Machine unlearning survey

Yiwen Jiang, Shenglong Liu, Tao Zhao, Wei Li, Xianzhou Gao

Yiwen Jiang, Shenglong Liu, Tao Zhao, Wei Li, Xianzhou Gao, "Machine unlearning survey," Proc. SPIE 12500, Fifth International Conference on Mechatronics and Computer Technology Engineering (MCTE 2022), 125006J (16 December 2022); doi: 10.1117/12.2660330

SPIE.

Event: 5th International Conference on Mechatronics and Computer Technology Engineering (MCTE 2022), 2022, Chongqing, China

Machine Unlearning Survey

Yiwen Jiang[1a], Shenglong Liu[2b], Tao Zhao[3c], *Wei Li[4d], Xianzhou Gao[5e]
[1]Big Data Center of State Grid Corporation of China, Beijing, China
[2]Big Data Center of State Grid Corporation of China, Beijing, China
[3]Big Data Center of State Grid Corporation of China, Beijing, China
[4]Zhejiang Xiyou Information Technology Co., Ltd, Nanjing, China
[5]Global energy interconnection research institute Co., Ltd, Nanjing, China
[a]yiwen-jiang@sgcc.com.cn, [b]shenglong-liu@sgc.com.cn, [c]tao-zhao@sgcc.com.cn,
[d*] Corresponding author: honvey2008@163.com, [e]gaoxianzhou@geiri.sgcc.com.cn

Abstract

Many online platforms have widely deployed machine learning models as a service. Many of these applications require users to upload their data for model training, but it also induces privacy risks. Once the user wants to leave the application, how to make the application unlearn the uploaded data, which is called machine unlearning, is worthy of study. In this article, we provide a survey of machine unlearning with an approximate and exact guarantee. We summarize the existing machine unlearning approaches and discuss their merits and drawbacks in this field.

Keywords-machine unlearning; exact machine unlearning; approximate machine unlearning

1. INTRODUCTION

Many applications have adopted machine learning to build several services. In order to achieve a better accuracy of the machine learning model, most of these applications need to analyze a large amount of data collected from individual users. Once the model has been well trained to fit users' data, it can provide online prediction service for other users [1]. But it may cause privacy leakage when the application is opened for external users. Previous works, such as membership inference attack [2]–[4] and model inversion attack [5]–[7], have illustrated the possibility to extract the origin training data from the machine learning model, even if the attacks have access only in the black-box manner to the model [8], [9].

Since these data may be sensitive and involve private information such as medical records, the privacy issue has attracted a lot of attention recently. Several legislations, such as the General Data Protection Regulation (GDPR) in the European Union, the California Consumer Privacy Act in the United States, and PIPEDA privacy legislation in Canada, have introduced a series of laws to declare *the right to be forgotten* [10]. It mandates that *companies should take reasonable steps to achieve the erasure of personal data concerning* [11]. When applying *the right to be forgotten* to the machine learning applications, it requires not only the deletion of the users' raw data from the storages, but also the erasure of remaining information in the machine learning models. In other words, when a user proposes a request for data deletion, these machine learning models should take several steps to unlearn this data from the model weights to satisfy the guarantee of users' data privacy.

There are several approaches to protect the users' privacy when using machine learning services. Differential privacy [12] can protect the users' privacy by making it difficult to distinguish two neighboring datasets. For a randomized mechanism, M is if for any two neighboring datasets, we have. Differential privacy can be applied to many privacy-preserving ML applications [13], [14], but it is tangential to the target of *the right to be forgotten* [10]. Although differential privacy can give a bound of the effect of any individual data, there remains a non-zero contribution in the model, which does not completely achieve *the right to be forgotten*.

In this survey, we give an overview of the recent surge of the deletion of the users' information from a well-trained model, which is also referred to machine unlearning, to satisfy *the right to be forgotten* [10]. A basic solution is to retrain the model without the forgotten data, but apparently it needs much computational cost and time. Previous works have studied many approaches to realize exact or approximate machine unlearning on the many machine learning models. We will first

Fifth International Conference on Mechatronics and Computer Technology Engineering
(MCTE 2022), edited by Dalin Zhang, Proc. of SPIE Vol. 12500, 125006J
 · 0277-786X · doi: 10.1117/12.2660330

give a general introduction to the scenario which these works consider, then give a comprehensive review of these previous works.

2. OVERVIEW

First, we summarize several previous works in Table 1. Then we present a review of exact machine unlearning and approximate machine unlearning respectively in the following sections.

TABLE 1. SUMMARY

Approach	*Unlearning Type*	*Application*	*Different Details*
[20]	Exact	Statistical query (SQ) learning	Use summations instead of raw data
[21]	Exact	K-means clustering	Quantizes the centroids, save metadata
[22]	Exact	Deep learning	Sharded, Isolated, Sliced, and Aggregate
[24]	Exact	Native learning	Decremental update procedures
[25]	Exact	Extremely randomized trees	Maintain several variants for non-robust splits
[26]	Exact	Random forest	Store and update the data count in the tree node
[15]	Approximate	Deep learning	Scrubbing information from the Hessian matrix
[28]	Approximate	Deep learning	Exploit Neural Tangent Kernel to scrub information
[32]	Approximate	Deep learning	Train model in the mixed-privacy setting
[33]	Approximate	Deep learning	Apply the Newton step based on the gradients
[34]	Approximate	Deep learning	Utilize the gradient to compute the weight update.
[29]	Approximate	Bayesian learning	Variational inference and Markov chain Monte Carlo
[36]	Approximate	Federated learning	Reconstruct and calibrate the unlearned model

3. EXACT MACHINE UNLEARNING

The primary goal of realizing the right to be forgotten [10] is to remove the information in the machine model totally, which is exact machine unlearning. Exact machine unlearning can be conducted by several approaches. The simple one is to retrain the entire model from scratch with the remaining dataset, but it requires a large computational cost to put it into practice. Previous works have studied several approaches to train the model with consideration of unlearning cost, such as using summations instead of raw data in statistical query (SQ) learning [17] to make unlearning easy or utilize shards and

slice to isolate the data contribution during the training phase. And there are also decision trees [18] and random forest [19] learning algorithms that can boost the unlearning phase. All these approaches can achieve the exact machine unlearning, which means there will be nothing information left after the unlearning phase, as is.

3.1 SQ learning approach

[20] proposes a general, efficient unlearning approach by introducing a layer of a small number of summations between the learning algorithm and the training data to break down the dependencies. As the learning algorithm will not use the origin data directly, it is simple to remove an origin data and its summation, then compute the updated model. The summation form follows statistical query (SQ) learning [17] which only permits the algorithm to query the statistics attributes of the entire dataset. By using a naive Bayes classifier, it can generate the summations from the raw data and build it into a final model. To unlearn a sample, it can be conducted by updating the corresponding summations. This approach can achieve efficient and exact machine unlearning, but the limitation is that it only suits for the simple dataset which can apply the SQ learning.

3.2 Clustering approach

[21] investigates the setting of k-means clustering and proposes provably efficient deletion algorithms Q-k-means which is the quantized variant of the k-means algorithm. In the iterative process, it first quantized the centroids before updating the partition. Then it memorizes the optimization state into the model's metadata for use at deletion time and introduces a balance correction step to compensate for the partitions. In the unlearning phase, it can utilize the metadata saved from training time to check whether the deleted data may cause a different quantized centroid or not. If it does, a totally retraining from scratch is needed for unlearning. Otherwise, it only needs to update the metadata to adjust from the deletion of the data which is quite quick. The Q-k-means supports exact data deletion by quantizing the centroids to avoid totally retraining in the clustering scenario.

3.3 SISA approach

[22] proposes a SISA approach to realize fast machine unlearning. SISA is short for Sharded, Isolated, Sliced, and Aggregate. In the training phase, it first divides the total dataset into S shards randomly, i.e. Then for each shard, splitting it into R slices. After that, the training phase can be started with the model which is random initialization, and train S individual models in these shards. The model is first trained with for epochs and saves the corresponding model state into storage. Then trained with for epochs and saves the model status. Repeat this process until every slice has been incrementally trained for the model. When every shard has been trained over, SISA ensemble [23] these models with a simple voting algorithm to avoid the additional information overlap. Since every shard is isolated in the training phase, there is no information sharing with them, so a single data will only contribute to exactly one model of the corresponding shard.

Owing to the isolation of shards, when the user proposes to remove the data, it can easily be conducted by finding the corresponding shard and the slice the data belongs to, then remove these models after this slice. Then remove the origin data from this slice and retrain the model with the same process of training phase. Since the scale of each slice is much smaller than the total dataset, this retraining phase will be much faster than retraining the model from scratch. But it also has an obvious drawback that the storage costs needed for these internal model states are large. Besides, the sliced based incremental training is only suited for the iterative learning algorithms, and the number of each shard is also smaller which induces the "weak learner" and leads to the significant degradation of predictive performance.

3.4 Decremental update approach

[24] has developed decrement update procedures for three learning tasks. For the recommender system, it proposes to use item-based collaborative filtering which compares user interactions to find related items. Since the input data can be represented in a binary history matrix, which is the user-set and is the item-set. The learning phase is to compute the co-occurrence matrix and the similarity matrix by inspecting co-occurrence counts. The unlearning phase can be conducted by looping over the history matrix, and change the co-occurrence matrix which related to the removal data. Then update the similarity matrix S to finish the unlearning process. For the regression task, it utilizes ridge regression to solve the equation. To remove user data, it computes to get the updated model. For classification tasks, it leverages a nearest neighbor-based classification algorithm with local sensitive hashing. For each data, it conducts a projection to the corresponding buckets in the hash table, then computes the k-nearest neighbor to classify the input data. In the unlearning phase, it removes the corresponding data with the buckets it belongs to from the hash table. These decrement update

approaches can suit the basic task to realize exact unlearning but lack the ability to handle the more complex dataset and models.

3.5 Tree-based approach

[25] proposes Hedgecut to utilize the ensemble of extremely randomizedtrees to realize low-latency machine unlearning. In the training phase, according to whether the node will be changed in the unlearning phase or not, Hedgecut splits the tree node into two kinds: robust splits and non-robust splits. For the non-robust splits, Hedgecut will maintain several variants for the cases where split decisions change. In the unlearning phase, Hedgecut only updates the leaf node in the non-robust splits and never revises the robust node. To unlearn a certain sample, it removes the corresponding leaf node and activates subtree variants which might become the preferred split. Hedgecut can realize data removal without retraining by pre-computing several alternative subtrees for sensitive data splits. But it assumes only a very small percentage of instances would be deleted and only can be applied to extremely randomized trees.

[26] proposes Data Removal-Enabled (DaRE) forest which is a tree ensemble and each tree in it is trained independently. When building the trees, DaRE will store and update the count for the number of instances and positive instances for decision nodes as well as a set of thresholds of these attributes. For leaf nodes, DaRE only stores the count for the number of instances and positive instances, along with a list of training instances. When deleting a specific instance, DaRE first updates these statistics and check if a subtree needs retraining. DaRE only retrains these subtrees when they meet the threshold. Otherwise, DaRE can finish deleting by updating the corresponding counts in the decision nodes and leaf nodes.

4. APPROXIMATE MACHINE UNLEARNING

Unlike exact machine unlearning which tries to clear all the information of the forgotten data, many previous works also study the challenge to make a well-trained model to 'forget' the specific data and output the resulting model which behaves like an ideal model which is only trained on the remaining dataset. Formally speaking, for a resulted model after the unlearning phase, and an ideal model which involves nothing about the forgotten data, we have, in which represents the upper bound of remaining information in the resulting model. To achieve this goal, [27] proposes an output filtering technique to prevent private data from being leaked. Based on the gradients in the SGD, [15] [28] propose a scrubbing mechanism for deep neural networks to scrub the information in the model parameters. [29] develops forgetting algorithms in the Bayesian inference using variational inference. We summarize these works in the following section.

4.1 Gradient-based approach

[15] proposes a scrubbing function that can be applied to the origin model and clear the corresponding information from the model's weights. It first illustrates the definition of the forgetting problem by studying the mutual information between two models. Then it proposes the optimal scrubbing algorithm for the model which uses the quadratic loss. By considering a more complex scenario, it uses the Newton update to simplify the scrubbing function. By utilizing the Fisher Information [30] approximation, the scrubbing can be applied to the deep neural network which is widely used in many applications. The evaluation shows this method can realize the data forgotten with a small loss of accuracy on the testing dataset. This method can realize unlearning without any retraining but requires a large computation cost which makes it hard to practice.

[28] studies the mutual information and gives a close form bound for gaussian scrubbing. Then by exploiting Neural Tangent Kernel (NTK) [31] which can posit the large networks as their linear approximation, it gives a new scrubbing procedure which consists of NTK approximation and gaussian noise. The experiments show that it can forget several data samples effectively and reduce the probability of membership inference attack.

[32] proposes an unlearning algorithm on the large-scale vision network with a mixed-privacy setting. It first divides the dataset into two kinds: the core dataset and the user dataset. The core dataset is owned by the server and the user dataset is uploaded by users. Only the user dataset will be unlearned. According to the dataset attribute, it proposes to build a complex model in the core dataset and build a linear model in the user dataset. Since the Hessian matrix can be easily computed in the linear model, it can perform mixed-linear forgetting efficiently. It also studies the practical scenario which may need to forget several data in a sequential way.

[33] focuses on the linear and regression models and proposes a removal mechanism to remove the influence of the deleted data points. The removal mechanism applies the Newton step based on the gradients of model parameters. And the residual error of this mechanism decreases quadratically with the size of the training set. Then to ensure that an adversary

cannot extract information from the resulting model, it also develops a certified-removal mechanism to mask the residual using an approach that randomly perturbs the training loss. This method can be applied to the last fully connected layer of a deep neural network.

[34] also focuses on the linear and logistic models to propose a new approximate deletion method which has linear computation cost in the feature dimension . The projective residual update (PRU) utilizes the gradient of the loss to compute the model weight update. It also introduces the feature injection test for evaluating data removal from models which can measure the removal of the model's knowledge of a sensitive, highly predictive feature present in the data.

4.2 Bayesian approach

An energy-based Bayesian inference forgetting (BIF) framework is proposed in [29] to unlearn the data sample. The pre-defined energy functions can characterize the influence of some specific datums on the learned models.The approximation error of BIF framework can be bounded with, which is negligible when the training sample set is sufficiently large. It develops two certified knowledge removal algorithms--variational inference forgetting and Markov chain Monte Carlo (MCMC) forgetting. The experiments show the BIF can be applied to the Gaussian Mixture Model as well as Bayesian Neural Network [35] and it is significantly faster than re-training from scratch.

4.3 Federated approach

[36] focuses on the federated learning scenario to exploit the gap due to inherent distinction in the way FL and ML learn from data. It takes the first step to fill this gap by presenting FedEraser which can eliminate the influence of a federated client's data on the global FL model while significantly reducing the time used for constructing the unlearned FL model. FedEraser reconstructs the unlearned model by leveraging the historical parameter updates which were saved in the training phase. To further increase the model usability, FedEraser uses a novel calibration method to calibrate the retained updates. By training and updating the calibration on the client side, the server can aggregate these calibration updates into the final model to finish the unlearning phase. FedEraser can fit many datasets and models and it is non-intrusive and can serve as an opt-in component inside existing FL systems.

5. EVALUATION

In this section, we evaluate the model usability and efficiency with different kinds of unlearning methods. We run different methods on two real-world datasets. The MNIST[37] and CIFAR10[38] are 10-classes image datasets that have been widely used in model evaluation. Considering the task complexity of different datasets, we use LeNet in the MNIST dataset and we use ResNet18 on the CIFAR10 dataset.

We choose SISA [22] as the representative of exact machine unlearning methods and FedEraser [36] as the representative of approximate machine unlearning methods. We set the number of clients to 20. For the SISA method, we split the dataset into 20 parts with equal size. For the FedEraser method, we set the calibration ratio = 0.5, and the retaining interval = 2. As for other training hyper-parameters, such as learning rate, training epochs, and batch size, we use the same settings. We also implement the baseline model, which does not consider the unlearning request so it handles the unlearning requests by retaining from scratch, which gives it a higher accuracy and long unlearning time. We evaluate the performance of different methods using standard metrics in the ML field, including the accuracy. We also measure the unlearning time consumed by different methods to make a given global model forget one of the clients.

TABLE 2. MODEL ACCURACY OF BASELINE, SISA AND FEDERASER

Datasets	***Baseline***	***SISA***	***FedEraser***
MNIST	98.6%	98.2%	98.6%
CIFAR10	88.1%	76.2%	56.2%

The results of model accuracy of the two scenarios are presented in Table 2. From the results we can see that both methods perform closely as baseline on MNIST dataset. However, both methods perform not well enough on CIFAR10 dataset. The SISA method achieves an accuracy of 76.2%, which has 11.9% difference from that of baseline. The FedEraser method achieves an accuracy of 56.2%, which is almost unacceptable.

TABLE 3. TIME CONSUMPTION OF BASELINE, SISA AND FEDERASER

Datasets	***Baseline***	***SISA***	***FedEraser***
MNIST	1080s	219s	223s
CIFAR10	3010s	569s	751s

Table 3 shows the time consumption of the two methods in constructing the global models. According to the results, it is obvious that SISA takes the same order of magnitude of the time as FedEraser to reconstruct the global model while the baseline takes a higher order of magnitude of the time.

6. CONCLUSIONS

In this survey, we have given a comprehensive overview with the machine unlearning tasks, which refers to the information removal from the well-trained model. As the right to be forgotten [10] is adopted by many legislations, totally deleting the users' data from both the datasets and the model weights is a more complex task which needs to be studied. Previous works have searched many approaches in the retraining or the scrubbing manner and can achieve machine unlearning with exact guarantee or approximate guarantee. But how to make a trade-off to balance the unlearning cost with model performance is still an open question worth studying.

Besides, there are still abundant interesting directions opened up ahead. To name a few, unlearning verification is a vital topic. As the costs of machine unlearning are considerably high, the server may deceive the clients that the data is forgotten. Unlearning of different granularity also remains a significant problem. Different unlearning requests, such as user-level and data-level, can lead to different retraining strategies. We plan to investigate these appealing subjects in the near future.

REFERENCES

[1] M. Ribeiro, K. Grolinger, and M. A. M. Capretz, "MLaaS: Machine learning as a service," *Proc. - 2015 IEEE 14th Int. Conf. Mach. Learn. Appl. ICMLA 2015*, no. c, pp. 896–902, 2016, doi: 10.1109/ICMLA.2015.152.

[2] M. A. Rahman, T. Rahman, R. Laganière, N. Mohammed, and Y. Wang, "Membership inference attack against differentially private deep learning model," *Trans. Data Priv.*, vol. 11, no. 1, pp. 61–79, 2018.

[3] P. Irolla and G. Chatel, "Demystifying the Membership Inference Attack," *2019 12th C. Conf. Cybersecurity Privacy, C. 2019*, pp. 1-17, 2019, doi: 10.1109/CMI48017.2019.8962136.

[4] R. Xu, N. Baracaldo, and J. Joshi, "Privacy-Preserving Machine Learning: Methods, Challenges and Directions," pp. 1–40, 2021, [Online]. Available: http://arxiv.org/abs/2108.04417.

[5] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The secret revealer: Generative model-inversion attacks against deep neural networks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 250–258, 2020, doi: 10.1109/CVPR42600.2020.00033.

[6] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," *Proc. ACM Conf. Comput. Commun. Secur.*, vol. 2015-Octob, pp. 1322–1333, 2015, doi: 10.1145/2810103.2813677.

[7] Y. Wang, C. Si, and X. Wu, "Regression model fitting under differential privacy and model inversion attack," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2015-Janua, no. Ijcai, pp. 1003–1009, 2015.

[8] A. N. Bhagoji, W. He, B. Li, and D. Song, "Practical black-box attacks on deep neural networks using efficient query mechanisms," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11216 LNCS, pp. 158–174, 2018, doi: 10.1007/978-3-030-01258-8_10.

[9] A. Eyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," *35th Int. Conf. Mach. Learn. ICML 2018*, vol. 5, pp. 3392–3401, 2018.

[10] S. Shastri, M. Wasserman, and V. Chidambaram, "The seven sins of personal-data processing systems under GDPR," *11th USENIX Work. Hot Top. Cloud Comput. HotCloud 2019, co-located with USENIX ATC 2019*, no. i, pp. 1-7, 2019.

[11] "Regulation(eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation)," OJ, vol. L 119, pp. 1–88, 4.5.2016.
[12] M. Abadi *et al.*, "Deep learning with differential privacy," *Proc. ACM Conf. Comput. Commun. Secur.*, vol. 24-28-Octo, pp. 308–318, 2016, doi: 10.1145/2976749.2978318.
[13] J. Hamm, A. C. Champion, G. Chen, M. Belkin, and D. Xuan, "Crowd-ML: A Privacy-Preserving Learning Framework for a Crowd of Smart Devices," *Proc. - Int. Conf. Distrib. Comput. Syst.*, vol. 2015-July, pp. 11–20, 2015, doi: 10.1109/ICDCS.2015.10.
[14] M. Al-Rubaie and J. M. Chang, "Privacy-Preserving Machine Learning: Threats and Solutions," *IEEE Secur. Priv.*, vol. 17, no. 2, pp. 49–58, 2019, doi: 10.1109/MSEC.2018.2888775.
[15] A. Golatkar, A. Achille, and S. Soatto, "Eternal sunshine of the spotless net: Selective forgetting in deep networks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 9301–9309, 2020, doi: 10.1109/CVPR42600.2020.00932.
[16] J. R. Hershey and P. A. Olsen, "Approximating the Kullback Leibler Divergence," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process.*, no. 7, pp. 317–320, 2007.
[17] V. Feldman, C. Guzmán, and S. Vempala, "Statistical query algorithms for mean vector estimation and stochastic convex optimization," *Math. Oper. Res.*, vol. 46, no. 3, pp. 912–945, 2021, doi: 10.1287/MOOR.2020.1111.
[18] B. R. Patel and K. K. Rana, "A Survey on Decision Tree Algorithm For Classification," *Ijedr*, vol. 2, no. 1, pp. 1–5, 2014.
[19] A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintha, and S. Kundu, "Improved Random Forest for Classification," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 4012–4024, 2018, doi: 10.1109/TIP.2018.2834830.
[20] Y. Cao and J. Yang, "Towards making systems forget with machine unlearning," *Proc. - IEEE Symp. Secur. Priv.*, vol. 2015-July, pp. 463–480, 2015, doi: 10.1109/SP.2015.35.
[21] A. A. Ginart, M. Y. Guan, G. Valiant, and J. Zou, "Making AI forget you: Data deletion in machine learning," *Adv. Neural Inf. Process. Syst.*, vol. 32, no. NeurIPS, pp. 1–28, 2019.
[22] L. Bourtoule *et al.*, "Machine unlearning," *Proc. - IEEE Symp. Secur. Priv.*, vol. 2021-May, pp. 141–159, 2021, doi: 10.1109/SP40001.2021.00019.
[23] H. M. Gomes, J. P. Barddal, A. F. Enembreck, and A. Bifet, "A survey on ensemble learning for data stream classification," *ACM Comput. Surv.*, vol. 50, no. 2, pp. 1-36, 2017, doi: 10.1145/3054925.
[24] S. Schelter, "'Amnesia'-Towards Machine Learning Models That Can Forget User Data Very Fast," *Cidr*, no. Section 6, pp. 1–4, 2020.
[25] S. Schelter, S. Grafberger, and T. Dunning, "HedgeCut: Maintaining Randomised Trees for Low-Latency Machine Unlearning," *Proc. ACM SIGMOD Int. Conf. Manag. Data*, pp. 1545–1557, 2021, doi: 10.1145/3448016.3457239.
[26] J. Brophy and D. Lowd, "Machine Unlearning for Random Forests," pp. 1-29 ,2020, [Online]. Available: http://arxiv.org/abs/2009.05567.
[27] T. Baumhauer, P. Schöttle, and M. Zeppelzauer, "Machine Unlearning: Linear Filtration for Logit-based Classifiers," *arXiv*, 2020, [Online]. Available: http://arxiv.org/abs/2002.02730.
[28] A. Golatkar, A. Achille, and S. Soatto, "Forgetting Outside the Box: Scrubbing Deep Networks of Information Accessible from Input-Output Observations," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12374 LNCS, pp. 383–398, 2020, doi: 10.1007/978-3-030-58526-6_23.
[29] S. Fu, F. He, Y. Xu, and D. Tao, "Bayesian Inference Forgetting," pp. 1–33, 2021, [Online]. Available: http://arxiv.org/abs/2101.06417.
[30] J. J. Rissanen, "Fisher information and stochastic complexity," *IEEE Trans. Inf. Theory*, vol. 42, no. 1, pp. 40–47, 1996, doi: 10.1109/18.481776.
[31] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 2018-December, no. 5, pp. 8571–8580, 2018.
[32] A. Golatkar, A. Achille, A. Ravichandran, M. Polito, and S. Soatto, "Mixed-Privacy Forgetting in Deep Networks," pp. 1-21, 2020, doi: 10.1109/cvpr46437.2021.00085.
[33] C. Guo, T. Goldstein, A. Hannun, and L. van der Maaten, "Certified data removal from machine learning models," *37th Int. Conf. Mach. Learn. ICML 2020*, vol. PartF168147-5, no. i, pp. 3790–3800, 2020.

[34] Z. Izzo, M. A. Smart, K. Chaudhuri, and J. Zou, “Approximate Data Deletion from Machine Learning Models,” *arXiv*, vol. 130, pp. 1-20, 2020, [Online]. Available: http://arxiv.org/abs/2002.10077.
[35] I. Kononenko, “Bayesian neural networks,” *Biol. Cybern.*, vol. 61, no. 5, pp. 361–370, 1989, doi: 10.1007/BF00200801.
[36] G. Liu, X. Ma, Y. Yang, C. Wang, and J. Liu, “Federated Unlearning,” pp. 1–10, 2020, [Online]. Available: http://arxiv.org/abs/2012.13891.
[37] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, “Emnist: Extending mnist to handwritten letters,” in 2017 international joint conference on neural networks (IJCNN). IEEE, 2017, pp. 2921–2926.
[38] A. Krizhevsky, G. Hinton et al., “Learning multiple layers of features from tiny images,” pp. 54-57, 2009.