No Matter How You Slice It: Machine Unlearning with SISA Comes at the Expense of Minority Classes

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—Machine unlearning using the SISA technique promises a significant speedup in model retraining with only minor sacrifices in performance. Even greater speedups can be achieved in a distribution-aware setting, where training samples are sorted by their individual unlearning likelihood. Yet, the side effects of these techniques on model performance are still poorly understood. In this paper, we lay out the impact of SISA unlearning in settings where classes are imbalanced, as well as in settings where class membership is correlated with unlearning likelihood. We show that the performance decrease that is associated with using SISA is primarily carried by minority classes and that conventional techniques for imbalanced datasets are unable to close this gap. We demonstrate that even for a class imbalance of just 1:10, simply down-sampling the dataset to a more balanced single shard outperforms SISA while providing the same unlearning speedup. We show that when minority class membership is correlated with a higher- or lower-than-average unlearning likelihood, the accuracy of those classes is either improved or diminished in distribution-aware SISA models. This relationship makes the model sensitive to naturally occurring unlearning likelihood correlations. Our work contributes to a better understanding of the side effects and trade-offs that are associated with SISA training.**

***Index Terms*—machine unlearning, class imbalance**

I. Introduction

Under legislation such as the California Consumer Privacy Act (CCPA)[1] in California, the General Data Protection Regulation (GDPR)[2] in the European Union, or Personal Information Protection and Electronic Documents Act (PIPEDA)[3] in Canada, people have a right to request the deletion of their personal data, also referred to as *right to be forgotten*. While this right can often be honored easily by deleting all database entries relating to a specific person, erasure becomes much more difficult if the characteristics of ones personal data have already been ingrained in a trained machine learning model.

The impossibility to separate training data from trained models is exemplified by membership inference attacks [1], which allow an adversary to determine if a given data record was part of the training set of a black-box model. This inseparability motivates the need for *machine unlearning* [2], [3], which refers to any technique that is able to remove the influence that any particular training point had on the final model. The term *exact unlearning* refers to methods that formally remove the data points influence by retraining the model without it, while *approximate unlearning* refers to methods that try to approximate the model parameters that exact unlearning would yield without actually retraining the model [4].

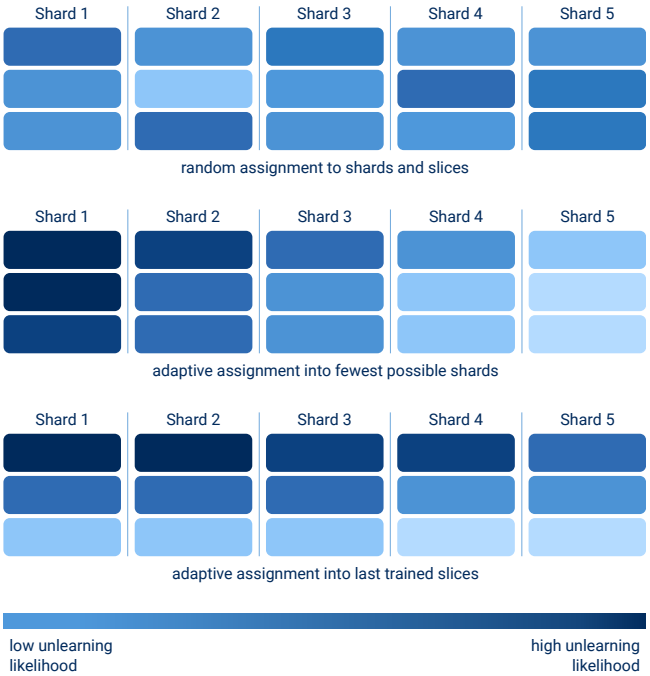

Fig. 1. Average slice unlearning likelihoods for different SISA strategies in a SISA model with 5 shards and 3 slices each. Slices in a shard are trained from bottom to top. The adaptive strategies evaluated by us place samples with a high unlearning likelihood in specific locations while keeping the shard and slice size fixed.

[1] https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375

[2] http://data.europa.eu/eli/reg/2016/679/2016-05-04

[3] https://laws-lois.justice.gc.ca/PDF/P-8.6.pdf

Retraining a model from scratch is time-, energy- and cost-intensive, especially when the model has many parameters or the training dataset is large. One exact unlearning method that tries to reduce the computational overhead associated with retraining is SISA [5], which stands for Sharded, Isolated, Sliced, and Aggregated training. SISA training divides the training data into n disjoint shards of approximately equal size. The data in each shard is then further separated into m disjoint slices. During training, one constituent model is trained per shard. In a single shard, this means training a model on the first slice and then saving the parameters of the model. The model parameters are then loaded again for each of the remaining slices, further trained, and saved again, until the training data in the last slice has been processed. These last obtained parameters define the final constituent model of that shard.

At inference time, the input is processed by each constituent model and the responses are then aggregated, for example through majority voting [5]. If any given data point has to be deleted, only the constituent model associated with the shard containing the data point has to be modified. Retraining only has to take place for the slice containing the data point as well as all following slices in the same shard, and the last saved checkpoint before those slices can be used as starting point. For a single data point, retraining is quicker if it is located in a later slice. If multiple data points have to be removed at once, it is beneficial if those data points are located in the same shard. This effectively limits the total cost of retraining to the maximum individual retraining cost, as all following slices have to be retrained anyways.

By default, data points in SISA training are assigned to shards and slices at random [5]. Even if the number of deletion requests processed at a time is relatively low, this makes it likely that all constituent models will have to be retrained. However, if the unlearning likelihood of individual data points is either known in advance or can be estimated with reasonable accuracy, the authors [5] suggest that this would allow us to group high-likelihood data points together to further improve the cost of retraining. This can be realized by placing data points with a high unlearning likelihood in designated shards, which the authors call *distribution-aware sharding*. Alternatively, and motivated by the observation made in the previous paragraph, high-likelihood data points can also be preferably placed in later slices, which we call *distribution-aware slicing*. Examples of the resulting shard and slice composition of all strategies can be seen in Fig. 1.

As the complete model parameters have to be saved $n * m$ times instead of just once, SISA effectively represents a trade-off between disk space and computing effort for retraining [5]. As disk space is considerably cheaper than GPU compute, this trade-off is beneficial. The speedup associated with using SISA ranged between $1.36\times$ and $4.63\times$ in the experiments conducted in the original publication [5] for reasonable amounts of shards and slices. At the same time, using SISA was associated with a decline in model accuracy.

The authors discussed that the speedup of retraining is mainly dependent on the number of shards, and increasing the number of slices has a quickly diminishing return beyond a few slices [5]. However, the same holds true for the accuracy degradation of the final model, which gets bigger if more shards are used. The explanation given by the authors is that each constituent model must be presented with sufficiently many data points during training in order to reach a good performance [5]. A single constituent model sees the exact same amount of training samples for any number of slices – only in a different order. However, when the number of shards is increased, the absolute amount of training samples per constituent model sinks proportionally. This impairs each constituent models ability to generalize and thus lowers the accuracy of the entire ensemble.

This observation motivates the question how SISA models behave when the number of shards is low and the absolute number of training examples per shard is high, but the number of training examples for individual underrepresented class is low. If a *magic threshold* for good constituent model generalization exists, can it be exceeded by one class, but fallen short of by another?

In summary, the contributions of this paper are:

- We show that the accuracy gap introduced by SISA training is bigger for minority classes, and gets bigger as the imbalance ratio rises.
- We demonstrate that both simple and advanced methods against class imbalance, even though successful in bringing the performance of majority and minority classes closer together, are unable to eliminate this unequal burden.
- We show that SISA is outperformed by a simple down-sampled model with 1 shard on minority classes while preserving the same retraining speedup.
- We demonstrate that minority classes are sensitive to correlations with unlearning likelihood in distribution-aware SISA settings and that a positive correlation between unlearning likelihood and minority class membership can improve the performance of said classes.

II. Related Work

The topic of *Machine Unlearning* [2], [3] or the removal of data from trained models is a recent topic in machine learning [4], [6]–[9]. One framework for unlearning is SISA training (Sharded, Isolated, Sliced, and Aggregated training) [5]. SISA is discussed in literature as a tool for data protection and trustworthy AI [10]–[12]. Besides allowing effective machine unlearning, SISA seems to protect well from attacks on privacy as the aggregation step of SISA reduces the influence of individual samples on the final prediction [13]. However, it is theoretically possible for an adversarial actor to completely degrade the accuracy of SISA [14].

Another topic related to machine learning are *imbalanced classes* [15], [16]. Classes are imbalanced if the size of at least one class (minority class) is considerably smaller than another class (majority class). As a result, the performance of the model on the minority class has a smaller impact

on the average accuracy on the dataset, which in return leads to solutions that classify minority classes inaccurately. According to [17], most existing solutions for the problem of imbalanced classes include introducing learning bias against majority classes, under- or oversampling of data, and cost-sensitive learning. Overviews of current methods to tackle class imbalance can be found in [18], [19].

To the best knowledge of the authors, no work has yet investigated the impact of class imbalance on SISA learning or machine unlearning.

III. How Does SISA Affect Imbalanced Datasets?

In many applied contexts, different classification outcomes are not equally likely. One such context where the processed data is both highly imbalanced and highly sensitive is the medical domain (for examples, see [20]–[22]). If you intend to build a classifier that is able to detect the presence of a rare disease, you will most likely have to work with many samples from healthy patients and only very few samples from sick patients. For many tasks, imbalances of 1:100 and beyond are not unheard of (e.g. [21]). At the same time, medical datasets, especially image datasets, can be very large and computationally expensive to process. For example, full-scale histopathology images can reach the size of gigapixels [23] and models working with these images take long to train even when using helpful techniques such as downsampling and tiling. If the training data for such models fall under a privacy-focused jurisdiction, using SISA seems like a good idea to ensure that retraining is both faster and cheaper. But does SISA impact imbalanced datasets differently?

A. Datasets

When SISA was introduced [5], the effects on the performance were evaluated on the MNIST [24], Purchase [25], SVHN [26], CIFAR-100 [27], Imagenet [28], and Mini-Imagenet [29] dataset. The authors assigned task complexities to each of these datasets, with the first three being regarded as easy and the latter three as hard tasks. According to the authors, sharding and slicing combined have no significant impact on accuracy for easy tasks, and result in only a small accuracy decrease in the single percentage point realm for hard tasks when pretraining is used [5].

TABLE I
Dataset characteristics

Dataset	Train size	# Classes	Max. imbalance
MNIST [24]	60,000	10	1:1
Purchase [25]	160,058	2	1:2.7
SVHN [26]	604,388	10	1:2.7
CIFAR-100 [27]	60,000	100	1:1
Imagenet [28]	1,281,167	1000	1:5.2
Mini-Imagenet [29]	60,000	100	1:1
EMNIST Digits [30]	240,000	10	1:1
Modified EMNIST Digits (ours)	170,664	10	1:1000

When comparing the ratio of the largest and smallest class in each dataset (see Tab. I), the maximal imbalance ratio for the evaluated datasets ranges between 1:1 and 1:5.2. In the original paper [5], results were reported in terms of overall average model accuracy, so even for the datasets where imbalances were present, the accuracies of individual classes were not reported separately.

In order to evaluate the impact SISA has on imbalanced classes, we are using the EMNIST [30] ("Extended MNIST") dataset, a superset of MNIST containing handwritten digits and letters. More specifically, we are utilizing the *EMNIST Digits* dataset, a balanced dataset containing 24,000 training and 4,000 testing samples per class. Having a dataset multiple times bigger than MNIST allows us to explore the limits of SISA to a larger extent, and create synthetic class imbalances up to 1:1000. Even though the SVHN dataset also has a large total size and 10 classes [26], the fact that house number digits do not occur with an equal probability gives this dataset a natural imbalance, making it more difficult to introduce specific imbalances in an experimental setting.

B. Learning with Class Imbalance

Many methods for learning with class imbalance have been proposed over the years, which can roughly be divided into data-level methods, algorithm-level methods and hybrid methods [18]. The following methods are utilized in our paper:

- **data-level methods**: random over-sampling (ROS), random under-sampling (RUS)
- **algorithm-level method**: cost-sensitive learning, focal loss, label-distribution-aware margin (LDAM) loss

All methods used in our experiments require knowledge of the global class distribution. However, when using SISA, shards and slices are isolated, and we only have local knowledge reflecting the class distribution in the current slice. Because of this, the used sample numbers and class costs in our experiments are recalculated for each slice, and will differ from slice to slice. If any individual shard has a class distribution substantially different from the overall dataset, the sample numbers and class costs will reflect that difference and differ substantially as well.

Both **ROS** and **RUS** [31], [32] work by manipulating the dataset. In ROS, random copies of minority samples are added to the dataset until all classes are of equal size. In RUS, random samples from the majority classes are removed from the dataset until all classes are of equal size. In the context of machine unlearning, ROS is undesirable, as it increases the size of the dataset and thus extends the retraining time. RUS is desirable, as it decreases the size of the training dataset and thus shortens the retraining time.

Because of this, RUS can in itself be considered a method against class imbalance that comes with improved machine unlearning as a free add-on. To achieve an equivalent speedup as a SISA model with n shards, RUS has to result in a final dataset size that is smaller than the original by a factor of $\sqrt{n}$. This relationship can be derived from the fact that a $1/n$ monolithic baseline model is always faster than a SISA model with n shards by a factor of n [5]. Because of this, we will

not combine RUS with SISA, but rather evaluate it as a third kind of machine unlearning method that can be combined with additional algorithm-level methods against class imbalance. In order for the mentioned equivalence to hold, the RUS model has to be sliced the same way as the SISA model.

Cost-sensitive learning [33] allows us to assign a different importance or *cost* to each class. Then, during training, instead of minimizing our prediction error, we are minimizing the costs of our prediction errors. While cost-sensitive learning can be used to achieve a number of goals [33], the intention for using it with imbalanced datasets is to produce models that have more similar accuracies on majority and minority classes, despite substantial differences in the number of training examples. In our experiments, we are combining the loss with class-wise costs, where the costs are equal to the square root of the inverse number of samples per class, normalized by the average number of samples per class. This means that a perfectly balanced dataset has class costs of only ones, whereas any imbalanced dataset will have class costs above one (minority classes) and below one (majority classes). Thus, any individual sample belonging to a minority class will have a larger effect on the batch loss than a sample belonging to a majority class, which counteracts the lower expected absolute number of such samples in each batch.

Focal loss [34] was originally designed for dense object detection, where a classifier was moved over many locations of an image. However, this means that many locations have no object and only a few include the target object. If all locations are used for training this generates a highly imbalanced dataset. To counter this effect, Lin et al. [34] added a modulating term to cross-entropy loss which gives well-classified samples a lower influence on the average batch loss than hard misclassified samples. The scaling factor γ in the modulating term controls how much the impact of easy samples is reduced.

LDAM loss [35] focuses on the decision boundary of the classifier and forces the model to have a larger margin between the decision boundary and minority classes than between the decision boundary and majority classes. This allows the model to achieve better generalization for minority classes while losing only minor generalization on the majority class.

Both focal loss and LDAM loss can be combined with cost-sensitive learning and have been shown to complement each other [35]. In our experiments, they are thus combined with the costs described in the section above.

C. Implementation details

To find out which effect SISA has on majority and minority classes, we establish an experimental regime that allows us to measure the impact of SISA on prediction performance depending on the imbalance of each class.

Out of the 10 EMNIST [30] digit classes, we keep 7 classes as they are, and keep samples from the 3 remaining classes with a probability of 10%, 1% and 0.1%, corresponding to resulting class imbalances of 1:10, 1:100 and 1:1000. Each experiment is repeated 10 times in a way that ensures that each digit gets assigned each of the three imbalance ratios exactly once. This ensures that classes which are easier or harder from the beginning (such as discerning the digits 3 and 8) have no effect on the reported results.

The used model architecture is a ResNet-18 [36] which was modified to process black-and-white instead of RGB images and is followed by a classifier consisting of 3 fully connected layers with dropout. We are using a learning rate of 6e-4, Adam [37] as optimizer and train each model for 5 epochs. At inference time, the post-softmax class distributions [5] of all constituent models are summed up and then evaluated. Compared to simple majority voting, this method allows for a more fine-grained aggregation of model predictions in the ensembling step. The authors of the original SISA publication [5] found that this aggregation strategy yields better performance on Imagenet and Mini-Imagenet, while not hurting the performance on the SVHN and Purchase dataset.

During evaluation, we record prediction error rates ($1 -$ accuracy) per class using the balanced test set of 40,000 samples. We do not care about the error rates for individual digits, but rather for the error rates depending on if that class was a majority or minority class. Hence, results are not reported for 10 classes, but rather grouped by class size in the training data or imbalance ratio respectively.

Given our dataset, which was chosen to be as large as possible, we chose 5 shards and 3 slices as a compromise between speedup and sample representation. If too many slices have 0 samples of a given class, we are not able to draw meaningful conclusions about the methods we are evaluating. Specifically, the likelihood of having at least one slice with 0 samples of the 1:1000 class using random allocation and having only 24 samples at hand is already 98.5%. Because of this, we deviated from the default shard and slice values from the original SISA paper [5] but rather addressed their influence in our ablation experiments (Sec. III-E).

We repeat our experiment for a monolithic baseline model and a SISA model. Both models are trained regularly as well as using all methods against class imbalance described in Sec. III-B. For the experiments that use focal loss, we have chosen $\gamma = 1.0$. RUS is evaluated on its own as well as in combination with cost-sensitive learning with regular cross-entropy loss. The resulting RUS dataset has a size of $170,664/\sqrt{5} = 76,323$ samples and includes all of the original minority class samples. It will be trained like a SISA model with 1 shard and 3 slices.

D. Results

Fig. 2 and Tab. II show the error rates grouped by model type and class imbalance. The error rate becomes higher when the class imbalance becomes more extreme. For the monolithic baseline model, mean error rates compared to the majority class are $2.7\times$ as high for an imbalance of 1:10, $14.4\times$ as high for an imbalance of 1:100, and $196.3\times$ as high for an imbalance of 1:1000.

Similar observations can be made for SISA, but the rate of deterioration is considerably higher. While the majority class

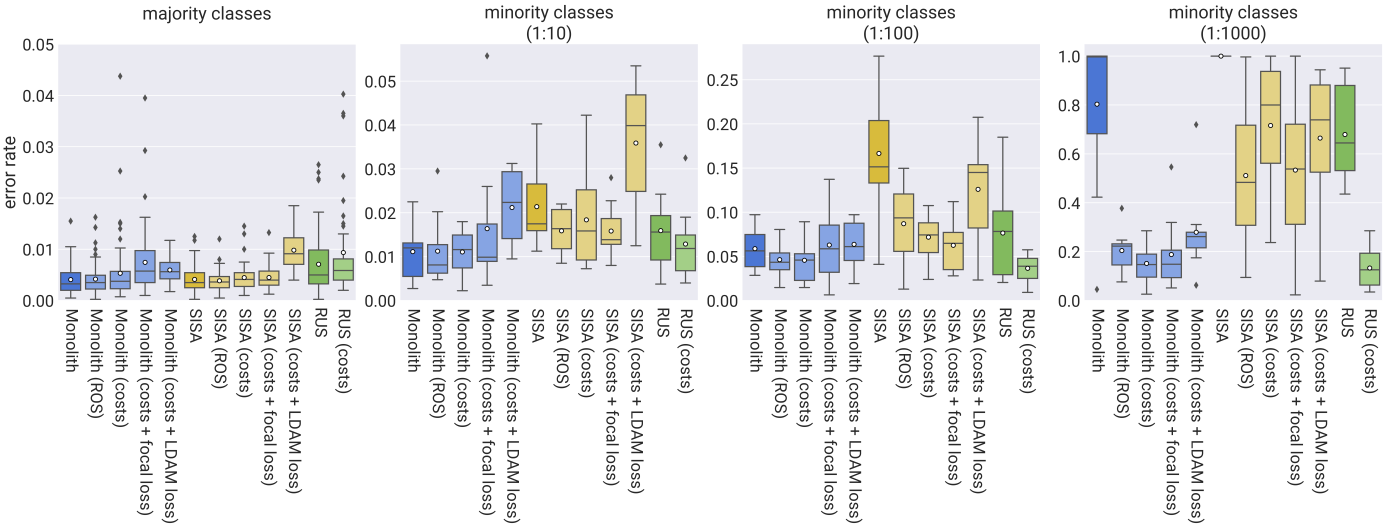

Fig. 2. Error rates for all classes with a given imbalance ratio, with annotations for the mean error rates. For the majority class group N=70, for each minority class group N=10. Whiskers at Q1−/Q3+1.5 IQR.

TABLE II
ERROR RATES AND RATES OF DETERIORATION

monolith

	majority	1:10	1:100	1:1000
regular	0.0041 (:=*mcer*)	0.0112 (2.7×*mcer*)	0.0588 (14.4×*mcer*)	0.8033 (196.3×*mcer*)
ROS	0.0042 (:=*mcer*)	0.0113 (2.7×*mcer*)	0.0463 (11.1×*mcer*)	0.2044 (48.9×*mcer*)
costs + LDAM loss	0.0150 (:=*mcer*)	0.0212 (3.6×*mcer*)	0.0637 (10.7×*mcer*)	0.2796 (46.8×*mcer*)
costs + focal loss	0.0074 (:=*mcer*)	0.0164 (2.2×*mcer*)	0.0629 (8.4×*mcer*)	0.1884 (25.3×*mcer*)
costs	0.0053 (:=*mcer*)	0.0111 (2.1×*mcer*)	0.0456 (8.6×*mcer*)	0.1519 (28.5×*mcer*)

SISA

	majority	1:10	1:100	1:1000
regular	0.0041 (:=*mcer*)	0.0214 (5.1×*mcer*)	0.1666 (40.5×*mcer*)	1.0* (242.8×*mcer*)
ROS	0.0039 (:=*mcer*)	0.0159 (4.1×*mcer*)	0.0870 (22.6×*mcer*)	0.5115 (132.7×*mcer*)
costs + LDAM loss	0.0098 (:=*mcer*)	0.0359 (3.7×*mcer*)	0.1260 (12.8×*mcer*)	0.6646 (67.7×*mcer*)
costs + focal loss	0.0045 (:=*mcer*)	0.0159 (3.5×*mcer*)	0.0626 (13.9×*mcer*)	0.5340 (119.0×*mcer*)
costs	0.0045 (:=*mcer*)	0.0184 (4.1×*mcer*)	0.0717 (16.0×*mcer*)	0.7164 (160.0×*mcer*)

RUS to $N/\sqrt{n}$

	majority	1:10	1:100	1:1000
regular	0.0071 (:=*mcer*)	0.0159 (2.2×*mcer*)	0.0765 (10.8×*mcer*)	0.6794 (95.8×*mcer*)
costs	0.0094 (:=*mcer*)	0.0129 (1.4×*mcer*)	0.0365 (3.9×*mcer*)	0.1331 (14.2×*mcer*)

mcer = majority class error rate
** = maximum reached*

performance remains practically unchanged, the mean error rates compared to the majority class are already $5.1\times$ higher for an imbalance of 1:10, $40.5\times$ higher for an imbalance of 1:100, and $242.8\times$ higher for an imbalance of 1:1000. Compared to the previous rate of deterioration, this reflects an increase of +92%, +184%, and +24.5% respectively. One should note that the performance deterioration for the classes imbalanced by 1:1000 has already reached its natural maximum (100% error rate/0% accuracy), and thus also the +24.5% change in the speed of deterioration reflects the maximum possible rate of increase. These results suggest that not only does the performance of minority classes also deteriorate when using SISA, but it does so much more quickly than if SISA were not used for training.

Almost all evaluated methods against class imbalance improve the performance of minority classes both in the monolithic as well as the SISA model. Cost-sensitive learning with LDAM loss improves the performance of the most extreme 1:1000 classes but yields higher error rates for the majority and some of the other minority classes. In the monolithic model, cost-sensitive learning with regular cross-entropy loss delivers

the best overall error rates. In the SISA model, cost-sensitive learning with focal loss has the best overall error rates.

However, two observations hold for all evaluated methods: First, that the minority class error rates become larger multiples of the majority class error rate with higher imbalance ratios, meaning that none of the methods were able to remove the effects of the class imbalance completely. Second, and more importantly, that the majority class error rate multiples in the SISA models are always larger than the corresponding values in the monolithic model. For example, while the error rate for the 1:100 class in the monolithic ROS model was 11.1 times as high as the majority class error rate, it is 22.6 times as high in the SISA model. This means that while the overall performance of minority classes improves using methods against class imbalance, the unequally distributed burden introduced by SISA remains.

RUS results in majority class error rates worse than both the monolith and the SISA model. This is not surprising, as it has fewer training examples to learn from. Combined with cost-sensitive learning with cross-entropy it outperforms all other evaluated models in the 1:100 and 1:1000 classes and only performs slightly worse than the – slower – monolith model with costs on the 1:10 classes. The pattern of an increased burden shouldered by minority classes is also reversed – in all cases, the *mcer* multiple is lower than in the corresponding monolith model. This improvement was however also paid for with an increased absolute error rate for the majority class.

E. Effect of the Number of Shards and Slices

The experiments conducted in the original SISA publication indicated that the number of shards and slices should be carefully chosen in order to ensure that the accuracy gap between monolithic and SISA model does not become too large. In a small ablation study, we evaluate how the impact on minority classes changes when the number of shards or the number of slices is increased while keeping the other fixed. We repeat the same experiment as before with 5, 10, and 20 shards and 3 slices, as well as 5 shards and 3, 6, and 12 slices. The results for varying shard numbers are shown in Fig. 3 and Tab. III, the results for varying slice numbers in Fig. 4 and Tab. IV.

TABLE III
RATES OF DETERIORATION BY NUMBER OF SHARDS

Model	1:10	1:100	1:1000
monolith	2.7×*mcer*	14.4×*mcer*	196.3×*mcer*
SISA (5 shards)	5.1×*mcer*	40.5×*mcer*	242.8×*mcer**
SISA (10 shards)	6.9×*mcer*	118.5×*mcer*	227.3×*mcer**
SISA (20 shards)	6.7×*mcer*	160.9×*mcer*	172.4×*mcer**

all SISA models have 3 slices
mcer = majority class error rate
** = maximum reached*

The results for the number of shards show a relationship between an increasing number of shards and a more pronounced performance gap between majority and minority classes. Unless the error rate ceiling is reached, the rate of

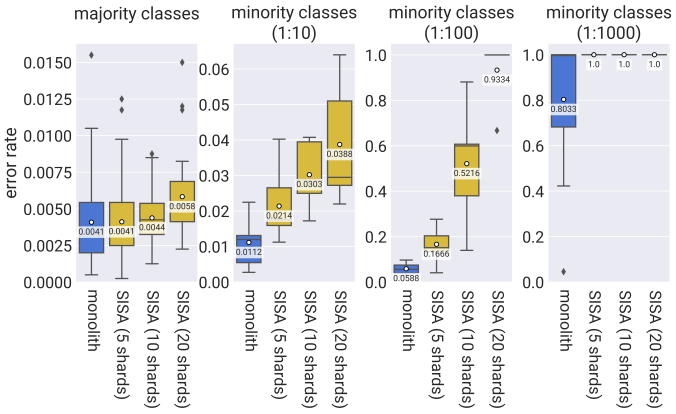

Fig. 3. Effect of the number of shards on error rates for all classes with a given imbalance ratio, with annotations for the mean error rates. All SISA models have 3 slices per shard.

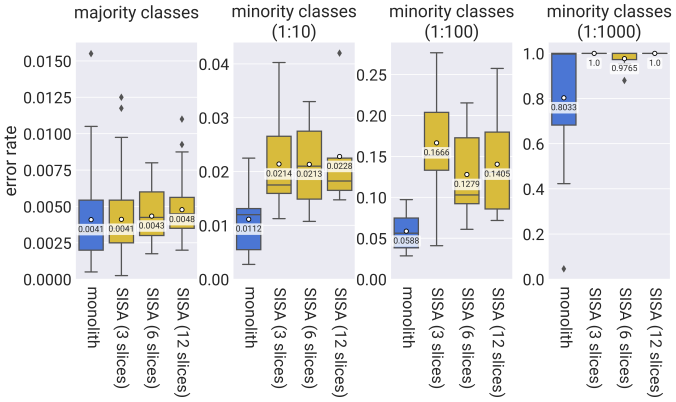

Fig. 4. Effect of the number of slices on error rates for all classes with a given imbalance ratio, with annotations for the mean error rates. All SISA models have 5 shards per slice.

deterioration increases with the number of shards, except for the SISA model with 20 shards and the imbalance ratio of 1:10, which remains almost unchanged.

There seems to be no clear relationship between the number of slices and the performance of minority classes. These findings align with the results of the original SISA paper [5], where accuracy was also mainly dependent on the number of shards, but less so on the number of slices if the model is trained for enough epochs.

TABLE IV
RATES OF DETERIORATION BY NUMBER OF SLICES

Model	1:10	1:100	1:1000
monolith	2.7×*mcer*	14.4×*mcer*	196.3×*mcer*
SISA (3 slices)	5.1×*mcer*	40.5×*mcer*	242.8×*mcer**
SISA (6 slices)	5.0×*mcer*	29.7×*mcer*	227.0×*mcer*
SISA (12 slices)	4.8×*mcer*	29.3×*mcer*	208.×*mcer**

all SISA models have 5 shards
mcer = majority class error rate
** = maximum reached*

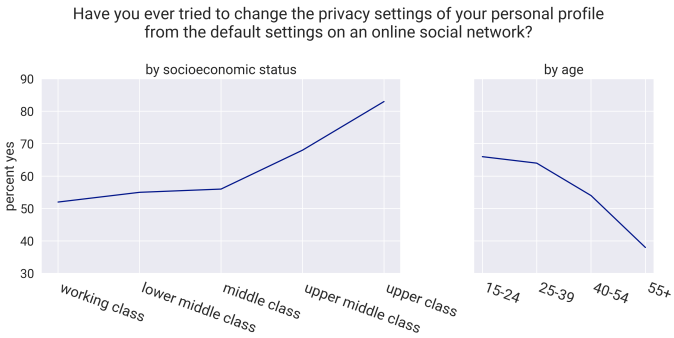

Fig. 5. Responses in the Eurobarometer survey [39] to a question regarding privacy settings. Base: online social network users in the EU (N=17,537).

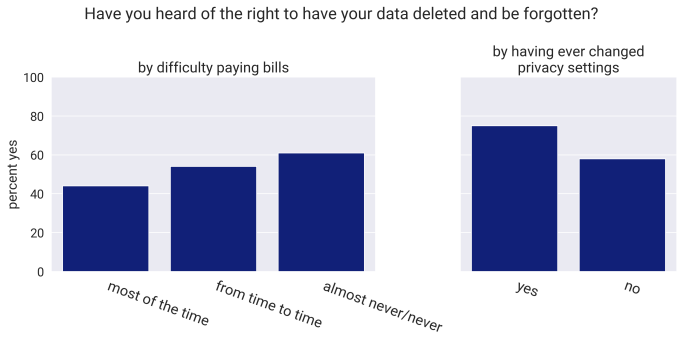

Fig. 6. Responses in the Eurobarometer survey [39] to a question regarding GDPR right awareness. Base: all respondents (N=27,524).

IV. Knowledge of Privacy Rights Is Not Evenly Distributed

As mentioned in the introduction several legislations give people the right to request the deletion of their personal data. However, who is aware of and exerting this right is not evenly distributed among the population, as we will discuss in this section.

As long as a service provider is able to estimate the likelihood of a given individual to submit a data deletion request with reasonable accuracy, sorting the training samples according to that likelihood will lead to an improvement in the average unlearning speedup by limiting the retraining to fewer shards and slices. Such estimates can be derived on a per-country level, as simulated by the original authors [5]. Their experiment was motivated by work published by Google [38], which showed differences of up to 1:10 in the number of URL deletion requests per capita for individual EU member states. But unlearning likelihood estimates can stem from any number of useful predictive features at the service provider's disposal. As many service providers have privacy settings that the user can change, the fact that the user changed those from the default settings may indicate an increased user concern about privacy. But also other general features such as age, socioeconomic status, gender, education, internet usage, the date of the last login, etc. may serve as useful predictors for the likelihood of an approaching deletion request. Large search engines and online social networks that earn money through targeted advertising have access to these features.

A. Unlearning for the Young and Rich

After the GDPR came into effect in 2018, the European Commission Directorate-General for Justice and Consumers commissioned a *Eurobarometer* survey [39] that set to explore the awareness of the newly introduced legislation in the EU, including questions on data sharing and data protection in general as well as knowledge of the newly introduced rights. Likewise, *Consumer Action* and the *Consumer Federation of America*, two US non-profits, conducted a survey [40] that evaluated the awareness of and experience with the CCPA among California residents. Both surveys found considerable dependencies of awareness and exertion of deletion rights and

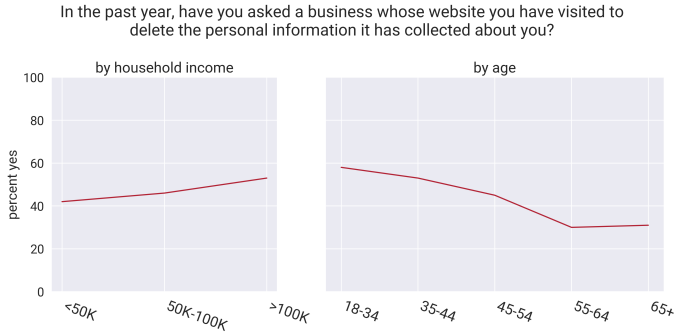

Fig. 7. Responses in the Consumer Action/Consumer Federation of America survey [40] to a question regarding submitted deletion requests. Base: all respondents (N=1,507).

socioeconomic status, age, education, previous privacy setting changes, internet usage, race, gender, and more. In the EU survey [39], the highest awareness of deletion rights recorded across all categories was among the subgroup of managers, with an awareness rate of 79%. The EU survey validates privacy setting changes as useful predictor for deletion right awareness and highlights disparities among which groups actually change those settings. Selected results from both studies are shown in Fig. 5–7.

If there are complex relationships between socioeconomic factors and unlearning likelihood, it is hard to imagine any classification scenario involving personal data in which the target variables are uncorrelated with those factors. In the medical domain being sick is positively correlated with age (and negatively with household wealth [41]), in the banking context creditworthiness is positively correlated with age and socioeconomic status, and even in the retail domain customer lifetime value shows a positive correlation with household income.

When class label and unlearning likelihood are correlated and an adaptive SISA strategy is used, this can further amplify class imbalances in shards and slices. In the following ablation study we will evaluate the effect of correlated unlearning likelihood on model performance.

B. Experimental Setup

In the original SISA paper [5], the algorithm presented for distribution-aware sharding sorts all samples by their unlearning likelihood and then places them into shards until the expected cumulative probability $\mathbb{E}(\chi_i)$ of that shard being unlearned reaches a threshold C. Then a new shard is created and filled with the remaining samples until it reaches the threshold again, which is repeated until no samples are left. This procedure accumulates high-likelihood samples into fewer shards that are smaller in size. While this does not result in a speedup for a single deletion request, as all shards are equally likely, it decreases the expected number of samples to be retrained for multiple batched requests.

In this ablation study, we are however not interested in investigating the impact of different shard *sizes*, but the impact of different shard *compositions* that have been introduced through the sorting step. We will therefore evaluate the methods for distribution-aware sharding and slicing shown in Fig. 1. In the *fewer shards* setting, samples are sorted by their unlearning likelihood, then distributed into n shards according to their likelihood, and finally sorted into m slices according to their likelihood. In the *later slices* setting, the sorted samples are first assigned their earlier or later shard number, and only afterwards sorted into shards. In essence, both strategies can be regarded as the transpose of each other.

Both methods produce shards and slices of equal size but with different expected unlearning probabilities, which in return result in the intended speedup for multiple deletion requests. The absence of unequal shard and slice sizes allows us to measure the effects of the data composition, and not the shard size or aggregation method.

For the experiments, we are modeling unlearning likelihood as a normally distributed variable with a given mean and standard deviation, where the mean for all minority classes is either one standard deviation higher or one standard deviation lower than the majority class. This results in a slightly higher or lower concentration of minority-class samples in the respective slices or shards. SISA models with 5 shards and 3 slices are then trained in the same way as in the previous experiments without any mitigation method described in Sec. III-B. Like before, we record error rates grouped by imbalance ratio, which this time also reflect the associated higher or lower average unlearning likelihood.

C. Results

The results for the strategy where samples with a high unlearning likelihood are placed into fewer shards can be seen in Fig. 8, the results for the placement in later trained slices in Fig. 9.

While the placement seems to have no large effect on the error rates of the majority class, the performance of minority classes is consistently better when belonging to a minority class is associated with a higher-than-average unlearning likelihood. In fact, the performance of minority classes is as good as when trained with cost-sensitive learning in the fewer shards setting, and outperforms the previously best cost-sensitive focal loss in the later slices setting when the unlearning likelihood is higher.

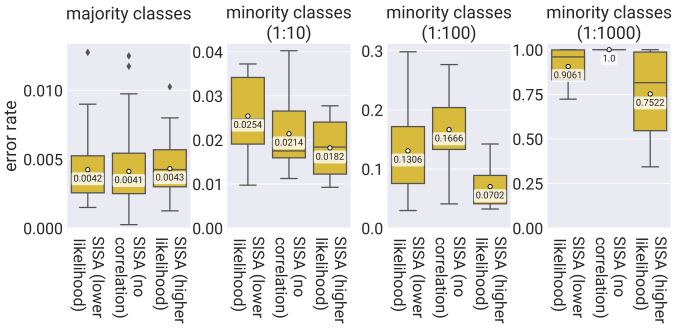

Fig. 8. Results for adaptive assignment into fewer shards. Higher/lower likelihood refers to the likelihood of the minority classes. Models have 5 shards and 3 slices.

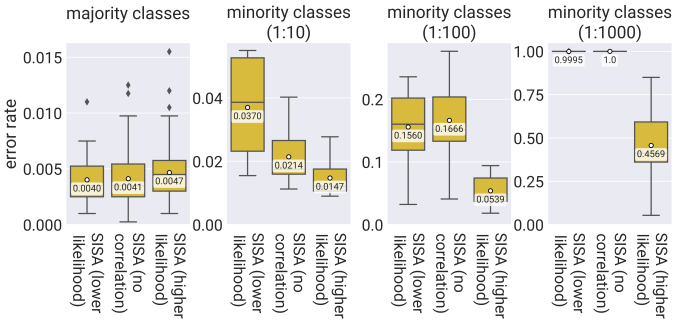

Fig. 9. Results of adaptive assignment into later slices. Higher/lower likelihood refers to the likelihood of the minority classes. Models have 5 shards and 3 slices.

In contrast, when minority classes have a lower than average unlearning likelihood their performance is sometimes but not always worse than when no correlation between minority class membership and unlearning likelihood exists. It is always worse than when the correlation is positive.

V. Discussion

The results of our first experiment show that the performance gap that comes with using SISA is larger for minority classes than for majority classes. If we assume that this generalizes to other tasks, this introduces a dilemma: Is it more important to allow efficient unlearning or is it more important to have a high accuracy across all classes?

A. The Limits of SISA

The original SISA paper [5] highlighted the importance of the absolute number of samples per shard to achieving good generalization and model performance. We interpret our results as further evidence for this finding but extend the scope to the absolute numbers of samples per shard *and class*. The inability of both data-level as well as algorithm-level methods (see Sec. III-B) to alleviate the problem of increased deterioration rates for minority classes highlights how difficult it is to overcome the increased accuracy gap that SISA introduces.

The fact that the deterioration rate in our experiments was closely related to the number of shards but not the number of slices suggests that the root cause of lower performance for SISA in general and the reason for the disadvantage for minority classes are the same. Hence, finding solutions for the problems of imbalanced classes with SISA may help improve the accuracy of SISA models in general, even when classes are balanced.

In an apples-to-apples (in terms of retraining time) comparison between SISA and the $N/\sqrt{n}$ RUS baseline, the baseline outperformed SISA on all minority classes. The superiority of the baseline becomes more pronounced as the imbalance ratio rises. This means that as long as a slight decrease in majority class performance is acceptable, the performance of minority classes can be boosted considerably by using the baseline while preserving the same average-case retraining speedup. At the same time, using the baseline reduces the worst-case retraining time by a factor of $\sqrt{n}$, and reduces the best-case retraining time to 0. The lack of an ensembling step comes with further advantages: different learning tasks such as contrastive representation learning, for which ensembling methods are less intuitive than simply summing class probabilities, are now possible too. Unfortunately, the experiments in the original SISA paper were only conducted in an apple-to-oranges way, where SISA performance was compared to a N/n baseline with a different retraining time that puts the baseline at an artificial disadvantage. We leave the question of whether there are even more scenarios in which SISA is outperformed by a $N/\sqrt{n}$ baseline to future research.

Another observation that can be made is that the variance of majority class error rates seems to be increase when using RUS. A possible reason for this could be that important prototypical training samples were randomly removed from the majority classes. If the majority classes were not downsampled randomly but strategically the average majority class performance could possibly be improved while preserving the beneficial speedup and class balancing effect.

While the performance of SISA on imbalanced datasets could probably be improved incrementally by combining more algorithm-level methods or even by coming up with more elaborate ensembling methods, the more sensible option is likely choosing the $N/\sqrt{n}$ RUS baseline instead. In fact, even a N/n RUS baseline, which is n times faster than regular SISA, compared favorably to the SISA minority class error rates when we evaluated it. The answer to whether SISA or another baseline yields better results depends therefore greatly on the dataset and is not as clear as the original SISA paper [5] likes to make it out.

We were able to show that distribution-aware SISA models are in fact sensitive to correlations between class membership and unlearning likelihood. When minority class membership is positively correlated with a high unlearning likelihood samples tend to be remembered better by the constituent models. The reason for this may be that "expert shards" are able to produce logits high enough to skew the aggregated class distribution in the correct direction, or that the placement of samples in later slices increases the ability of the model to remember samples from minority classes, which were seen more recently.

However, the fact that even in the fewer shards setting higher unlearning likelihood samples were placed in later slices inside that shard makes it more difficult to distinguish the effects of placement in fewer shards and later slices. Even though less optimal in terms of retraining time, a setting in which samples are sorted in fewer shards but then randomly placed in slices, and a setting in which samples are sorted in later slices but randomly allocated to shards could allow for better experimental isolation of both effects.

In essence, decreasing the density of minority class samples in earlier slices of the model and increasing their density in later slices is equivalent to assigning lower class costs to minority classes at the beginning of the training and assigning higher class costs at the end of the training of the constituent models. This involuntary introduction of cost-sensitivity to the model may also explain the improved performance of minority classes in this setting. Parallels can be drawn to *dynamic curriculum learning* [42], where a scheduler dynamically adapts the label distribution from imbalanced to balanced over time. Applying dynamic curriculum learning could possibly also further improve the performance of our RUS baseline.

B. A Potential Attack Vector

In the context of distribution-aware SISA unlearning we also want to highlight a potential attack vector that has so far not been mentioned in the literature. While Marchant, Rubinstein and Alfeld [43] presented an attack on approximate machine unlearning methods that strategically places poisoned samples to incur the maximum possible retraining cost if their deletion is requested, the same might be possible for SISA. If an attacker is able to produce data points that the service provider will likely consider high or low unlearning likelihood samples, for example by creating user accounts in particular countries or with an appropriate age, such data points will be placed in specific locations in a distribution-aware SISA model. While this could obviously be used to maliciously increase retraining cost, the predictable placement of data points could also allow for more effective poisoning attacks on the model.

Parallels of this attack vector can be drawn to data ordering attacks [44], where an adversary is able to skew the model towards a desired direction or prevent it from learning altogether purely by manipulating the order in which samples are presented to a machine learning model during training. Even though our adversary would need to be able to insert novel data points as well, they could also exert some control over when those data points are seen by the model. Especially when classes are imbalanced and the absolute number of minority class samples is small, changes to the correlation of class label and unlearning likelihood can also be introduced easily by an adversary, and our experiments have shown that such correlations do have an effect on model performance. As we have not conducted experiments testing the effectiveness of the mentioned attack we leave the exploration of this attack vector to future research.

C. Potential Impact on Fairness

One important observation in our experiments is that classes with higher unlearning likelihood seem to have better accuracy. As discussed in Sec. IV-A, the likelihood of unlearning is not evenly distributed: the more upper class and the younger the person, the more likely they are privacy-aware. This gives rise to the risk of unfairness being introduced in a model by applying a biased function to predict unlearning likelihood. If privileged population groups are given preferential placement in shards and slices, prediction accuracies for those groups may be higher than for others.

A reverse effect could be observed if our $N/\sqrt{n}$ baseline model is combined with an adaptive under-sampling strategy that discards samples with a high unlearning likelihood. If those samples never make it into the model, populations with a higher unlearning likelihood may have lower accuracy. Hence, the problem of unlearning likelihood and other parameters being correlated may not only be relevant for regular SISA training.

In this study, we analyzed the impact of SISA on imbalanced classes, but an impact on the subgroup fairness of a model is also conceivable. For example, models trained on datasets that contain only very few samples from a protected subgroup tend to perform worse on that subgroup. If SISA has a more detrimental impact on minority classes it could also be that it hurts the performance of minority subgroups more – independent of class membership. As we have not tested this hypothesis in our own experiments, we suggest that the impact of SISA on fairness should be investigated by future research.

A concept similar to cost-sensitive learning can be found in *fair machine learning* [45], where reweighting [46] aims to remove an existing correlation between belonging to a certain subgroup and class membership by assigning associated weights to each group. Another possible direction for future research could therefore be whether or not the efficacy of fairness-enhancing techniques such as reweighting are impacted by SISA as well.

D. The Role of the Dataset

The larger a dataset is, the larger the costs for retraining are and the higher the importance of efficient methods for machine unlearning. Unfortunately, the EMNIST [30] dataset we used for our experiments can only be considered to be medium-sized. As training an individual SISA model took less than an hour, no service provider would be willing to accept the lower performance of SISA in return for saving mere minutes of retraining time. Since large datasets might provide additional benefits for learning (a 1:1000 minority class in a 10,000,000 sample dataset still consists of 10,000 samples) the trade-off that service providers operating on a global scale face could be more favorable than suggested by our experiments.

One of the goals of this study was to analyze the impact of minority classes without the introduction of additional biases through the dataset. We were able to demonstrate the effects on minority classes using a synthetically unbalanced dataset without additional confounds. Whenever datasets with naturally occurring imbalances are used for training additional confounds may exist, and special care should be taken to determine which effect is responsible for reduced model performance.

VI. Conclusion

SISA [5] (Sharded, Isolated, Sliced, and Aggregated) training is a framework that allows efficient machine unlearning. In this paper, we analyzed the impacts of SISA on the performance of imbalanced classes. We were able to demonstrate that the difference in error rates between majority and minority classes increased with SISA, even for small imbalance ratios of 1:10. The increase in performance difference between the majority and minority classes mainly depends on the number of shards of the model. Both data-level and algorithm-level methods for learning with class imbalance improved minority class accuracy. Yet, the problem of unequal degradation rates persists when applying those methods.

When the performance of minority classes is important, simply down-sampling the dataset into a more balanced single shard of size $N/\sqrt{n}$ yields much better results than applying SISA while preserving the same average retraining speedup. We were able to show that SISA does not always win over a smaller model without ensembling and that the makeup of the dataset should play an important role when deciding on which machine unlearning method to choose.

In addition, we were able to demonstrate that SISA models are sensitive to correlations between class membership and unlearning likelihood. While this relationship can be beneficial to the performance of minority classes, the opposite can be true as well. We point out a potential attack vector that this relationship could open up to adversaries that aim to reduce the model performance.

Our work illustrates the importance of researching the side effects that are associated with machine unlearning and highlights the need for a detailed observation of model performance that goes beyond measuring average accuracy.

Acknowledgment

We would like to thank Consumer Action and the Consumer Federation of America for providing a detailed breakdown of survey responses by groups. We would further like to thank the reviewers for taking the time to review this manuscript and for their thoughtful comments and efforts toward improving our paper.

References

[1] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pp. 3–18, IEEE Computer Society, 2017.

[2] Y. Cao and J. Yang, "Towards making systems forget with machine unlearning," in *2015 IEEE Symposium on Security and Privacy*, pp. 463–480, 2015.

[3] A. Ginart, M. Guan, G. Valiant, and J. Y. Zou, "Making ai forget you: Data deletion in machine learning," in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

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

[5] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot, "Machine unlearning," in *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pp. 141–159, IEEE, 2021.

[6] Z. Ma, Y. Liu, X. Liu, J. Liu, J. Ma, and K. Ren, "Learn to forget: Machine unlearning via neuron masking," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–14, 2022.

[7] S. Neel, A. Roth, and S. Sharifi-Malvajerdi, "Descent-to-delete: Gradient-based methods for machine unlearning," in *Proceedings of the 32nd International Conference on Algorithmic Learning Theory* (V. Feldman, K. Ligett, and S. Sabato, eds.), vol. 132 of *Proceedings of Machine Learning Research*, pp. 931–962, PMLR, 2021.

[8] N. Aldaghri, H. Mahdavifar, and A. Beirami, "Coded machine unlearning," *IEEE Access*, vol. 9, pp. 88137–88150, 2021.

[9] S. Greengard, "Can ai learn to forget?," *Commun. ACM*, vol. 65, no. 4, p. 9–11, 2022.

[10] D. Kaur, S. Uslu, K. J. Rittichier, and A. Durresi, "Trustworthy artificial intelligence: A review," *ACM Computing Surveys*, vol. 55, no. 2, 2022.

[11] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Machine Intelligence*, vol. 2, p. 305–311, 2020.

[12] Q.-V. Dang, "Right to be forgotten in the age of machine learning," in *Advances in Digital Science* (T. Antipova, ed.), (Cham), pp. 403–411, Springer International Publishing, 2021.

[13] M. Chen, Z. Zhang, T. Wang, M. Backes, M. Humbert, and Y. Zhang, "When machine unlearning jeopardizes privacy," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, (New York, NY, USA), p. 896–911, Association for Computing Machinery, 2021.

[14] V. Gupta, C. Jung, S. Neel, A. Roth, S. Sharifi-Malvajerdi, and C. Waites, "Adaptive machine unlearning," in *Advances in Neural Information Processing Systems* (M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds.), vol. 34, pp. 16319–16330, Curran Associates, Inc., 2021.

[15] N. Rout, D. Mishra, and M. K. Mallick, "Handling imbalanced data: A survey," in *International Proceedings on Advances in Soft Computing, Intelligent Systems and Applications* (M. S. Reddy, K. Viswanath, and S. P. K.M., eds.), (Singapore), pp. 431–443, Springer Singapore, 2018.

[16] V. S. Spelmen and R. Porkodi, "A review on handling imbalanced data," in *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, pp. 1–11, 2018.

[17] Q. Dong, S. Gong, and X. Zhu, "Imbalanced deep learning by minority class incremental rectification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 6, pp. 1367–1381, 2019.

[18] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J. Big Data*, vol. 6, p. 27, 2019.

[19] Y. Zhang, B. Kang, B. Hooi, S. Yan, and J. Feng, "Deep long-tailed learning: A survey," 2021.

[20] M. Wang, X. Yao, and Y. Chen, "An imbalanced-data processing algorithm for the prediction of heart attack in stroke patients," *IEEE Access*, vol. 9, pp. 25394–25404, 2021.

[21] G. Cohen, M. Hilario, H. Sax, S. Hugonnet, and A. Geissbuhler, "Learning from imbalanced data in surveillance of nosocomial infection," *Artificial Intelligence in Medicine*, vol. 37, no. 1, pp. 7–18, 2006. Intelligent Data Analysis in Medicine.

[22] M. Saarela, O.-P. Ryynänen, and S. Äyrämö, "Predicting hospital associated disability from imbalanced data using supervised learning," *Artificial Intelligence in Medicine*, vol. 95, pp. 88–95, 2019.

[23] Y. Xu, Z. Jia, L.-B. Wang, Y. Ai, F. Zhang, M. Lai, and E. I.-C. Chang, "Large scale tissue histopathology image classification, segmentation, and visualization via deep convolutional activation features," *BMC Bioinformatics*, vol. 18, no. 1, p. 281, 2017.

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[25] C. O. Sakar, S. O. Polat, M. Katircioglu, and Y. Kastro, "Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks," *Neural Comput. Appl.*, vol. 31, no. 10, pp. 6893–6908, 2019.

[26] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[27] A. Krizhevsky, "Learning multiple layers of features from tiny images," tech. rep., 2009.

[28] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pp. 248–255, IEEE Computer Society, 2009.

[29] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain* (D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, eds.), pp. 3630–3638, 2016.

[30] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: an extension of MNIST to handwritten letters," *CoRR*, vol. abs/1702.05373, 2017.

[31] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.

[32] N. Japkowicz, "The class imbalance problem: Significance and strategies," in *Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000)*, 2000.

[33] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Cost-Sensitive Learning*, pp. 63–78. Cham: Springer International Publishing, 2018.

[34] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[35] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778, IEEE Computer Society, 2016.

[37] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv e-prints*, p. arXiv:1412.6980, Dec. 2014.

[38] T. Bertram, E. Bursztein, S. Caro, H. Chao, R. C. Feman, P. Fleischer, A. Gustafsson, J. Hemerly, C. Hibbert, L. Invernizzi, L. K. Donnelly, J. Ketover, J. Laefer, P. Nicholas, Y. Niu, H. Obhi, D. Price, A. Strait, K. Thomas, and A. Verney, "Five years of the right to be forgotten," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019* (L. Cavallaro, J. Kinder, X. Wang, and J. Katz, eds.), pp. 959–972, ACM, 2019.

[39] *Special Eurobarometer 487a – March 2019 – "The General Data Protection Regulation"*. European Commission, 2019.

[40] *Survey Report: Too Many Californians Are Still Unaware of Privacy Rights*. Consumer Action and Consumer Federation of America, 2022.

[41] N. Habibov, A. Auchynnikava, and R. Luo, "Poverty does make us sick," *Annals of Global Health*, vol. 85, no. 1, 2019.

[42] Y. Wang, W. Gan, J. Yang, W. Wu, and J. Yan, "Dynamic curriculum learning for imbalanced data classification," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 5016–5025, IEEE, 2019.

[43] N. G. Marchant, B. I. P. Rubinstein, and S. Alfeld, "Hard to forget: Poisoning attacks on certified machine unlearning," in *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 7691–7700, AAAI Press, 2022.

[44] I. Shumailov, Z. Shumaylov, D. Kazhdan, Y. Zhao, N. Papernot, M. A. Erdogdu, and R. J. Anderson, "Manipulating SGD with data ordering attacks," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual* (M. Ranzato, A. Beygelz-

imer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, eds.), pp. 18021–18032, 2021.

[45] D. Pessach and E. Shmueli, "A review on fairness in machine learning," *ACM Comput. Surv.*, vol. 55, no. 3, 2022.

[46] F. Kamiran and T. Calders, "Data preprocessing techniques for classification without discrimination," *Knowl. Inf. Syst.*, vol. 33, no. 1, pp. 1–33, 2011.