Journal Pre-proof

An overview of machine unlearning

Chunxiao Li, Haipeng Jiang, Jiankang Chen, Yu Zhao, Shuxuan Fu, Fangming Jing, Yu Guo

PII: S2667-2952(24)00057-6
DOI: https://doi.org/10.1016/j.hcc.2024.100254
Reference: HCC 100254

To appear in: *High-Confidence Computing*

Received date : 12 January 2024
Revised date : 19 February 2024
Accepted date : 7 June 2024

Please cite this article as: C. Li, H. Jiang, J. Chen et al., An overview of machine unlearning, *High-Confidence Computing* (2024), doi: https://doi.org/10.1016/j.hcc.2024.100254.

An Overview of Machine Unlearning

Chunxiao Li[a], Haipeng Jiang[a], Jiankang Chen[a], Yu Zhao[a], Shuxuan Fu[b], Fangming Jing[c] and Yu Guo[a]

[a]Beijing Normal University, , Beijing, 100875, Beijing, China
[b]North China Electric Power University., , Beijing, 102206, Beijing, China
[c]Ministry of Civil Affairs of the People's Republic of China, , Beijing, 100721, Beijing, China

ARTICLE INFO

Keywords:
machine unlearning
unlearning definition
unlearning requirements and validation
unlearning algorithms

ABSTRACT

Nowadays, machine learning is widely used in various applications. Training a model requires huge amounts of data, but it can pose a threat to user privacy. With the growing concern for privacy, the "Right to be Forgotten" has been proposed, which means that users have the right to request that their personal information be removed from machine learning models. The emergence of machine unlearning is a response to this need. Implementing machine unlearning is not easy because simply deleting samples from a database does not allow the model to "forget" the data. Therefore, this paper summarises the definition of the machine unlearning formulation, process, deletion requests, design requirements and validation, algorithms, applications, and future perspectives, in the hope that it will help future researchers in machine unlearning.

1. Introduction

1.1. Motivation of machine unlearning

Machine Learning has attracted a lot of attention and has evolved into a key technology for successful applications such as intelligent computer vision, speech recognition, medical diagnostics, etc. However, for reasons of privacy and usability and the right to be forgotten, some sample-specific information needs to be removed from the model. However, for reasons of privacy, usability, and the right to be forgotten, information about a particular sample needs to be removed from the model, which is known as Machine Unlearning. There are many reasons why users may want to remove their information from the model. We have grouped these reasons into four main categories: security, privacy, usability, and accuracy. Each of these reasons is explained below:

Security. Deep learning models are vulnerable to adversarial attacks that generate adversarial data similar to the original data, forcing the model to make incorrect predictions, which can have serious consequences. For example, in healthcare, wrong predictions can lead to incorrect diagnosis, inappropriate treatment, and even death. Therefore, detecting and deleting adversarial data is crucial to ensure the security of models, which need to be able to delete adversarial data through machine unlearning mechanisms once an attack is detected Cao and Yang (2015); Marchant, Rubinstein and Alfeld (2022).

Privacy. With privacy regulations such as the European Union's GDPR, users have the right to request that cloud servers delete their personal information. To some extent, the rise of this legislation is a result of privacy breaches. For example, cloud servers may leak user data due to multiple copies of data held by different parties, backup policies, and replication policies Singh and Anand (2017). As a result, users want to delete their data to avoid the risk of data leakage.

Usability. People have different preferences for online applications or services, and this is particularly evident in recommender systems. If an application is unable to completely remove incorrect data associated with a user (e.g., noise, malicious data, out-of-distribution data), it will generate inappropriate recommendations. For example, a person may accidentally search for an illegal product on his laptop, and then find that even after clearing his web browsing history, he continues to receive recommendations for that product on his mobile phone. This unwanted usability due to the inability to forget data not only generates false predictions, but also leads to fewer users.

Accuracy. Machine learning models can unfairly discriminate due to data bias, such as on the basis of gender or race. For example, in COMPAS, the software used by U.S. courts to adjudicate parole cases, the software prefers African-American offenders to have higher risk scores than white offenders, even though racial information is not part of the input Wang, Li, Wang, Tang and Zhou (2009). Thus, forgetting this biased data can help improve the fairness and accuracy of the model.

1.2. Challenges in machine unlearning

The task of removing specific information from machine learning models is imminent. However, the fact that machine learning models are trained randomly, that their performance is affected by previous data samples, and that inappropriate data deletion can lead to catastrophic unlearning poses a number of challenges for the development of machine unlearning:

First, the training process of machine learning models is stochastic, and we don't know the impact of each data point we see during the training process on the machine learning model. For example, neural networks are usually trained on

*Corresponding author
26059659@qq.com (F. Jing); yuguo@bnu.edu.cn (Y. Guo)
ORCID(s):

small random batches containing a certain number of data samples. In addition, the order of the training batches is also random Bourtoule, Chandrasekaran, Choquette-Choo, Jia, Travers, Zhang, Lie and Papernot (2021), however, specific data samples to be removed need to be removed from all batches. Therefore, how to overcome the difficulties caused by this randomness is an important challenge in the development of machine unlearning.

Second, model training is an incremental process Bourtoule et al. (2021). In short, the tuning of one data sample affects the model's performance on subsequent input data samples, and the model's processing of the current data sample is also affected by the previous data sample. This means that the model's performance is as much related to the data it was exposed to before as it is to the data it will be exposed to after. Determining a way to eliminate the further impact of deleted training samples on model performance is one of the challenges of machine unlearning.

Third, catastrophic unlearning. In general, machine unlearning models typically perform worse than models retrained on the remaining data Bourtoule et al. (2021); Nguyen, Oikawa, Divakaran, Chan and Low (2022a). However, when more data is removed, model degradation can be exponential. This sudden model degradation is often referred to as catastrophic unlearning Nguyen, Low and Jaillet (2020). Although some studies Du, Chen, Liu, Oak and Song (2019) have explored ways to mitigate catastrophic unlearning by designing special loss functions, how to naturally prevent catastrophic unlearning is still an open question.

1.3. Contributions

The purpose of this paper is to provide a comprehensive review of research on machine unlearning and to discuss potential new research directions in machine unlearning. Therefore, the contributions of our survey can be summarised as follows.

First, we design a framework for the unlearning process based on the definition. Based on this, we discuss the design requirements, the different types of unlearning requests, and how to validate the unlearning model.

Second, we summarise the definition of the unlearning problem in machine unlearning systems. This includes formulations of exact and approximate unlearning, as well as definitions of indistinguishable metrics for comparing two given models (i.e., a unlearning model and a retraining model).

In addition, we categorise the main types of algorithms for machine unlearning, and introduce the latest research on applications by domain, so as to provide a clear picture of the development of machine unlearning.

Finally, we analyse some of the current results and potential trends in machine unlearning, and discuss some open research questions that we hope future research can address.

2. Definition of machine unlearning

2.1. Definition of the problem equation

While the application of machine unlearning may arise from security, usability, and privacy reasons, it is often formulated as a privacy concern, where users can request that their data be removed from computer systems and machine learning models Bourtoule et al. (2021); Golatkar, Achille and Soatto (2020); Garg, Goldwasser and Vasudevan (2020); Ginart, Guan, Valiant and Zou (2019). Unlearning requests can also be done for security and usability reasons. For example, a model may be attacked with adversarial data and produce incorrect outputs. Once these types of attacks are detected, the corresponding adversarial data must be removed without compromising the predictive performance of the model.

When completing a deletion request, the computer system needs to delete all the user's data and "forget" about any impact of the model trained on that data. Since the effect of deleting data from the database on the model is negligible, the literature has focused on how to delete data from the model Sekhari, Acharya, Kamath and Suresh (2021); Guo, Goldstein, Hannun and Van Der Maaten (2019); Neel, Roth and Sharifi-Malvajerdi (2021).

In order to properly formulate a unlearning problem, let us first introduce some concepts. First, let Z be the space of examples, i.e., the space of data items or examples. Then, the set of all possible training datasets is denoted as Z^*. Given a particular dataset $D \in Z^*$ as input, we would like to obtain a machine learning model from the hypothesis space H, which covers the parameters and data of the model. The process of training a model on D in a given computer system is implemented by means of a learning algorithm, denoted by a function $A : Z^* \to H$, and the trained model is denoted by $A(D)$.

In order to implement unlearning, a computer system needs a unlearning mechanism represented by a function U that takes as input a training data set $D \in Z^*$, an unlearning set $D_f \in D$ (the data to be unlearned) and a model variable $A(D)$. It returns an unlearning model $U(D_f, D, A(D)) \in H$. An unlearning model should be the same or similar to a retrained model $A(D \backslash D_f)$ (i.e., one that seems to be trained on the remaining data). Note that we assume that A and U are stochastic algorithms, i.e., the outputs are uncertain and can be modelled as conditional probability distributions on a hypothesis space given the input data Marchant et al. (2022), since many learning algorithms are inherently stochastic (e.g., SGD), and some floating-point operations involve stochasticity in their computer implementations Bourtoule et al. (2021). Another caveat is that we do not define the function U precisely in advance, as its definition varies with different settings.

2.2. Accurate unlearning

The central problem in machine unlearning concerns the comparison of two distributions of machine learning models Bourtoule et al. (2021); Brophy and Lowd (2021); Thudi,

Table 1
Important symbols

notation	define
Z	example space
H	assumption space in which the model is located
D	training dataset
D_f	unlearning data set
$D_r = D \backslash D_f$	retraining dataset
$A(.)$	learning algorithm
$U(.)$	unlearning algorithm

Deza, Chandrasekaran and Papernot (2022a). Let $Pr(A(D))$ be the distribution of all models obtained by the learning algorithm $A(.)$ on the dataset D, and $Pr(D_f, D, A(D))$ be the distribution of forgotten models on the dataset D, and Pr(Df, D, A(D)) is the distribution of the machine unlearning model. As mentioned before, the learning algorithm $A(.)$ and the machine unlearning algorithm $U(.)$ are the distributions of the machine unlearning models. and the machine unlearning algorithm $U(.)$ are randomised, so the output $U(.)$ is a distribution rather than a single point. is a distribution rather than a single point.

Definition 1 Given a learning algorithm $A(.)$, a dataset D, and an unlearning set $D_f \in D$, $U(.)$is an exact unlearning process if:

$$Pr(A(D \backslash D_f)) = Pr(U(D, D_f, A(D))) \quad (1)$$

Two key aspects can be derived from this definition. First, the definition does not require that model $A(D)$ be retrained from scratch on $D \backslash D_f$. Instead, it requires some evidence that it looks like a model trained from scratch on $D \backslash D_f$. Second, two models trained on the same dataset should belong to the same distribution. However, defining this distribution is difficult. Therefore, to avoid unlearning that an algorithm is specific to a particular training dataset, we have a more general definition Ginart et al. (2019); Brophy and Lowd (2021).

Definition 2 Given a learning algorithm $A(.)$, $U(.)$ is an exact unlearning process if $\forall T \in H$, $D \in Z^*$, $D_f \subset D$:

$$Pr(A(D \backslash D_f) \in T) = Pr(U(D, D_f, A(D)) \in T) \quad (2)$$

This definition allows us to define the distribution of the model ourselves. A model can either be viewed as a mapping from inputs to outputs, or as a model structured with a specific parameter θ. The model can also be viewed as a distribution of the inputs to the outputs.

If the unlearning process $U(.)$ is set to retraining, absolute equality is guaranteed. For this reason, retraining is sometimes considered to be the only accurate unlearning method. However, retraining itself is very computationally expensive, especially for large models Thudi et al. (2022a). Another disadvantage of retraining is that it does not allow for batch processing, e.g., multiple deletion requests occurring at the same time.

There are many different metrics for comparing the distribution of values in output space and weight space, but sometimes the overhead of doing so is very high. To alleviate this problem, some methods have devised an alternative point-based metric to calculate the distance between two models Shokri, Stronati, Song and Shmatikov (2017).

2.3. Approximate unlearning

Approximate unlearning aims to address these cost-related overheads, and in lieu of retraining, there are strategies such as modifying the architecture and filtering the output Nguyen et al. (2020).

Definition 3 Given $\epsilon \lg 0$, if $\forall T \in H$, $D \in Z^*$, $z \in D$:

$$e^{-\epsilon} \leq \frac{\Pr(U(\mathrm{D}, \mathbf{z} A(D)) \in \mathrm{T})}{\Pr(\mathrm{A}(\backslash z) \in \mathrm{T})} \leq e^{\epsilon} \quad (3)$$

Then we claim that an unlearning algorithm U performs ϵ-validated approximate unlearning on a learning algorithm A. We say that an unlearning algorithm U performs ϵ-validated approximate unlearning. where z is the sample to be removed.

It is worth noting that Equation 3 only defines bounds on a single sample. It is still an open question whether constant bounds can be provided for larger subsets of D. Furthermore, we use exponential bounds because probability distributions are usually modelled with logarithmic functions. In addition, we use exponential bounds because probability distributions are usually modelled with logarithmic functions.

A relaxed version of the ϵ-verification approximation unlearning is also defined in Neel et al. (2021) :

Definition 4 Given $\epsilon, \delta > 0$, if $\forall T \in H$, $D \in Z^*$, $z \in D$:

$$Pr(U(D, z, A(D)) \in T) \leq e \in Pr(A(D \backslash z) \in T) + \delta \quad (4)$$

besides

$$Pr(A(D \backslash z) \in T) \leq e^{\epsilon} Pr(U(D, z, A(D)) \in T) + \delta \quad (5)$$

Then we claim that an unlearning algorithm U performs ϵ-validated approximate unlearning of a learning algorithm A.

Relationship to differential privacy Differential privacy is defined as:

$$\forall T \subseteq H, D, D' : e^{-\epsilon} \leq \frac{\Pr(U(\mathrm{D}, \mathbf{z} A(D)) \in \mathrm{T})}{\Pr(\mathrm{A}(\backslash z) \in \mathrm{T})} \leq e^{\epsilon} \quad (6)$$

Where z is the sample to be removed. Differential privacy implies approximate unlearning: if the learning algorithm A has never memorised the training data in the first place, then removing the training data is not a problem Guo et al. (2019). However, this is the paradox of differential privacy and machine learning. If learning algorithm A is differentially private with respect to any data, it will not learn anything from the data itself Bourtoule et al. (2021).

3. The framework of machine unlearning

3.1. The workflow of machine unlearning

Machine unlearning requires that samples and their effects can be completely and quickly removed from the training dataset and the training model Nguyen et al. (2020);

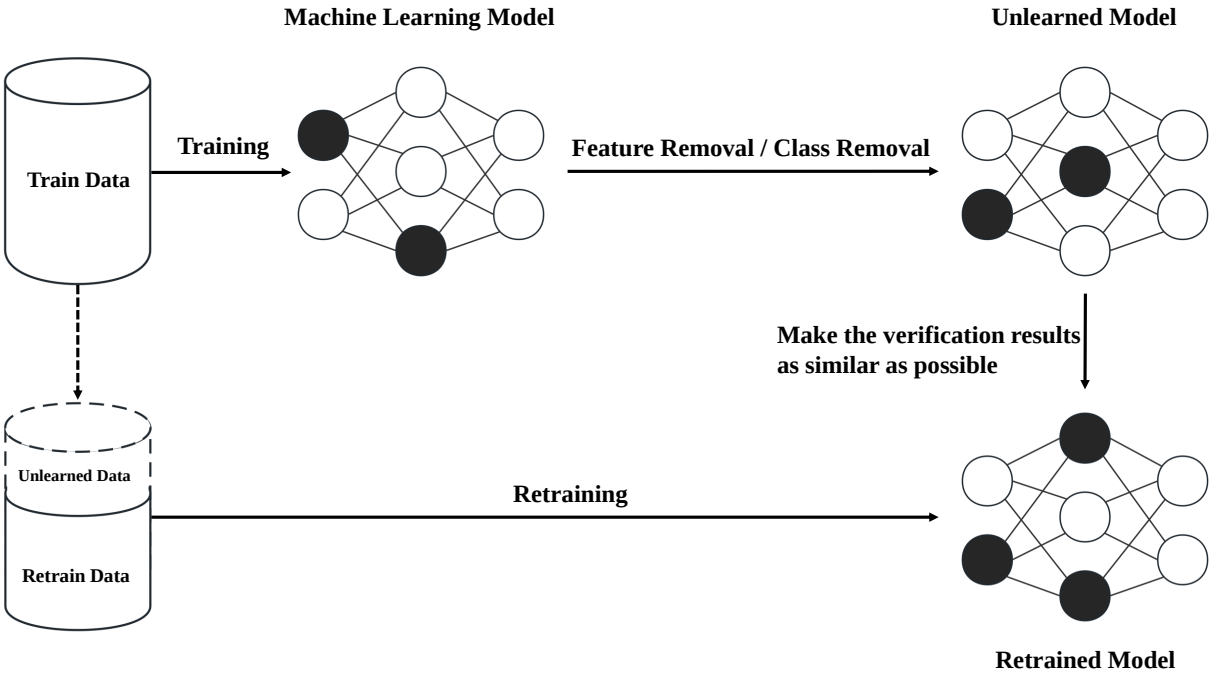

Figure 1: Workflow of Machine Unlearning

Garg et al. (2020); Ginart et al. (2019). Figure 1 illustrates a typical unlearning process. A model is first trained on the full set of data and then learnt by unlearning in response to user requests for forgetting. The resulting unlearning model is compared to a retrained model trained on the dataset to be retained, with features as similar as possible.

3.2. Requests for machine unlearning

3.2.1. Feature deletion

Requests to remove certain samples from a training dataset are the most common requests in machine unlearning Bourtoule et al. (2021). However, in many cases, privacy breaches can arise not only from a single data item, but also from a set of data with similar characteristics or labels Warnecke, Pirch, Wressnegger and Rieck (2021). For example, in an applicant screening system, inappropriate features such as the gender or race of an applicant need to be rejected using unlearning.

In this case, it is unwise to just sequentially forget the affected data items because the overhead of repeated retraining is very high. In addition, unlearning too many data items will degrade the performance of the model, regardless of which unlearning mechanism is used. Therefore, there is a need to perform unlearning at feature levels with an arbitrary number of data items.

Warnecke et al Warnecke et al. (2021) proposed a feature removal technique based on the influence function. More precisely, the update of model parameters from training data is estimated and formulated in closed form. The first and second order derivatives are the key to efficiently compute this update Warnecke et al. (2021).

Guo et al. Guo, Guo, Zhang, Xu and Wang (2022) proposed another technique based on de-entanglement representation to remove features from data. The core idea is to learn the correlation between features from the latent space and the effect of each feature on the output space. Using this information, certain features can be gradually separated from the learning model as needed, and the remaining features can be retained to maintain good accuracy. However, this approach is mainly applicable to deep neural networks in the image domain, where deeper convolutional layers become smaller and therefore abstract features matching the properties of real-world data can be recognised.

3.2.2. Class deletion

In many cases, the data to be unlearned belongs to one or more classes in the training model. For example, in a face recognition application, each class is a human face, so there may be thousands or millions of classes. However, when a user chooses to opt out of the system, their facial information must be deleted without using a sample of their face.

Similar to feature deletion, class deletion is more challenging than deleting certain samples. Although each unlearning step may require a small computational cost due to data partitioning, the overhead is cumulative. However, partitioning the data by class itself does not help train the model in the first place, as learning the differences between classes is at the heart of many learning algorithms Tanha, Abdi, Samadi, Razzaghi and Asadpour (2020). While some of the

above feature removal techniques can be applied to class removal Warnecke et al. (2021), this is not always the case, as class information may be implicit in many cases.
Tarun et al. Tarun, Chundawat, Mandal and Kankanhalli (2023) proposed a class removal method based on data augmentation. The basic concept is to introduce noise into the model to maximise the classification error of the target class. The model is updated by training on this noise without accessing any samples of the target class. Since this impairment step may interfere with the model weights and reduce the classification performance of the remaining classes, a repair step is needed to train the model on the remaining data for one or several periods. Their experiments show that this method can effectively solve the large-scale multi-class problem. In addition, the method is particularly effective in face recognition tasks, where deep neural networks are initially trained on ternary loss and negative samples, and thus the differences between categories are significant Masi, Wu, Hassner and Natarajan (2018).

3.3. Design requirements

When designing an unlearning algorithm, there are several requirements to be taken into account.
Verifiability. In addition to requests for unlearning, another user requirement is to ensure that the model, which has undergone unlearning, now effectively safeguards their privacy. For this reason, a good machine unlearning framework should provide an authentication mechanism for the end user. For example, a backdoor attack can validate machine unlearning by injecting backdoor samples into the training data Sommer, Song, Wagh and Mittal (2020). If the backdoor is detected in the original model but not in the forgetful learning model, the validation is considered successful. However, such validation may be too intrusive for a trustworthy machine learning system, and due to the inherent uncertainty of backdoor detection, validation may still introduce false positives. Further, Guo et al. Guo, Zhao, Hou, Wang and Jia (2023) propose a backdoor-assisted verification method that allows users to efficiently verify machine learning proofs while maintaining the quality of the retrained model.
Consistency. A good unlearning algorithm should be consistent Cao and Yang (2015). That is, a model that has undergone unlearning and a model that has been retrained should make the same predictions for any possible sample of data (whether correct or incorrect). One way to measure this consistency is to calculate the percentage of identical predictions on the test data. By comparing the difference in output space between the two models, this requirement can be designed as an optimisation objective in a unlearning algorithm.
Accuracy. A model that has undergone unlearning should be able to correctly predict the test samples. Or at least its accuracy should be comparable to that of a retrained model. However, since retraining is computationally expensive, retrained models are not always available for comparison. To solve this problem, the accuracy of a model that has undergone unlearning is usually measured on a new test set or compared to the original model He, Meng, Chen, He and Hu (2021).

4. Validation of machine unlearning

The purpose of the validation of machine unlearning is to ensure that one cannot easily distinguish between a unlearning model and a retrained model Thudi, Jia, Shumailov and Papernot (2022b). Furthermore, cloud servers may claim that they have removed these effects from the model, but this is not the case He et al. (2021). Especially for complex deep models with large training datasets, removing a small number of samples will only have a negligible effect on the model. Sometimes even if the samples to be forgotten are indeed removed, the model still has a good chance of making correct predictions because other users may have provided similar samples. This makes the problem of validation of unlearning very difficult.
The most intuitive validation is to retrain a model which naturally does not contain samples to be forgotten.

4.1. Membership inference attack

The goal of this attack is to detect if the target model leaks data Chen, Zhang, Wang, Backes, Humbert and Zhang (2021b); Thudi, Shumailov, Boenisch and Papernot (2022c). Specifically, an inference model is trained to identify new data samples in the training data used to optimise the target model. In Shokri et al. (2017), a set of shadow models is trained on a set of new data items different from the target model. The attack model is then trained based on the predictions of the shadow model on the training and test data to predict whether a data item belongs to the training data or not. The training sets of the shadow and attack models need to have similar data distributions as the target model. Membership inference attacks help to detect data leakage. Therefore, they are useful for validating Chen et al. (2021b).

4.2. Backdoor attacks

A backdoor attack is the injection of a backdoor into data to spoof a machine learning model Wang, Yao, Shan, Li, Viswanath, Zheng and Zhao (2019). The spoofed model makes correct predictions for clean data, but incorrect predictions for toxic data in the target class where the backdoor trigger is set. Backdoor attacks were used in Sommer et al. (2020); Sommer, Song, Wagh and Mittal (2022) to validate the effectiveness of machine unlearning, specifically, all users trained the model with a mixture of clean and toxic data, and some users wanted their data deleted. If a user's data is not deleted successfully, the toxic sample is predicted as the target class. Otherwise, the model will fail to predict the toxic sample as the target class. However, even though the user can increase the proportion of toxic samples to make the failure rate lower, there is no absolute guarantee that this rule will always be correct. Furthermore, backdoor triggers should not be easily detected by the cloud server. For this reason, in Guo et al. (2023), a method based on invisible backdoor triggers is proposed to perform machine

unlearning authentication while preventing dishonest cloud servers from forging proofs to bypass the authentication.

5. The algorithms of unlearning

As mentioned in the definition section, machine unlearning can save time and computational resources by deleting specific data without having to retrain the machine learning model from scratch Chen, Huang and Wang (2021a); Wang, Guo, Xie and Qi (2022). Specific approaches to machine learning can be categorised as model-agnostic, model-intrinsic and data-driven.

5.1. Model-agnostic methods

Model-agnostic machine unlearning methods include machine unleaning processes or frameworks that are applicable to different models. However, in some cases, theoretical guarantees are only applicable to specific classes of models (e.g., linear models). Nevertheless, they are still considered model-agnostic because their core ideas are applicable to complex models (e.g., deep neural networks) and have practical effects.

5.1.1. Differential privacy

Differential privacy has previously been used to constrain the effect of data samples on machine learning models. In the case of machine unlearning, differential privacy can also be applied to limit the level of variation in the relevant model parameters by setting parameters to forget the data samples.

5.1.2. Statistical query learning

Statistical query learning is a form of machine learning that trains a model by querying statistical information on the training data rather than the data itself. In this approach, data samples can be effectively forgotten by recomputing the statistical information on the remaining data Bourtoule et al. (2021).

5.1.3. MCMC unlearning (Parameter sampling)

Parameter sampling based machine unlearning is also considered as an approach to train standard machine learning models to forget data samples from the training data Nguyen et al. (2022a). The idea is to sample the distribution of model parameters using Markov Chain Monte Carlo (MCMC), assuming that the dataset to be forgotten is usually much smaller than the training data (otherwise retraining might be a better solution). As a result, the distribution of the parameters of the retrained model is not very different from that of the original model.

5.2. Model-intrinsic methods

Model intrinsic methods are mainly machine unlearning methods designed for specific types of models. Although they are specific to a particular class of models, they have a wide range of applications because many machine learning models are similar in type.

5.2.1. Softmax classifiers (logic-based classifiers) machine unlearning

In the field of machine learning, the softmax classifier is a commonly used method for multi-category classification tasks. It determines the likelihood of each category by transforming the model output into a probability distribution using the softmax function. Unlearning based on softmax classifiers usually involves adjusting the parameters of the model to ensure that the forgotten data no longer influences the decision-making process of the model. This may involve re-evaluating and adjusting the weights of the model so that they no longer reflect the characteristics of the deleted data. Baumhauer et al. proposed a unlearning method for softmax classifiers based on a linear filtering operator Baumhauer, Schöttle and Zeppelzauer (2022), where samples from classes that need to be forgotten are proportionally transferred to other classes. However, this method is only applicable to class deletion.

5.2.2. Machine unlearning of linear models

In the field of machine learning, unlearning for linear models involves removing the influence of specific data points from a trained linear model without completely retraining the model. Linear models, such as linear regression and logistic regression, usually incorporate information from all data points during the training process. Therefore, when certain data points need to be forgotten, specific measures need to be taken to adjust the model parameters to eliminate the effects of these data points. For example, Izzo et al. proposed an approximate unlearning method for linear models Izzo, Anne Smart, Chaudhuri and Zou (2021). The method is based on the influence function, and they approximate the Hessian matrix by combining the gradient method with synthetic data and using projected residual updates. This method is suitable for unlearning small groups of data points from a learning model.

5.2.3. Machine unlearning for tree-based modelling

Tree-based modelling is a classification technique that recursively divides the feature space, where the features and truncation thresholds used to classify the data are determined based on certain criteria, such as information gain. There is a class of tree-based models, called extreme random trees Geurts, Ernst and Wehenkel (2006), which are constructed from a collection of decision trees. These models are very efficient because the candidate sets for splitting features and truncation thresholds are randomly generated. The best candidates are also selected by reducing the Gini impurity, avoiding the heavy computational effort of logarithmic operations. Schelter et al. proposed a unlearning method for extreme random trees by measuring the robustness of splitting decisions Schelter, Grafberger and Dunning (2021). A split decision is robust if the removal of k data items does not change the split decision. Therefore, the learning algorithm is redesigned so that most splits, especially high level splits, are robust. For non-robust splits, all subtree variants are grown from all split candidates and remain until a removal

request is made to revise the split. When this happens, the split is switched to a variant with a higher Gini gain. Therefore, the unlearning process involves recalculating the Gini gain and updating the split if necessary.

5.2.4. Bayesian modelling of machine unlearning

A Bayesian model is a probabilistic model used to estimate posterior probabilities. This process, also known as Bayesian inference, is particularly useful in situations where the loss function is not well defined or does not exist at all. Bayesian models encompass a wide range of machine learning algorithms, including Bayesian neural networks (applying Bayesian methods to neural networks), probabilistic graphical models (representing dependencies between variables through graphs), generative models, topic modelling, and probabilistic matrix decomposition. Machine unlearning of Bayesian models requires special treatment because training already involves optimising the posterior distribution of the model parameters.

5.2.5. Machine unlearning based on DNN models

Deep neural networks are models that automatically learn features from data. Therefore, it is difficult to precisely determine the specific contribution of each data item to the model update. Fortunately, deep neural networks consist of multiple layers, and for layers with convex activation functions, existing methods for unlearning, such as authentication removal mechanisms, can be applied Cao, Wang, Si, Huang and Xiao (2022). For non-convex layers, Golatkar et al. propose a caching method that trains the model on data known to be permanent Golatkar, Achille, Ravichandran, Polito and Soatto (2021). The model is then fine-tuned on user data using certain convex optimisation methods.

5.3. Data-driven methods

Data-driven approach is mainly based on the concept of lean analysis and data closure, through the business data and data business, collect data and use data as production data, through data analysis and mining methods to refine the laws and gain insights, and then apply them to the algorithm to achieve the process of cyclic positive feedback, promote algorithm optimization, and achieve the data-centric approach to machine unlearning.

5.3.1. Data partitioning (efficient retraining)

Methods in the Data Partitioning (Efficient Retraining) category use data partitioning mechanisms to speed up the retraining process. Alternatively, they partially retrain the model by setting certain limits on the accuracy. Bourtoule et al. proposed the well-known SISA framework Bourtoule et al. (2021), which splits the data into partitions and subslices. Each partition has a separate model, and the final output is the aggregated result of multiple models on these partitions. For a sub-slice of each partition, a model checkpoint is stored during training so that a new model can be retrained from an intermediate state.

5.3.2. Data enhancement

Data augmentation is the process of adding more data to support model training, and such a mechanism can also be used for machine unlearning. Huang et al. introduced the concept of error-minimising noise Huang, Ma, Erfani, Bailey and Wang (2021), which works by tricking the model into thinking that there is nothing to learn from a given dataset (i.e., the loss does not change). However, this method can only be used to protect specific data items before the model is trained.

5.3.3. Data impact

The Data Influence class of machine unlearning methods investigates how changes in the training data affect the parameters of the model, where the influence is computed by means of an influence function. However, the influence function depends on the current state of the learning algorithm Wu, Hashemi and Srinivasa (2022). To alleviate this problem, some research has found that it is possible to store the training history of intermediate quantities (e.g., model parameters or gradients) generated at each step of the model training process. The unlearning process then becomes a process of subtracting these history updates. However, due to catastrophic unlearning, the accuracy of the model can be significantly degraded, as the order of input of the training data is important for model learning. In addition, the data impact itself does not verify that the data to be forgotten is still included in the unlearning model.

5.4. Algorithm evaluation

Regarding machine unlearning, the most commonly used metrics to evaluate the performance of anomaly and detection include accuracy, completeness, unlearning time, etc. The following table summarises their definitions.

6. Applications of machine unlearning

6.1. Machine unlearning in federated learning

Due to user privacy, commercial confidentiality, laws and regulations, a large number of information silos have been created, resulting in the inability of various organisations and institutions to integrate raw data together, and then jointly train a large model with better effects, greater information density and stronger capabilities, which has seriously constrained the development of AI. In order to solve such problems, Federated Learning has emerged. Federated Learning is a new machine learning paradigm, in which each participant can use other parties' data to conduct joint modelling in the process of machine learning. The parties do not need to share data resources, i.e., the data are not out of the local context, but are jointly trained to build a shared machine learning model.

Federated Learning is essentially a machine learning paradigm and algorithm designed to solve the problem of data silos. The goal is to share models while keeping data private. For example, if there are multiple participants, each of which owns a set of private clusters and data, and these participants want to train a model together, but traditional machine

Table 2
Evaluation metrics for machine unlearning

norm	define
accuracy	The accuracy of the model is measured on three data sets: the unlearning set, the retained set, and the test set.
completeness	Ensure that the effect of the removed samples on the unlearning model is completely eliminated and measure the compatibility of the unlearning model with the retrained model.
unlearning time and retraining time	Quantify the time saved by using unlearning rather than retraining for model updating.
model inversion attack	Use model inversion attacks to determine whether the model retains information about forgotten samples.
membership inference attack	A membership inference attack is used to determine whether the model retains information about the forgotten samples.

learning algorithms are unable to solve this problem, then federated learning is needed to solve the problem.

In federated learning, machine unlearning cannot be simply migrated because the global weights are derived by aggregation rather than by raw gradient computation. In addition, these clients may have some overlapping data, making it difficult to quantify the impact of each training item on the model weights. The use of classical gradient-operated machine unlearning methods can even lead to serious accuracy degradation and new privacy threats. How to implement machine unlearning in federated learning models has become the focus of current research, and domestic and international researchers have made good progress on different issues:

For one thing, current research on machine unlearning in federated learning usually assumes that the data to be deleted belongs exclusively to one client. In this way, the historical contributions of a particular client to the training of the global model can be easily recorded and deleted. However, deleting historical parameter updates may still corrupt the global model, but there are many strategies to overcome this problem. For example, Liu et al. Liu, Ma, Yang, Wang and Liu (2021) propose a calibrated training method to separate individual client contributions as much as possible. This mechanism is not applicable to deep neural networks, but it is suitable for shallow architectures such as 2-layer CNNs or networks with two fully connected layers.

Second, given the cost of storing historical information on federated servers, machine unlearning research also requires a trade-off between model scalability and accuracy. Wu et al. Wu, Zhu and Mitra proposed a knowledge distillation strategy that uses a primary global model to train an initial model on the remaining data. However, since the server does not have access to the client's data, some unlabelled data that follows the distribution of the whole dataset needs to be sampled and additional information needs to be exchanged between the client and the server. Therefore, the communication cost of the whole process is very high. In addition, further bias may occur when the data are not independently and identically distributed Liu, Xu, Yuan, Wang and Li (2022b).

Third, Liu et al. Liu et al. (2022b) proposed an intelligent retraining method without communication protocols in the study of federated machine unlearning. This method uses the L-BFGS algorithm Berahas, Nocedal and Takác (2016); Bollapragada, Nocedal, Mudigere, Shi and Tang (2018) to efficiently solve the Hessian approximation with historical parameter updates for global model retraining. However, this method is only applicable to small models (parameters $\leq$ 10K). In addition, it involves storing snapshots of old models (including historical gradients and parameters), which poses some privacy threats.

6.2. Machine unlearning in lifelong learning

Lifelong learning, also known as continuous learning, increment learning, never ending learning. Usually in machine learning, a single model solves only a single or a few tasks. For new tasks, we usually retrain new models. Lifelong learning, on the other hand, uses a model on task 1, then continues to use it on task 2, and so on until task n. Lifelong learning explores the question of whether a model can perform well on many tasks. As it goes on, the model becomes more and more capable. This is similar to the way that humans learn new things all the time, and thus acquire a lot of different knowledge. Research on machine unlearning has also contributed significantly to the development of lifelong learning:

For one, machine unlearning has been developed for many years as a study to combat catastrophic unlearning in deep neural networks Du et al. (2019); Liu, Liu and Stone (2022a). catastrophic unlearning is a phenomenon in which deep neural networks perform poorly after learning too many tasks Kirkpatrick, Pascanu, Rabinowitz, Veness, Desjardins, Rusu, Milan, Quan, Ramalho, Grabska-Barwinska et al. (2017). A simple solution to this problem is to retrain the model on historical data. Obviously, this solution is impractical, not only because of the high computational cost, but also because there is no guarantee that the model will converge or that catastrophic unlearning will not occur again Parisi, Kemker, Part, Kanan and Wermter (2019). Therefore, Du et al. proposed a solution based on unlearning Du et al. (2019) to prevent catastrophic unlearning. The core idea is to forget harmful samples (e.g., false-negatives/false-positives in case

samples), and then update the model so that its performance remains unchanged before the catastrophic unlearning effect. Secondly, Machine unlearning is also used to deal with Explosive Loss in Machine Learning. In machine learning, "explosion loss" refers to the phenomenon that the value of the loss function increases dramatically when training neural networks because the gradient value becomes very large. This usually occurs in deep neural networks, where the gradient value becomes too large and the update of the network parameters becomes too large, resulting in a rapid increase in the value of the loss function. The explosion loss will cause the network to fail to converge and the training process will fail. Therefore, Du Du et al. (2019) et al. proposed a unlearning method to alleviate this problem by using the unlearning loss to regulate the extreme cases.
Thirdly, unlearning has also been investigated in other lifelong learning scenarios using incremental models, such as decision trees and plain Bayes, which allow the model to dynamically unlearn samples of data. Liu et al. considered the unlearning requests of lifelong models for a particular task Liu et al. (2022a), and in particular the three types of requests in lifelong learning: (i) learning a task permanently; (ii) learning a task temporarily and then unlearning it when privacy is requested; and (iii) unlearning a task. (i) learning a task permanently; (ii) learning a task temporarily and then unlearning it when privacy is requested; and (iii) unlearning a task. Unlike traditional machine unlearning, unlearning in lifelong learning not only maintains the transfer of knowledge between tasks, but also retains all knowledge for the remaining tasks. In addition, the training process is extremely difficult because the scenarios depend on the order of the tasks, which are learnt online during the lifetime of the model. In addition, the model does not retain all previous data (zero-glance unlearning), which makes the unlearning process even more challenging. inspired by SISA (Smart Intelligent Retraining Bourtoule et al. (2021)), Liu et al. proposed a solution: create an isolated temporary model for each task and merge the isolated model into the main model.

7. Discussion and future prospects

The research on machine unlearning is still in the ascendant, and has made good progress, and there is still a potential development trend. However, the existing machine unlearning algorithms still have many deficiencies in terms of technology, balance, privacy protection cost and practical applications. By identifying these problems and challenges, we can look forward to the opportunities and directions for the future development of machine unlearning research.

7.1. Summary and trends

We analyse some of the current achievements and potential trends in machine unlearning Nguyen, Huynh, Nguyen, Liew, Yin and Nguyen (2022b) and summarise our findings: (1) Influence functions are the primary method. Understanding the impact of a given data item on model parameters or model performance is the key to machine unlearning. By simply reversing the model updates associated with the target data, this approach will greatly accelerate the unlearning process. Although there may be some bias in doing so, it has been shown that this bias is bounded Chundawat, Tarun, Mandal and Kankanhalli (2023).
(2) Accessibility of model parameters needs to be considered. Existing research defines "learning by forgetting" as obtaining a new model that is as accurate as a retrained model without forgetting the data Golatkar et al. (2020). We believe that this parameter space assumption should be carefully considered. Given that model parameters can be determined with or without data, is there a case where the original model and the unlearning model share the same parameters ? Although some studies have used parameter distributions to constrain the problem Guo et al. (2019), some of the effects of forgetting the data may still be present in the unlearning model.
(3) Machine unlearning validation is required. Validation of unlearning is the process of determining whether particular data have been eliminated from the model. Independent validation of the effects of unlearning is necessary to fully realise the right to regulate forgetting. There is little research on validation of unlearning Gao, Ma, Wang, Sun, Li, Ji, Cheng and Chen (2022). However, the definition of successful validation remains controversial because different unlearning solutions use different evaluation metrics, especially when the cut-off threshold of the validation metrics still depends on the domain of application Thudi et al. (2022b).
(4) Federated unlearning is emerging. Federated unlearning provides a unique environment for machine unlearning research Liu et al. (2021). It has independent clients involved in the federated training process, and the user data on a client mainly contributes to making correct predictions about that user. As a result, clients can be accurately removed from the federation using historical updates. This localisation helps to avoid the catastrophic unlearning that occurs in traditional machine unlearning environments. However, in many cases, the data for federated unlearning is not independently and homogeneously distributed, and sometimes the deletion request covers only a portion of the client data.
(5) Machine unlearning can be used to repair models. Machine learning models can be poisoned by adversarial attacks. Intuitively, if toxic data is detected and eliminated, and then the model is retrained, the new model should be non-toxic. However, the cost of retraining is prohibitive. This is indeed similar to a forgetful learning environment. In contrast to existing defence methods, machine learning models determine and update internal problem weights via an influence function. A similar application is the removal of model bias due to certain biased features in the data. Current research on fairness and debiased learning focuses on learning a fair and unbiased feature representation Nam, Cha, Ahn, Lee and Shin (2020), where machine unlearning, e.g., feature unlearning Guo et al. (2022), allows for correctly removing biased features while maintaining model quality.

7.2. Open research questions

We have summarised some of the open research questions that we hope future research will address. This section lists some of them and discusses the underlying themes they address in machine unlearning research.

(1) Harmonisation of design requirements. None of the current unlearning methods can satisfy all the design requirements. Not only do we need to focus on approximate unlearning scenarios and data item removal, but we also need to consider deletion requests for feature deletion, class deletion, task deletion, stream deletion, and other types of deletion. In addition, meeting all design requirements - completeness, timeliness, accuracy, etc. - will make unlearning solutions more suitable for industrial grade systems.

(2) Uniform benchmarking. Although there are many recent studies on machine unlearning, they do not have a common standard for benchmarking. Schelter et al. conducted an empirical study Schelter (2020), but their benchmark is limited to incremental learning methods, focusing only on efficiency. There is still a long way to go to standardise benchmarking.

(3) Adversarial machine unlearning. In order to better understand and protect our machine learning systems, we need to do more research on attacks against them. Adversarial machine unlearning is the study of attacks on unlearning algorithms to better understand unlearning models. Unlike using machine unlearning to mitigate adversarial attacks, adversarial machine unlearning is much more rigorous, not only in terms of accuracy, but also in terms of privacy protection.

(4) Interpretable machine unlearning. In the future, explanations of machine unlearning could be used to increase confidence in human-AI interactions and enable verification of unlearning. However, the inverse nature of machine unlearning may pose problems in terms of the applicability of explanatory methods. Designing techniques to explain the process of unlearning remains an unfinished task.

(5) Oblivious machine learning in evolutionary data streams. Evolving data streams pose problems for machine learning models, especially neural networks, due to changes in data distribution and model predictions. Some experiments have found that machine unlearning can be useful in repairing outdated models by forgetting old data that contradicts the detected conceptual drift Hu, Tang, Miao, Hua and Zhang (2021). However, research needs to consider the problem of analysing old and new data in a contradictory way.

8. Summary

The core idea of machine unlearning is to enable machine learning models to selectively "forget" some information, especially data that is no longer relevant or may lead to bias. The emergence of machine unlearning has brought a great opportunity to solve the problems of privacy protection and system usability in the field of machine learning.

In this paper, we make a thorough research and in-depth analysis on the latest research on machine unlearning, introduce the definition and framework of machine unlearning, analyse the challenges, and summarise the main types of algorithms. It summarises the main types of algorithms, summarises the applications by fields, and introduces the latest achievements in each field, so that the development of machine unlearning is well organised. Finally, we discuss the future development of machine unlearning based on the existing research. Although there have been many studies on machine unlearning, there are still many problems that need to be solved, such as inconsistent design requirements and incomplete benchmarking, which are still far from reaching a mature level. Machine unlearning is still waiting for more extensive and in-depth research.

9. Acknowledgement

This research was supported by the National Natural Science Foundation of China under Grants 62102035 and the National Key Research and Development Program of China [Grant No. 2022ZD0115901].

References

Baumhauer, T., Schöttle, P., Zeppelzauer, M., 2022. Machine unlearning: Linear filtration for logit-based classifiers. Machine Learning 111, 3203–3226.

Berahas, A.S., Nocedal, J., Takác, M., 2016. A multi-batch l-bfgs method for machine learning. Advances in Neural Information Processing Systems 29.

Bollapragada, R., Nocedal, J., Mudigere, D., Shi, H.J., Tang, P.T.P., 2018. A progressive batching l-bfgs method for machine learning, in: International Conference on Machine Learning, PMLR. pp. 620–629.

Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C.A., Jia, H., Travers, A., Zhang, B., Lie, D., Papernot, N., 2021. Machine unlearning, in: 2021 IEEE Symposium on Security and Privacy (SP), IEEE. pp. 141–159.

Brophy, J., Lowd, D., 2021. Machine unlearning for random forests, in: International Conference on Machine Learning, PMLR. pp. 1092–1104.

Cao, Y., Yang, J., 2015. Towards making systems forget with machine unlearning, in: 2015 IEEE symposium on security and privacy, IEEE. pp. 463–480.

Cao, Z., Wang, J., Si, S., Huang, Z., Xiao, J., 2022. Machine unlearning method based on projection residual, in: 2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA), IEEE. pp. 1–8.

Chen, K., Huang, Y., Wang, Y., 2021a. Machine unlearning via gan. arXiv preprint arXiv:2111.11869 .

Chen, M., Zhang, Z., Wang, T., Backes, M., Humbert, M., Zhang, Y., 2021b. When machine unlearning jeopardizes privacy, in: Proceedings of the 2021 ACM SIGSAC conference on computer and communications security, pp. 896–911.

Chundawat, V.S., Tarun, A.K., Mandal, M., Kankanhalli, M., 2023. Zero-shot machine unlearning. IEEE Transactions on Information Forensics and Security .

Du, M., Chen, Z., Liu, C., Oak, R., Song, D., 2019. Lifelong anomaly detection through unlearning, in: Proceedings of the 2019 ACM SIGSAC conference on computer and communications security, pp. 1283–1297.

Gao, X., Ma, X., Wang, J., Sun, Y., Li, B., Ji, S., Cheng, P., Chen, J., 2022. Verifi: Towards verifiable federated unlearning. arXiv preprint arXiv:2205.12709 .

Garg, S., Goldwasser, S., Vasudevan, P.N., 2020. Formalizing data deletion in the context of the right to be forgotten, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer. pp. 373–402.

Geurts, P., Ernst, D., Wehenkel, L., 2006. Extremely randomized trees. Machine learning 63, 3–42.

Ginart, A., Guan, M., Valiant, G., Zou, J.Y., 2019. Making ai forget you: Data deletion in machine learning. Advances in neural information processing systems 32.

Golatkar, A., Achille, A., Ravichandran, A., Polito, M., Soatto, S., 2021. Mixed-privacy forgetting in deep networks, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 792–801.

Golatkar, A., Achille, A., Soatto, S., 2020. Eternal sunshine of the spotless net: Selective forgetting in deep networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9304–9312.

Guo, C., Goldstein, T., Hannun, A., Van Der Maaten, L., 2019. Certified data removal from machine learning models. arXiv preprint arXiv:1911.03030 .

Guo, T., Guo, S., Zhang, J., Xu, W., Wang, J., 2022. Efficient attribute unlearning: Towards selective removal of input attributes from feature representations. arXiv preprint arXiv:2202.13295 .

Guo, Y., Zhao, Y., Hou, S., Wang, C., Jia, X., 2023. Verifying in the dark: Verifiable machine unlearning by using invisible backdoor triggers. IEEE Transactions on Information Forensics and Security .

He, Y., Meng, G., Chen, K., He, J., Hu, X., 2021. Deepobliviate: a powerful charm for erasing data residual memory in deep neural networks. arXiv preprint arXiv:2105.06209 .

Hu, X., Tang, K., Miao, C., Hua, X.S., Zhang, H., 2021. Distilling causal effect of data in class-incremental learning, in: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition, pp. 3957–3966.

Huang, H., Ma, X., Erfani, S.M., Bailey, J., Wang, Y., 2021. Unlearnable examples: Making personal data unexploitable. arXiv preprint arXiv:2101.04898 .

Izzo, Z., Anne Smart, M., Chaudhuri, K., Zou, J., 2021. Approximate data deletion from machine learning models, in: Banerjee, A., Fukumizu, K. (Eds.), Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, PMLR. pp. 2008–2016.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al., 2017. Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences 114, 3521–3526.

Liu, B., Liu, Q., Stone, P., 2022a. Continual learning and private unlearning, in: Conference on Lifelong Learning Agents, PMLR. pp. 243–254.

Liu, G., Ma, X., Yang, Y., Wang, C., Liu, J., 2021. Federaser: Enabling efficient client-level data removal from federated learning models, in: 2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS), IEEE. pp. 1–10.

Liu, Y., Xu, L., Yuan, X., Wang, C., Li, B., 2022b. The right to be forgotten in federated learning: An efficient realization with rapid retraining, in: IEEE INFOCOM 2022-IEEE Conference on Computer Communications, IEEE. pp. 1749–1758.

Marchant, N.G., Rubinstein, B.I., Alfeld, S., 2022. Hard to forget: Poisoning attacks on certified machine unlearning, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 7691–7700.

Masi, I., Wu, Y., Hassner, T., Natarajan, P., 2018. Deep face recognition: A survey, in: 2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI), IEEE. pp. 471–478.

Nam, J., Cha, H., Ahn, S., Lee, J., Shin, J., 2020. Learning from failure: Debiasing classifier from biased classifier. Advances in Neural Information Processing Systems 33, 20673–20684.

Neel, S., Roth, A., Sharifi-Malvajerdi, S., 2021. Descent-to-delete: Gradient-based methods for machine unlearning, in: Algorithmic Learning Theory, PMLR. pp. 931–962.

Nguyen, Q.P., Low, B.K.H., Jaillet, P., 2020. Variational bayesian unlearning. Advances in Neural Information Processing Systems 33, 16025–16036.

Nguyen, Q.P., Oikawa, R., Divakaran, D.M., Chan, M.C., Low, B.K.H., 2022a. Markov chain monte carlo-based machine unlearning: Unlearning what needs to be forgotten, in: Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security, pp. 351–363.

Nguyen, T.T., Huynh, T.T., Nguyen, P.L., Liew, A.W.C., Yin, H., Nguyen, Q.V.H., 2022b. A survey of machine unlearning. arXiv preprint arXiv:2209.02299 .

Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S., 2019. Continual lifelong learning with neural networks: A review. Neural networks 113, 54–71.

Schelter, S., 2020. Amnesia-a selection of machine learning models that can forget user data very fast. suicide 8364, 46992.

Schelter, S., Grafberger, S., Dunning, T., 2021. Hedgecut: Maintaining randomised trees for low-latency machine unlearning, in: Proceedings of the 2021 International Conference on Management of Data, pp. 1545–1557.

Sekhari, A., Acharya, J., Kamath, G., Suresh, A.T., 2021. Remember what you want to forget: Algorithms for machine unlearning. Advances in Neural Information Processing Systems 34, 18075–18086.

Shokri, R., Stronati, M., Song, C., Shmatikov, V., 2017. Membership inference attacks against machine learning models, in: 2017 IEEE symposium on security and privacy (SP), IEEE. pp. 3–18.

Singh, A., Anand, A., 2017. Data leakage detection using cloud computing. International Journal Of Engineering And Computer Science 6.

Sommer, D.M., Song, L., Wagh, S., Mittal, P., 2020. Towards probabilistic verification of machine unlearning. arXiv preprint arXiv:2003.04247 .

Sommer, D.M., Song, L., Wagh, S., Mittal, P., 2022. Athena: Probabilistic verification of machine unlearning. Proc. Privacy Enhancing Technol 3, 268–290.

Tanha, J., Abdi, Y., Samadi, N., Razzaghi, N., Asadpour, M., 2020. Boosting methods for multi-class imbalanced data classification: an experimental review. Journal of Big Data 7, 1–47.

Tarun, A.K., Chundawat, V.S., Mandal, M., Kankanhalli, M., 2023. Fast yet effective machine unlearning. IEEE Transactions on Neural Networks and Learning Systems .

Thudi, A., Deza, G., Chandrasekaran, V., Papernot, N., 2022a. Unrolling sgd: Understanding factors influencing machine unlearning, in: 2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P), IEEE. pp. 303–319.

Thudi, A., Jia, H., Shumailov, I., Papernot, N., 2022b. On the necessity of auditable algorithmic definitions for machine unlearning, in: 31st USENIX Security Symposium (USENIX Security 22), pp. 4007–4022.

Thudi, A., Shumailov, I., Boenisch, F., Papernot, N., 2022c. Bounding membership inference. arXiv preprint arXiv:2202.12232 .

Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., Zhao, B.Y., 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks, in: 2019 IEEE Symposium on Security and Privacy (SP), IEEE. pp. 707–723.

Wang, J., Guo, S., Xie, X., Qi, H., 2022. Federated unlearning via class-discriminative pruning, in: Proceedings of the ACM Web Conference 2022, pp. 622–632.

Wang, R., Li, Y.F., Wang, X., Tang, H., Zhou, X., 2009. Learning your identity and disease from research papers: information leaks in genome wide association study, in: Proceedings of the 16th ACM conference on Computer and communications security, pp. 534–544.

Warnecke, A., Pirch, L., Wressnegger, C., Rieck, K., 2021. Machine unlearning of features and labels. arXiv preprint arXiv:2108.11577 .

Wu, C., Zhu, S., Mitra, P., . Federated unlearning with knowledge distillation. arxiv 2022. arXiv preprint arXiv:2201.09441 .

Wu, G., Hashemi, M., Srinivasa, C., 2022. Puma: Performance unchanged model augmentation for training data removal, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 8675–8682.

Declaration of interests

☐ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The author is an Editorial Board Member/Editor-in-Chief/Associate Editor/Guest Editor for [Journal name] and was not involved in the editorial review or the decision to publish this article.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: