The Vision of Machine Unlearning

Mansi Ranka (mranka@iu.edu), Mohit Sharma (moshar@iu.edu), Tanmayi Balla (tballa@iu.edu), Harika Kanakam (hkanakam@iu.edu)

***Abstract*—Machine unlearning has come to light in recent times with a focus on solving user privacy regulations. Researchers are experimenting with a lot of techniques to effectively remove the impact of the user's information on the trained models based on their requests. Though retraining the model from scratch for the remaining dataset would be the gold standard, it would incur a huge cost in terms of computational capabilities and the runtime in a real-time setting. A lot of unlearning techniques have been evolved by researchers to achieve not just high unlearning performance but also retain the model accuracy. This research work is an attempt to contribute to the field of machine unlearning by implementing different techniques like the forget-remember cycle, self-contrastive adversarial learning, and stochastic re-initialization. We evaluated our models by injecting Membership Inference Attacks into the unlearned model and reported the scores of the same as forget quality. Our top-performing models have achieved a forget quality of 0.74 on the CIFAR10 dataset as compared to the baseline model's forget quality which sticks around 0.63. We have conducted extensive experimentation and troubleshooting to come up with different models that achieved decent forget quality scores, all of which have been reported in the later sections of the paper.**

***Keywords*—Machine Unlearning, Knowledge Distillation, Class Unlearning, Item Unlearning, KL-Divergence.**

Source code - https://github.iu.edu/mranka/DLS_Final_Project

I. INTRODUCTION

The idea of Machine Unlearning originated from [1] where the authors presented the technique as a solution for data lineage where data patterns build up on the existing models and the user information gets embedded in these models. Recent studies have shown the rise of malicious attacks [2-6] such as Membership Inference Attacks (MIA) that can infer if a data sample was used to train the target model, thereby identifying if a particular medical record was used to train a model leading to compromising the user's medical history. Apart from user privacy, malicious data ingestion into the existing models can compromise the model's accuracy by deteriorating its performance. The same issue arises with recommendations in social media, where a user might want to remove recommendations of a certain kind, even though they signed up for it previously. All these concerns have paved the way for machine unlearning where the model tries to erase information i.e., the user's data or the ingested noise that is required to be forgotten.

Technically, considering the samples to be forgotten as the forget set, an unlearning model tries to modify the feature space such that the performance of the model on the forget set is similar to its performance on the test set. While our research is mainly focused on item unlearning, we have also implemented a few solutions for class unlearning to test the efficiency of the same. With this said the main goal of our research is to explore and evaluate various techniques that effectively erase the information of the forget set from the model without compromising on the accuracy of the retained dataset. A simple approach to achieve this would be the SISA framework in [7], where different models are trained for each class and the model of the forget class is erased in the predictions. Though this approach guarantees complete knowledge removal of the forgotten set, it is computationally expensive as the model classes vary dynamically.

Authors in [8] have proposed the first unlearning algorithm in Bayesian inference, where they have formulated the unlearning problem into an optimization problem and minimized the KL-Divergence (KLD) between the datasets. Zero Shot Unlearning [9] has also been implemented that train the unlearning algorithm in a data-free setting by converting the forget set to noise using GANs. However, these algorithms are designed in such a way that they completely erase the forget set by modifying the samples instead of changing the feature space of the model, which doesn't align with the goals of this project. Hence we implemented a knowledge distillation approach, similar to [10-11] where the authors introduce a student-teacher framework such that the student model tries to match its distribution with the teacher model. Our implementation is based on [11] but unlike them, we have introduced a new "unlearn loss" that gives better accuracy as compared to the original loss in the paper.

Though this type of unlearning can be extended to applications like category-based user recommendations [12], we came up with more sensitive applications where class unlearning is not a viable option. The insurance industry assess various personal information including health records before moving ahead with their client. These insurance firms partner with health-based companies to track the medical records of the user and this raises a privacy concern from the user's end if the healthcare company has their data being used for medical predictive models and the insurance company might end up being biased towards their client.

Item unlearning comes into the picture here, where we only remove the effect of data samples from the forget set on the originally trained model, instead of an entire class. Item unlearning can be solved using different techniques like resetting/modifying gradients[13-15, 22], self-supervised contrastive learning, model pruning[16-18, 37-38, 40], gradient re-initialization[19-22, 35-36], and many more. We have implemented similar approaches and have observed decent accuracy scores. While the methodology of our implementation has been presented in Section II, the top-performing approaches have been discussed in Section III, followed by the results and conclusion.

II. METHODOLOGY

TABLE I. Symbols Dictionary

Symbols	Definitions
D_o	Full dataset
D_f	Forget set
D_r	Retain set $(D_r = D_o - D_f)$
M_o	Original model trained on D_o
M_u	Unlearned model trained on forget D_f
M_{rt}	Retrained model trained on D_r
FQ	Forget Quality

This section provides a general overview of the research in this paper. Table. I. includes some common symbols and their definitions which are used further in this paper. As mentioned in Section I, we have focused our attention on class and item unlearning.

For Class Unlearning, a convolution-based model has been created and trained on the CIFAR10 dataset as the original model (M_o). The unlearning model (M_u) uses a knowledge-distillation approach that tunes the parameters of M_u such that its performance matches with the performance of a randomly initialized stochastic

model on the forget dataset (D_f) and the M_o on the retain dataset (D_r). A detailed explanation of the same is provided in Section III.

The workflow of Item Unlearning starts by loading a model with pre-trained ResNet18 [23, 39] weights on the CIFAR10 dataset provided by [24, 39] as M_o. This model is then fed to the unlearning block and the resultant network along with D_f [25, 39], D_r [26, 39], and M_{rt}, provided by [27, 39] are used to evaluate the forget quality (FQ)[28-30] of the model by performing MIA. A more detailed explanation of the evaluation metrics is mentioned in the later part of this section. While the unlearning block varies for different algorithms tested, the M_o, M_{rt}, D_f, and D_r remain constant throughout the experimentation, ensuring a fair comparison between the models. Fig.1. depicts a quick flow of the unlearning models. Each model undergoes a forget stage and a remember stage where the forget stage helps the model forget about D_f, and the remember stage helps the model to relearn the patterns of D_r.

Techniques such as model weights pruning, weights resetting, removal of matching gradients, self-supervised contrastive unlearning, and uniform KL-Divergence (KLD) have been tested for item unlearning. The best-performing models i.e., weight resetting, self-supervised contrastive unlearning, and re-initializing matching gradients have been explained in detail in section III.

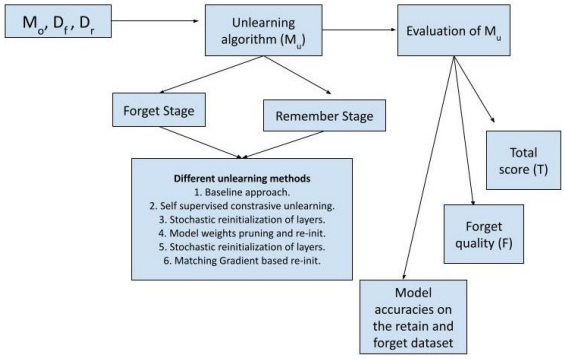

Fig. 1. Machine Unlearning Workflow

The **Forget Quality (FQ)** assesses the similarity between the distributions of M_{rt} and M_u. It should be noted that since the M_{rt} is the model that has been trained from scratch without the forget set, it serves as a benchmark model for the unlearning algorithms. The FQ is directly proportional to the similarity between the M_{rt} and M_u. Hence, a higher FQ implies that the distribution of M_u is very close to the distribution of M_{rt} - the main aim of all unlearning algorithms. An ideal unlearning algorithm will have an FQ of 1.0.

Computing FQ [28] involves performing a logistic regression-based black-box attack, also known as MIA to differentiate the distributions of M_{rt} and M_u without any information about the model weights. An ideal unlearning algorithm makes it challenging for the MIA model to distinguish between these distributions, thereby spiking up the False Positive Rate (FPR) and the False Negative Rate (FNR). (1) computes the value of epsilon which is later used to compute the FQ, thereby quantifying the unlearning algorithm's effectiveness.

$$\varepsilon = max(\frac{log(1-\delta-FPR)}{FNR}, \frac{log(1-\delta-FNR)}{FPR}) \quad (1)$$

δ in (1) is a hyperparameter set to 0.01 for all our unlearning algorithms. A higher ε implies a bad unlearning model. A scoring function H assigns the number of "points" to each example, based on the respective sample's ε, followed by aggregating the average per-example scores.

$H(s) = 2/2^{n(s)}$, where n is a function that takes in a forget example s and maps it to a "bin index" (an integer in the range [1, B], where B is the total number of bins), based on its ε. Bins with smaller indices are better bins, to which better (smaller) ε values get assigned.

$$H(s) = \begin{cases} 1 & 0.0 \leq \varepsilon^s < 0.5 \\ 0.5 & 0.5 \leq \varepsilon^s < 1.0 \\ \dots \end{cases}$$

Finally, the average of the H(s) function mentioned above, leads us to the FQ. An efficient unlearning algorithm should have a higher FQ. The **total score metric** is a combination of FQ, model utility and efficiency where the model utility (TA_ratio) is defined as the ratio of the accuracies of M_u and M_{rt} on the test set and the model efficiency (RA_ratio) is defined as the ratio of the accuracies of M_u and M_{rt} on D_r. Hence, total score = FQ*TA_ratio*RA_ratio. The total score for a perfect unlearning algorithm is 1.0.

III. Top Performing Models

The **Teacher-Student Class Unlearning** is based on the Knowledge Distillation approach where we try to match the distribution of the student and teacher network using a loss function (mostly KLD). The base idea of this technique has been extracted from [31-34]. However, unlike the method in [31], our implementation has extended the loss function beyond KLD.

In the **forget stage**, we try to match the distribution of M_u with a random distribution on D_f. To achieve this, a randomly initialized network (M_s) takes the role of a teacher, while the student network is a clone of the original model (M_o), initialized by the model parameters and weights. The forward pass is performed over D_f and the student network tries to match the distribution of M_s using an unlearn loss (U_l) represented in (2). The unlearn loss has been set in such a way that, along with matching the distribution of the models, the loss also penalizes the difference between entropies of the model distribution. This entropy difference essentially contributes to the randomness of the unlearned model predictions on D_f.

Remember phase is essentially a fine-tuning step where the original model takes the role of the teacher and the unlearned model as a student tries to adjust the model distribution for the D_r. This step makes sure that the model regains its accuracy on the retained dataset (D_r).

$$\begin{aligned} U_l = \alpha * KLDivergence(student, teacher) \\ +(1-\alpha) * MSE(Entropy(teacher) - Entropy(student)) \end{aligned} \quad (2)$$

Gradient Matching item unlearning is an approach where we try to minimize the effect of D_f distribution on the model by resetting and re-initializing the relevant layer weights. The main aim of the forget stage of this approach is to prune and re-initialize the weights of the model that have matching gradients for the D_f and D_r. The intuition behind the same boils down to the fact that these similar gradients pose challenges to the model while fine-tuning on D_r by retaining the distribution of D_f, thereby making it difficult for the model to erase information of D_f.

To find similar gradients, the gradients of the network are first updated using gradient descent on the cross entropy loss (CE_l) of D_r and gradient ascent on the negative CE_l of D_f. This step, represented by Step-1 below, essentially tries to set the gradients by minimizing the loss of D_r and maximizing the loss of D_f, resulting in low values of the weights having a gradient match between D_r and D_f. The least k gradients are re-initialized using He-initializer, where k is a hyper-parameter that can be tuned during training. The re-initialization is represented in Step-2 below.

1. *forward_pass(W, D_r): Gradient update: CE_i forward_pass(W, D_f: Gradient update: -1.0 * CE_i*
2. *For All $L_{l:0,l\in conv2d}$ He initializer (W_l [least k gradients])*

Step-1 and 2 are followed by a **remember stage** where we fine-tune our model on D_r.

Self supervised contrastive adversarial learning The **forget stage** is divided into two parts. Initially, the KLD between the output logits of D_f and uniform pseudo labels is maximized in order to erase the logits space of the model pertaining to D_f. This is followed by self-supervised contrastive loss which aims to push the distance between the positive and the negative classes where the positive class are the samples from D_r and negative class are the samples from D_f.

$$l_i = -\frac{1}{batchsize_2} \sum_{t=0}^{batch_2} log(\frac{e^{sim(x_i,y_t)/\tau}}{\sum_{j=0}^{batch_2} e^{sim(x_i,y_t)/\tau}}) \quad (3)$$

$$L_{forget} = \frac{1}{batchsize_1} \sum_{i=0}^{batch_1} l_i \quad (4)$$

(3) and (4) explain the self-supervised contrastive loss, where y_j is the sample from the positive class and xi is the sample from the negative class. The loss is scaled by a factor of $1/batchsize_2$ since each sample has an equal probability of being picked from the positive class. $batchsize_1$ corresponds to the total samples in the D_f while $backsize_2$ corresponds to the total samples in D_r.

Remember stage is used to fine tune our model on D_r.

Resetting weights of the original model and fine-tuning the same for D_r, have effectively improved the FQ mentioned in section II. The **forget stage** of this algorithm involves selectively resetting k layers in M_o and fine-tuning it on D_r. This disrupts connectivity, allowing the model to forget the distribution of D_f while preserving the learned features in the unaltered layers. The connectivity between the layer weights has been reconstructed in the **remember stage** by fine-tuning the model on D_r. This step not just enables the model to retain valuable data patterns, but also helps the model to adapt to a new distribution without the forget set. This approach offers a dynamic and efficient training strategy. Setting the value of k to 3 has given us the best forget quality of 0.74.

The results of item unlearning approaches have been presented in Table.VI of Section IV. All the other experimentations for these models have been discussed in the experimental analysis section.

IV. Results & Experimentation

Dataset: While our primary focus of experiments was towards the CIFAR_10 dataset, we have also evaluated a few top performing unlearning techniques for the Alzheimer's disease classification in the end of this section as a practical scenario. This choice reflects the real-world concern that some patients may opt-out of contributing their MRI data due to fears of insurance bias even though there are strong rules and regulations in place.

A. Evaluation Metrics

Class Unlearning: The mean difference between the accuracy of M_o and M_u is computed by taking a mean over the respective accuracies for each of the forget class ($F_c \in [0,10)$). This mean difference is computed for both D_f and D_r

A higher value of A_d^f implies that M_u is successful in forgetting the D_f, whereas a lower value of A_d^r implies that M_u retains its performance on D_r. Hence, for an ideal M_u,$A^f_{diff\_ou} = 1.0$ and $A^r_{diff\_ou} = 0.0$.

Similarly, mean differences above in point a are computed using M_u, and a fine-tuned model (M_{ft}), which is essentially the unlearning model without the forget stage. The values of $A^f_{diff\_ftu}$ and $A^r_{diff\_ftu}$ evaluate the significance of the forget stage of M_u. In an ideal setting, for an efficient M_u, $A^f_{diff\_ftu} >>> 0.0$ *and* $A^r_{diff\_ftu} \sim 0.0$

TABLE II. Symbols Dictionary

Metric Symbol	Metric Definition
A_o^f	accuracy of the M_o for D_f
A_o^r	accuracy of the M_o for D_r
$A_u o^f$	accuracy of the $M_u o$ for D_f
$A_u o^f$	accuracy of the $M_u o$ for D_r
$A^f_{diff\_ou}$	mean difference of accuracy of the M_o and M_u for D_f
$A^r_{diff\_ou}$	mean difference of accuracy of the M_o and M_u for D_r
M_{ft}	Model fine-tuned on D_r using M_o
$A^f_{diff\_ftu}$	mean difference of accuracy of the M_{ft} and M_u for D_f

TABLE III. Unlearning model metric symbol table

Metric Symbol	Metric Definition
A^f_{rt}	accuracy of the retrained model on the forget dataset
A^f_u	accuracy of the unlearned model on the forget dataset
A^r_{rt}	accuracy of the retrained model on the retain dataset
A^f_{rt}	accuracy of the unlearned model on the retain dataset

Item Unlearning: FQ, Total Score, A^f_{rt}, A^f_u, A^r_{rt}, A^f_u, where A_{rt} is the accuracy of the re-trained model on D_r. For an ideal unlearning algorithm, the FQ = 1.0, Total Score = 1.0, $A^f_{rt} = A^f_u$ and $A^t_{rt} = A^r_u$.

B. Benchmark

TABLE IV. Benchmark results of Class Unlearning

Approach	MO_df	MO_dr	MR_df	MR_dr
Teacher-Student (Using naïve KLD)	0.59	0.009	0.0066	0.16
Teacher-Student (Using our unlearn loss)	0.642	0.0006	0.011	0.23

Table IV. shows the results of the teacher-student model for class unlearning. The mean difference between the unlearned model and the original model for the forget set is high (62%) implying that the unlearning model is successful in forgetting D_f. Similarly, the same difference for the retain accuracy is very low (0.6%) which confirms that the model is still on par with the original model in terms of overall retain accuracy. A comparison between the KLD loss and our unlearn loss has been mentioned in the table. Our unlearning loss has boosted the mean difference on the forget set from 59% to 64.2% which suggests that matching the randomness of the models would also help M_u to align with its teacher network.

Table V. shows the results of the top-performing item unlearning approaches described in section III. Model 1 has performed well in the FQ for a value of k=3. This performance can be justified because of the fact that resetting the weights of k entire layers would create

TABLE V. Benchmark results of Item Unlearning

Approach	Forget Quality	Total Score	Retained Accuracy		Forget Accuracy	
			M_{rt}	M_u	M_{rt}	M_u
Baseline Model	0.6124	0.6524	0.995	0.998	0.882	0.823
(Model 1) Resseting Weights of k layers	0.747	0.67	0.995	0.9555	0.882	0.832
(Model 3) Gradient based model pruning and re-init	0.719	0.703	0.999	1.0	0.88	0.92
(Model 2) SCL	0.698	0.666	0.995	0.987	0.882	0.845

a disconnectivity for the model and this disconnectivity is restored with the patterns of D_r while fine-tuning. Hence, the model is able to forget D_f effectively. Though the FQ of Model 2 is less as compared to Model 1, it was able to achieve a better value for the Total Score suggesting higher similarity in the distributions of M_u and M_{rt} for D_r and D_t. This re-iterates to the fact that resetting matching gradients of D_r and D_f is a better strategy to obtain similar distributions, as compared to resetting an entire layer (Model 1).

C. Distribution Comparisons for Top performing item unlearning models

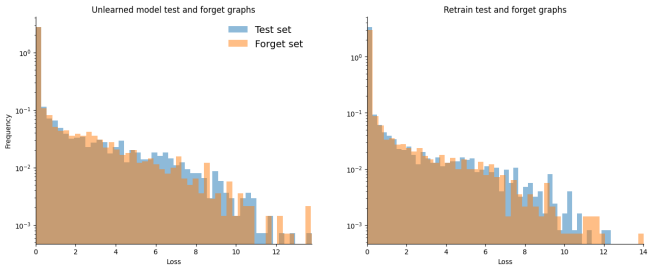

Fig. 2. Model 1 (a) Loss distribution M_u for the test dataset and D_r. (b) Loss distribution M_{rt} for the test dataset and D_f.

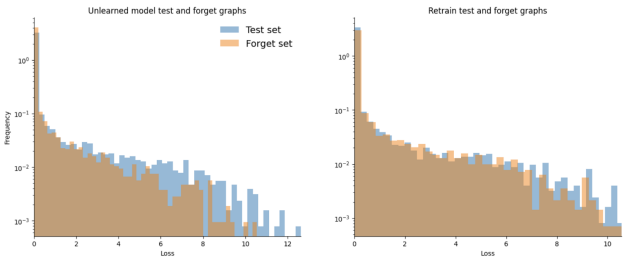

Fig. 3. Model 2 (a) Loss distribution M_u for the test dataset and D_r. (b) Loss distribution M_{rt} for the test dataset and D_f.

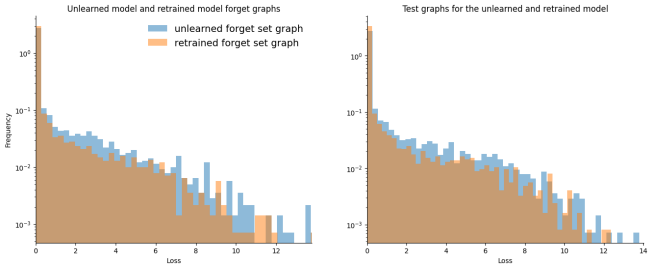

Fig. 4. Model 1 (a) Loss distribution M_u and M_{rt} for D_f. (b) Loss distribution M_u M_{rt} for the test dataset.

D. Results Description

The main aim of our unlearning algorithm was to have similar distributions of M_u and M_{rt}. We are evaluating this by comparing the loss distribution of M_u and M_{rt} on the D_r and test dataset. We are also observing the FQ, total score, and the accuracy of the M_u on the D_r and D_f to evaluate the quality of M_u. As can be seen, by the FQ for the three models in Table V all the proposed models(model 1, model 2, model 3) have better FQ than the baseline model indicating that the additional efforts we took in the forget steps were in the right direction. A comparison between Fig. 2. and Fig. 3. shows that model 1 has a higher degree of

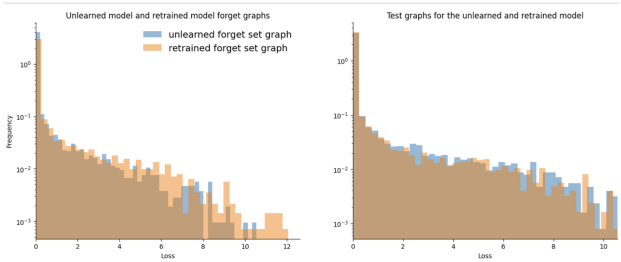

Fig. 5. Model 2 (a) Loss distribution M_u and M_{rt} for D_f. (b) Loss distribution M_u M_{rt} for the test dataset.

similarity between the loss distribution of M_u on the D_f and test dataset as compared to model 2. This can also be validated by the fact that model 1 has better FQ than model 2. Fig.4. and Fig.5. shows that model 2 has higher degrees of similarity between the loss distribution of M_u and M_{rt} when evaluated on the D_f and test dataset respectively. This is validated by the fact that we have a better total score for model 2 than model 1. For all the top 3 models i.e., model 1, model 2, and model 3 we can observe that our M_u accuracy on the D_r is comparable to the M_{rt} accuracy on the D_r and the M_u accuracy on the forget dataset is comparable to the M_{rt} accuracy on the forget dataset thereby meeting the project goals described in section I.

E. Experiential Analysis

TABLE VI. Item unlearning models performance score

	F-score	Total Score
Baseline	0.661	0.621
SCL	0.698	0.666
Weight Resetting	**0.747**	**0.668**
Model Pruning	0.63	0.56
Gradient based model pruning and re-init.	**0.719**	**0.703**
Uniform KLD	0.662	0.639

Baseline fine-tuning - we fine tuned our original model on the retain dataset. We have set this model as the base for all item unlearning algorithms and aimed to surpass the baseline in all approaches. As presented in Table VI. The baseline forget quality is around 0.66 and Total Score is around 0.62. Most of the unlearning algorithms we experimented with have surpassed the baseline. However, few didn't and we have reported the logical conclusion for the same in this section.

Uniform KL Divergence In the **Forget stage**, we aimed to maximize KL divergence between the output logits of D_f and a uniform pseudo label. This encourages the model to forget D_f by ensuring it predicts classes randomly for these instances, while in the **Remembering Stage**, we simply fine-tune tune to model on retain dataset. The results from Table VI., clearly show that the total score was better than the baseline model and the accuracies (on D_f and D_r) of M_u (0.99, 0.84) and M_{rt} (0.99, 0.87) are comparable and are both better than the baseline model. This motivated us to add the contrastive learning loss to our M_u which became one of our top-performing algorithms.

Model Pruning In this technique, we prune some percentage of model weights using L1 Unstructured Criteria that target the vectors with the lowest L1 norm, thereby introducing sparsity to the model. These pruned weights are randomly initialized before the fine-tuning step to introduce noise to the model resulting in forgetting D_f. In the fine-tuning step, rather than just using the Cross-Entropy loss, we created a custom loss function that introduces an entropy regularization parameter. This loss function is depicted in (5).

$$\begin{aligned} L_{model} &= L_{CE} + L_{EntropyReg} \\ L_{EntropyReg} &= MSE(Entropy_{M_o} - Entropy_{M_{ft}}) \end{aligned} \tag{5}$$

This model has not even achieved the accuracy of the baseline model. Due to the sparsity introduced in the model, the unlearn retain accuracy A_u^r has gone down as compared to the other approaches. If we change the pruning strategy to random, then this model performs similarly to the resetting weights model.

Real-world application: The Alzheimer's disease prediction [41] dataset contains brain MRI scans and has 4 classes corresponding to the type of Alzheimer's disease. We initially trained a ResNet18 model on this dataset. However, due to a high imbalance in classes, we were not able to achieve test accuracy beyond 62%. Data augmentation techniques like image cropping, rotation, flipping, and smudging didn't help much to improve the accuracy. Hence, we went ahead and evaluated the baseline and SCL unlearning algorithms on a random forget set. Though the baseline model gave decent unlearning accuracies on the retain and the forget datasets, i.e., approx. 75%, the FQ and the total score values went down significantly to 26%. However, this is just a starting point, and there are many better augmentation techniques, and loss functions that could be tested on this dataset to achieve better results.

V. Conclusion & Future Works

In this paper, we presented different algorithms for reducing the influence of data on a trained model at a class level and an item level. Our experiments have concluded that one of the best ways of unlearning is to create a discontinuity in the model. Metrics like Forget Quality and the total score are considered as the best metrics to evaluate unlearning models. In conclusion, we provide and evaluate different item unlearning approaches in order to mitigate the privacy concerns of the user's and handle MIA attacks and security data breaches for the existing predictive models.

As future works, these algorithms can be extended to real-world datasets such as the Breast Cancer Wisconsin dataset, Alzheimer's disease predictions, etc. which contains highly sensitive user data, and unlearning user samples is a mandate on their request. On the algorithms front, more focus can be given towards improving the self-supervised contrastive learning model, and the gradient-based model pruning and re-initialization. With the increase in privacy restrictions, the field of machine unlearning is on demand for innovative techniques to push the current state-of-the-art.

References

[1] Y. Cao and J. Yang, "Towards Making Systems Forget with Machine Unlearning," 2015 IEEE Symposium on Security and Privacy, San Jose, CA, USA, 2015, pp. 463-480, doi: 10.1109/SP.2015.35.

[2] Hongsheng Hu and Zoran Salcic and Lichao Sun and Gillian Dobbie and Philip S. Yu and Xuyun Zhang, "Membership Inference Attacks on Machine Learning: A Survey", doi: https://doi.org/10.48550/arXiv.2103.07853

[3] Thanh Tam Nguyen and Thanh Trung Huynh and Phi Le Nguyen and Alan Wee-Chung Liew and Hongzhi Yin and Quoc Viet Hung Nguyen, "A Survey of Machine Unlearning", doi: https://doi.org/10.48550/arXiv.2209.02299

[4] Ahmed Salem and Yang Zhang and Mathias Humbert and Pascal Berrang and Mario Fritz and Michael Backes, "ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models", doi: https://doi.org/10.48550/arXiv.1806.01246

[5] 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), San Jose, CA, USA, 2017 pp. 3-18. doi: 10.1109/SP.2017.41

[6] Yeom, Samuel et al. "Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting." 2018 IEEE 31st Computer Security Foundations Symposium (CSF) (2017): 268-282.

[7] Lucas Bourtoule and Varun Chandrasekaran and Christopher A. Choquette-Choo and Hengrui Jia and Adelin Travers and Baiwu Zhang and David Lie and Nicolas Papernot, "Machine Unlearning", doi: https://doi.org/10.48550/arXiv.1912.03817

[8] Shaopeng Fu and Fengxiang He and Dacheng Tao, "Knowledge Removal in Sampling-based Bayesian Inference", doi: https://doi.org/10.48550/arXiv.2203.12964

[9] Chundawat, Vikram S. and Tarun, Ayush K. and Mandal, Murari and Kankanhalli, Mohan, "Zero-Shot Machine Unlearning", doi: https://doi.org/10.48550/arXiv.2201.05629

[10] Vikram S Chundawat and Ayush K Tarun and Murari Mandal and Mohan Kankanhalli, "Can Bad Teaching Induce Forgetting? Unlearning in Deep Networks using an Incompetent Teacher", doi: https://doi.org/10.48550/arXiv.2205.08096

[11] Xulong Zhang and Jianzong Wang and Ning Cheng and Yifu Sun and Chuanyao Zhang and Jing Xiao, "Machine Unlearning Methodology base on Stochastic Teacher Network", doi: https://doi.org/10.48550/arXiv.2308.14322

[12] Chong Chen and Fei Sun and Min Zhang and Bolin Ding, "Recommendation Unlearning", doi: https://doi.org/10.48550/arXiv.2201.06820

[13] Neel, Seth et al. "Descent-to-Delete: Gradient-Based Methods for Machine Unlearning." ArXiv abs/2007.02923 (2020): n. Pag.

[14] Z. Ma, Y. Liu, X. Liu, J. Liu, J. Ma and K. Ren, "Learn to Forget: Machine Unlearning via Neuron Masking," in IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 4, pp. 3194-3207, 1 July-Aug. 2023, doi: 10.1109/TDSC.2022.3194884.

[15] Tarun, Ayush K. and Chundawat, Vikram S. and Mandal, Murari and Kankanhalli, Mohan, "Fast Yet Effective Machine Unlearning", doi: https://doi.org/10.48550/arXiv.2111.08947

[16] Jie Xu and Zihan Wu and Cong Wang and Xiaohua Jia, "Machine Unlearning: Solutions and Challenges", doi: https://doi.org/10.48550/arXiv.2308.07061

[17] Prannay Khosla and Piotr Teterwak and Chen Wang and Aaron Sarna and Yonglong Tian and Phillip Isola and Aaron Maschinot and Ce Liu and Dilip Krishnan, "Supervised Contrastive Learning", doi: https://doi.org/10.48550/arXiv.2004.11362

[18] Yifan Zhang and Bryan Hooi and Dapeng Hu and Jian Liang and Jiashi Feng, "Unleashing the Power of Contrastive Self-Supervised Visual Models via Contrast-Regularized Fine-Tuning", doi: https://doi.org/10.48550/arXiv.2102.06605

[19] Seohui Bae and Seoyoon Kim and Hyemin Jung and Woohyung Lim, "Gradient Surgery for One-shot Unlearning on Generative Model", doi: https://doi.org/10.48550/arXiv.2307.04550

[20] Li, Lei and Xu, Xiaoqian and Spagnolie, Saverio E., "A Locally Gradient-Preserving Reinitialization for Level Set Functions", doi: https://doi.org/10.48550/arXiv.1504.02064

[21] Chunming Li, Chenyang Xu, Changfeng Gui and M. D. Fox, "Level set evolution without re-initialization: a new variational formulation," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 430-436 vol. 1, doi: 10.1109/CVPR.2005.213.

[22] Alexander Warnecke and Lukas Pirch and Christian Wressnegger and Konrad Rieck, "Machine Unlearning of Features and Labels", doi: https://doi.org/10.48550/arXiv.2108.11577

[23] https://storage.googleapis.com/unlearning-challenge/retrain_weights_resnet18_cifar10.pth

[24] https://www.kaggle.com/competitions/neurips-2023-machine-unlearning

[25] https://storage.googleapis.com/unlearning-challenge/forget_idx.npy

[26] https://storage.googleapis.com/unlearning-challenge/forget_idx.npy

[27] https://storage.googleapis.com/unlearning-challenge/retrain_weights_resnet18_cifar10.pth

[28] https://unlearning-challenge.github.io/assets/data/Machine_Unlearning_Metric.pdf

[29] Alexander Becker and Thomas Liebig, "Evaluating Machine Unlearning via Epistemic Uncertainty", doi: https://doi.org/10.48550/arXiv.2208.10836

[30] Shashwat Goel and Ameya Prabhu and Amartya Sanyal and Ser-Nam Lim and Philip Torr and Ponnurangam Kumaraguru, "Towards Adversarial Evaluations for Inexact Machine Unlearning", doi: https://doi.org/10.48550/arXiv.2201.06640

[31] Xulong Zhang and Jianzong Wang and Ning Cheng and Yifu Sun and Chuanyao Zhang and Jing Xiao, "Machine Unlearning Methodology base on Stochastic Teacher Network", doi: https://doi.org/10.48550/arXiv.2308.14322

[32] Zhang Y, Lu Z, Zhang F, Wang H, Li S. Machine Unlearning by Reversing the Continual Learning. Applied Sciences. 2023; 13(16):9341. https://doi.org/10.3390/app13169341

[33] Thanveer Shaik and Xiaohui Tao and Haoran Xie and Lin Li and Xiaofeng Zhu and Qing Li, "Exploring the Landscape of Machine Unlearning: A Comprehensive Survey and Taxonomy", doi: https://doi.org/10.48550/arXiv.2305.06360

[34] Meghdad Kurmanji and Peter Triantafillou and Eleni Triantafillou, "The Brainy Student: Scalable Unlearning by Selectively Disobeying the Teacher"

[35] Chongyu Fan and Jiancheng Liu and Yihua Zhang and Dennis Wei and Eric Wong and Sijia Liu, "SalUn: Empowering Machine Unlearning via Gradient-based Weight Saliency in Both Image Classification and Generation", doi: https://doi.org/10.48550/arXiv.2310.12508

[36] Lingzhi Wang and Tong Chen and Wei Yuan and Xingshan Zeng and Kam-Fai Wong and Hongzhi Yin, "KGA: A General Machine Unlearning Framework Based on Knowledge Gap Alignment", doi: https://doi.org/10.48550/arXiv.2305.06535

[37] Marco Cotogni and Jacopo Bonato and Luigi Sabetta and Francesco Pelosin and Alessandro Nicolosi, "DUCK: Distance-based Unlearning via Centroid Kinematics", doi: https://doi.org/10.48550/arXiv.2312.02052

[38] https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/459200
[39] https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/overview
[40] Jiaming Zhang and Xingjun Ma and Qi Yi and Jitao Sang and Yu-Gang Jiang and Yaowei Wang and Changsheng Xu, “Unlearnable Clusters: Towards Label-agnostic Unlearnable Examples”, doi: https://doi.org/10.48550/arXiv.2301.01217
[41] https://www.kaggle.com/datasets/tourist55/alzheimers-dataset-4-class-of-images