	0.043	**0.079**	0.265
Vaccine	0.820	3.112	3.991	**0.430**	0.830	0.090	**0.078**	0.313
Adult	0.869	1.932	9.095	**0.305**	0.905	**0.071**	1.363	0.208
Bank Mktg.	0.836	1.309	4.017	**0.316**	0.888	**0.107**	0.128	0.169
Flight Delays	6.937	9.932	4.177	**0.137**	6.027	0.796	0.314	**0.112**
Diabetes	2.790	10.360	3.673	**0.207**	2.961	0.499	0.298	**0.139**
No-Show	1.751	3.453	3.983	**0.177**	1.715	0.395	0.357	**0.137**
Olympics	17.156	18.013	3.763	**0.096**	17.745	3.079	0.609	**0.080**
Census	11.518	7.481	3.972	**0.139**	10.849	2.098	0.798	**0.102**
Credit Card	8.435	41.665	N/A	**0.114**	9.989	28.081	N/A	**0.107**
CTR	16.389	42.319	N/A	**0.133**	17.427	31.975	N/A	**0.123**
Synthetic	35.863	24.942	7.021	**0.208**	34.019	21.902	5.246	**0.212**
Higgs	383.332	52.090	N/A	**0.203**	385.495	272.148	N/A	**0.286**
Average Time	37.494	16.826	5.563	**0.228**	37.660	27.791	0.927	**0.173**

The unlearning efficiency is mainly affected by two aspects: the size of $\mathcal{D}_r$ ($|\mathcal{D}_r|$) and the size of $\mathcal{D}_f$ ($|\mathcal{D}_f|$).

Impact of $|\mathcal{D}_f|$: (i) HedgeCut is the most efficient unlearning algorithm in most datasets when there are only 10 samples that need to be forgotten in the training data. Because only a few subtrees have changed and need to be reconstructed, the subtrees that HedgeCut has prepared in advance can be directly used for replacement. However, as the number of samples that need to be forgotten increases, many subtrees in the random forest have changed and need to be reconstructed. The subtrees prepared in advance in HedgeCut are insufficient to cope with such extensive changes, resulting in a significant increase in unlearning time. (ii) When the number of samples that need to be unlearned is small, such as 10, 100 samples, and $0.1\%\times n$ samples for the four smallest datasets, the unlearning time of MUMI is higher than that of DaRE. However, when the number of unlearned samples is large, such as 1000 samples or $0.1\%\times n$ samples for the largest nine datasets, the unlearning time of MUMI is much lower than that of DaRE. Because DaRE employs the random split at the initial stages of tree construction. It only selects the optimal nodes for splitting in the final few layers of the decision tree. Consequently, when the number of samples that need to be forgotten is small, the overall structure of the decision tree constructed by DaRE remains unchanged. However, as the number of samples requiring forgetting increases, the structure of the decision trees will change, necessitating the retraining of DaRE, the complexity of retraining DaRE escalates significantly.

Impact of $|\mathcal{D}_r|$: Another obvious phenomenon presented by Table 6 and Table 7 is that the unlearning time of DaRE increases with the increase of $|\mathcal{D}_r|$, while MUMI does not. MUMI has almost the same unlearning time on these data. This is mainly due to the following two reasons:

1. Firstly, DaRE necessitates the retraining of the entire retained dataset $\mathcal{D}_r$. In contrast, MUMI does not utilize $\mathcal{D}_r$ at all during the unlearning process. Instead, it relies solely on $\mathcal{D}_f$ to construct image samples, with $|\mathcal{D}_f| \ll |\mathcal{D}_r|$. As a result, the unlearning efficiency of MUMI is significantly higher than that of DaRE, particularly when the size of the dataset $|\mathcal{D}|$ is large, where the difference becomes even more pronounced.
2. Secondly, both DaRE and HedgeCut require processing each tree within their models. If the initial random forest consists of T trees, DaRE must retrain all T decision trees on the dataset $\mathcal{D}_r$. In contrast, MUMI only needs to train k trees ($k < T$) on the image samples derived from $\mathcal{D}_f$. Consequently, the unlearning time of MUMI is significantly lower than that of DaRE, making it a more efficient approach in terms of computational overhead.

In summary, our proposed algorithm MUMI demonstrates superiority over existing algorithms in the following three key aspects:

- The most significant advantage is that MUMI can effectively unlearn the trained random forest model, whereas existing algorithms are unable to achieve this level of unlearning. This capability is crucial for scenarios where data removal is required in a trained random forest model.
- MUMI outperforms existing algorithms in terms of model effectiveness, unlearning efficacy, and unlearning efficiency. It achieves faster unlearning while maintaining model accuracy, making it more efficient overall. This balance between unlearning efficiency, model effectiveness, and unlearning efficacy is a notable improvement over current methods.

* The unlearning process of MUMI is not influenced by the remaining data $\mathcal{D}_r$ but is solely focused on the data to be unlearned $\mathcal{D}_f$. This approach is both intuitive and practical for unlearning, as it ensures that only the necessary information is removed without unnecessary interference with the remaining data. This aligns with the fundamental principle of forgetting: to remove what needs to be forgotten without affecting what remains.

5 Discussion

The pros and cons of including retrain in the unlearning algorithm: Both DaRE and HedgeCut incorporate retraining into their unlearning algorithms. For instance, DaRE retrains the layers close to the leaves of the decision trees. This approach ensures that the data to be forgotten is completely removed from the model. However, it also alters the training process of original random forest, which can lead to a reduction in model performance. Moreover, by employing retraining, the focus shifts away from the subset D_f that needs to be forgotten, and instead, the entire retain dataset D_r is retrained. This significantly increases the unlearning cost, especially when D_r is large.

Boundary conditions of the image method: Many methods use random labeling methods [7,6,31], but they do not consider boundary conditions and are therefore different from the mirror method. The application of the method of images necessitates the fulfillment of specific boundary conditions. In the context of machine unlearning, the stringent boundary condition is that the performance of the unlearned model should be the same as that of the retrained model. However, in practice, obtaining a retrained model is often infeasible due to the prohibitively high cost of retraining (which is precisely why machine unlearning algorithms are needed). In this paper, we propose using performance that closely approximates the behavior of a retrained model, performing well on X_r while performing poorly on X_f, as a proxy for the boundary condition. Identifying other superior boundary conditions to replace the need for retraining remains an open question.

The impact of different voting methods: In the experiment, we used hard voting for each tree, but since the impact of each unlearned sample on image decision trees is different when we forget different numbers of samples, the weighted voting may have better unlearning performance.

6 Limitation and Future Work

MUMI unlearns data, leading to an increase in the number of decision trees. Although only a small number of decision trees are added each time, the cumulative effect becomes significant when the number of unlearning events is large. In contrast, DaRE does not experience this issue. One potential approach is to use these cumulative unlearned samples to train image decision trees instead of the previous multiple image decision trees after forgetting lots of samples. Designing an unlearning algorithm that can achieve data unlearning in an already trained random forest without adding additional decision trees is our future work.

7 Conclusion

In this paper, we introduce the method of images into the machine unlearning task and propose the first machine unlearning algorithm based on this method, called MUMI. MUMI is capable of achieving unlearning in a trained random forest, a capability that existing algorithms lack. We provide theoretical proof that our proposed algorithm can ensure both the good performance of the model after unlearning, i.e., the removal of the specified data. Our experimental results demonstrate that MUMI outperforms existing algorithms in terms of model effectiveness, unlearning efficacy, and unlearning efficiency.

Acknowledgments. Kai Ming Ting is supported by the National Natural Science Foundation of China (Grant No. 92470116).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Baumhauer, T., Schöttle, P., Zeppelzauer, M.: Machine unlearning: Linear filtration for logit-based classifiers. Machine Learning **111**(9), 3203–3226 (2022)
2. Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C.A., Jia, H., Travers, A., Zhang, B., Lie, D., Papernot, N.: Machine unlearning. In: 2021 IEEE symposium on security and privacy (SP). pp. 141–159. IEEE (2021)
3. Breiman, L.: Random Forest. Machine learning **45**, 5–32 (2001)
4. Brophy, J., Lowd, D.: Machine unlearning for random forest. In: International Conference on Machine Learning. pp. 1092–1104. PMLR (2021)
5. Cao, Y., Yang, J.: Towards making systems forget with machine unlearning. In: 2015 IEEE symposium on security and privacy. pp. 463–480. IEEE (2015)
6. Chen, Z., Wang, J., Zhuang, J., Reddy, A.G., Silvestri, F., Huang, J., Nag, K., Kuang, K., Ning, X., Tolomei, G.: Debiasing machine unlearning with counterfactual examples. arXiv preprint arXiv:2404.15760 (2024)
7. Chundawat, V.S., Tarun, A.K., Mandal, M., Kankanhalli, M.: Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 7210–7217 (2023)
8. Devineni, S.K.: Ai in data privacy and security. International Journal of Artificial Intelligence & Machine Learning (IJAIML) **3**(01), 35–49 (2024)
9. Ertel, W.: Introduction to artificial intelligence. Springer Nature (2024)
10. Fu, S., He, F., Tao, D.: Knowledge removal in sampling-based bayesian inference. In: International Conference on Learning Representations (2021)
11. Gastwirth, J.L.: The estimation of the lorenz curve and gini index. The review of economics and statistics pp. 306–316 (1972)

12. Ginart, A., Guan, M., Valiant, G., Zou, J.Y.: Making ai forget you: Data deletion in machine learning. Advances in neural information processing systems **32** (2019)
13. Golatkar, A., Achille, A., Ravichandran, A., Polito, M., Soatto, S.: Mixed-privacy forgetting in deep networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 792–801 (2021)
14. Golatkar, A., Achille, A., Soatto, S.: Eternal sunshine of the spotless net: Selective forgetting in deep networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9304–9312 (2020)
15. Golatkar, A., Achille, A., Soatto, S.: Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16. pp. 383–398. Springer (2020)
16. Guo, C., Goldstein, T., Hannun, A., Van Der Maaten, L.: Certified data removal from machine learning models. In: International Conference on Machine Learning. pp. 3832–3842. PMLR (2020)
17. Gupta, V., Jung, C., Neel, S., Roth, A., Sharifi-Malvajerdi, S., Waites, C.: Adaptive machine unlearning. Advances in Neural Information Processing Systems **34**, 16319–16330 (2021)
18. Hanley, J.A., McNeil, B.J.: The meaning and use of the area under a receiver operating characteristic (roc) curve. Radiology **143**(1), 29–36 (1982)
19. Hoang, T., Rana, S., Gupta, S., Venkatesh, S.: Learn to unlearn for deep neural networks: Minimizing unlearning interference with gradient projection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 4819–4828 (2024)
20. Hoofnagle, C.J., Van Der Sloot, B., Borgesius, F.Z.: The european union general data protection regulation: what it is and what it means. Information & Communications Technology Law **28**(1), 65–98 (2019)
21. Izzo, Z., Smart, M.A., Chaudhuri, K., Zou, J.: Approximate data deletion from machine learning models. In: International Conference on Artificial Intelligence and Statistics. pp. 2008–2016. PMLR (2021)
22. Jackson, J.D.: Classical electrodynamics. John Wiley & Sons (1998)
23. Li, N., Zhou, C., Gao, Y., Chen, H., Zhang, Z., Kuang, B., Fu, A.: Machine unlearning: Taxonomy, metrics, applications, challenges, and prospects. IEEE Transactions on Neural Networks and Learning Systems (2025)
24. Li, Y., Wang, C.H., Cheng, G.: Online forgetting process for linear regression models. In: International Conference on Artificial Intelligence and Statistics. pp. 217–225. PMLR (2021)
25. Lin, H., Chung, J.W., Lao, Y., Zhao, W.: Machine unlearning in gradient boosting decision trees. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 1374–1383 (2023)
26. Manheim, K., Kaplan, L.: Artificial intelligence: Risks to privacy and democracy. Yale JL & Tech. **21**, 106 (2019)
27. Neel, S., Roth, A., Sharifi-Malvajerdi, S.: Descent-to-delete: Gradient-based methods for machine unlearning. In: Algorithmic Learning Theory. pp. 931–962. PMLR (2021)
28. Nguyen, Q.P., Low, B.K.H., Jaillet, P.: Variational bayesian unlearning. Advances in Neural Information Processing Systems **33**, 16025–16036 (2020)
29. Nguyen, Q.P., Oikawa, R., Divakaran, D.M., Chan, M.C., Low, B.K.H.: Markov chain monte carlo-based machine unlearning: Unlearning what needs to be forgotten. In: Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security. pp. 351–363 (2022)

30. Nguyen, Q.P., Oikawa, R., Divakaran, D.M., Chan, M.C., Low, B.K.H.: Markov chain monte carlo-based machine unlearning: Unlearning what needs to be forgotten. In: Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security. pp. 351–363 (2022)
31. Pan, Z., Andrews, E., Chang, L., Mishra, P.: Privacy-preserving debiasing using data augmentation and machine unlearning. arXiv preprint arXiv:2404.13194 (2024)
32. Pardau, S.L.: The california consumer privacy act: Towards a european-style privacy regime in the united states. J. Tech. L. & Pol'y **23**, 68 (2018)
33. Perez-Lopez, R., Ghaffari Laleh, N., Mahmood, F., Kather, J.N.: A guide to artificial intelligence for cancer researchers. Nature Reviews Cancer **24**(6), 427–441 (2024)
34. Regulation, P.: General data protection regulation. Intouch **25**, 1–5 (2018)
35. Rényi, A.: On measures of entropy and information. In: Proceedings of the fourth Berkeley symposium on mathematical statistics and probability, volume 1: contributions to the theory of statistics. vol. 4, pp. 547–562. University of California Press (1961)
36. Schelter, S., Grafberger, S., Dunning, T.: HedgeCut: Maintaining randomised trees for low-latency machine unlearning. In: Proceedings of the 2021 International Conference on Management of Data. pp. 1545–1557 (2021)
37. Sekhari, A., Acharya, J., Kamath, G., Suresh, A.T.: Remember what you want to forget: Algorithms for machine unlearning. Advances in Neural Information Processing Systems **34**, 18075–18086 (2021)
38. Song, Y.Y., Ying, L.: Decision tree methods: applications for classification and prediction. Shanghai archives of psychiatry **27**(2), 130 (2015)
39. Ullah, E., Mai, T., Rao, A., Rossi, R.A., Arora, R.: Machine unlearning via algorithmic stability. In: Conference on Learning Theory. pp. 4126–4142. PMLR (2021)
40. Varghese, C., Harrison, E.M., O'Grady, G., Topol, E.J.: Artificial intelligence in surgery. Nature medicine **30**(5), 1257–1268 (2024)
41. Wu, Z., Zhu, J., Li, Q., He, B.: Deltaboost: Gradient boosting decision trees with efficient machine unlearning. Proceedings of the ACM on Management of Data **1**(2), 1–26 (2023)
42. Zhu, M.: Recall, precision and average precision. Department of Statistics and Actuarial Science, University of Waterloo, Waterloo **2**(30), 6 (2004)