SN Computer Science (2023) 4:337
https://doi.org/10.1007/s42979-023-01767-4

SURVEY ARTICLE

A Review on Machine Unlearning

Haibo Zhang[1] · Toru Nakamura[2] · Takamasa Isohara[2] · Kouichi Sakurai[3]

Received: 13 March 2022 / Accepted: 4 March 2023 / Published online: 19 April 2023

Abstract

Recently, an increasing number of laws have governed the useability of users' privacy. For example, Article 17 of the General Data Protection Regulation (GDPR), *the right to be forgotten*, requires machine learning applications to remove a portion of data from a dataset and retrain it if the user makes such a request. Furthermore, from the security perspective, training data for machine learning models, i.e., data that may contain user privacy, should be effectively protected, including appropriate erasure. Therefore, researchers propose various privacy-preserving methods to deal with such issues as machine unlearning. This paper provides an in-depth review of the security and privacy concerns in machine learning models. First, we present how machine learning can use users' private data in daily life and the role that the GDPR plays in this problem. Then, we introduce the concept of machine unlearning by describing the security threats in machine learning models and how to protect users' privacy from being violated using machine learning platforms. As the core content of the paper, we introduce and analyze current machine unlearning approaches and several representative results and discuss them in the context of the data lineage. Furthermore, we also discuss the future research challenges in this field.

Keywords Machine learning · Security · Privacy · Machine unlearning · Data lineage

Introduction and Background

Privacy protection has been a concern for researchers for a long time. In today's big data environment, users interact with data on various web platforms, such as sending and receiving emails, and browsing news, almost every day. For users, once they have provided their information in an application, it is difficult to remove it from the root. When machine learning is widely used today, most advanced features are obtained based on understanding and training data. As a result, users' privacy has been spread in every corner of the application, makings it more accessible for attackers to steal users' private data.

From the security perspective, if an attacker compromises the machine learning model by injecting some pollution data into its dataset, it is also necessary to remove such data from the dataset and retrain it [1]. For example, an attacker can open a backdoor in a machine learning model by injecting malicious data into the dataset used for training [2]. As a result, the attacker can steal all the private data in the model, as shown in Fig. 1.

For solving the above problems, it is necessary to retrain the machine learning model. However, the existing retraining methods cause a large amount of computational power and time consumption. Therefore, researchers propose machine unlearning as a more efficient research method [3].

The word "unlearning" means that the machine learning model is retrained to generate a new predictive model with a portion of the data forgotten. There are two ways to perform unlearning on machine learning models. One is to retrain

Toru Nakamura, Takamasa Isohara, and Kouichi Sakurai have contributed equally to this work.

✉ Haibo Zhang
zhang.haibo.892@s.kyushu-u.ac.jp

Toru Nakamura
tr-nakamura@kddi-research.jp

Takamasa Isohara
ta-isohara@kddi-research.jp

Kouichi Sakurai
sakurai@inf.kyushu-u.ac.jp

[1] Department of Information Science and Technology, Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan

[2] KDDI Research Inc., Fujimino 356-8502, Japan

[3] Department of Information Science and Technology, Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan

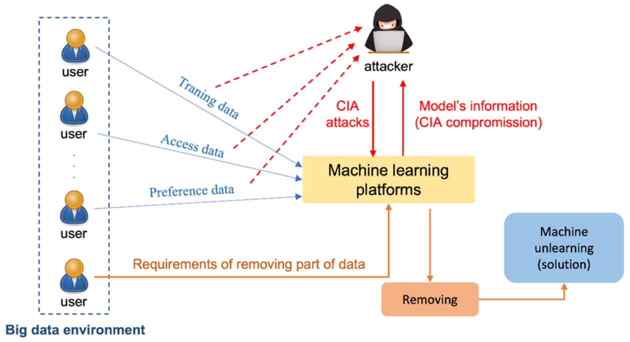

user
user
user
user
attacker
Traning data
Access data
Preference data
CIA attacks
Model's information (CIA compromission)
Machine learning platforms
Machine unlearning (solution)
Requirements of removing part of data
Removing
Big data environment

Fig. 1 The necessity of machine unlearning. The red arrow indicates that the attacker can access the training data or parameters of the machine learning model through malicious data injection or information stealing to obtain user privacy or even reconstruct the machine learning model. In this case, according to the orange arrow, the data owner will request to delete specific sensitive data, and the model owner needs to apply machine unlearning methods to remove the requested data

the new dataset from scratch after data removal (i.e., exact unlearning mentioned in "Exact Unlearning" section. The other is to modify the machine learning model and dataset to achieve an approximate unlearning effect (i.e., approximate unlearning mentioned in "Approximate Unlearning" section). The ultimate goal of either unlearning approach is to improve the accuracy of unlearning methods while being as efficient as possible.

This paper provides an in-depth analysis of machine learning models' security and privacy concerns, which also refers to the privacy-preserving machine learning [4]. This paper aims to provide a comprehensive analysis and summary of current machine unlearning techniques and future research potential.

First, we present how machine learning can use users' private data in daily life and the role that the GDPR plays in this problem. Then, we introduce the concept of machine unlearning by describing the security threats in machine learning models and how to protect users' privacy from being violated using machine learning platforms. In the next section, we introduce and analyze current machine unlearning approaches and several representative research results and discuss them in the context of the data lineage. Furthermore, we also discuss the future research challenges in this field.

How Can Machine Learning Use Users' Data?

Since the idea of simulating human intelligence was first proposed in the 1960s, artificial intelligence (AI) received widespread attention in both academic and industry fields. As the primary component of AI, machine learning also gained unprecedented development in recent years. Moreover, its application has spread to various fields of AI. For example, we can use machine learning to classify and locate objects in the field of computer vision, and we can also use deep neural networks to design and implement a high-accuracy face recognition system. In addition, we can also use machine learning in natural language processing to design and implement an intelligent question and answer system. In the modern Big Data environment, Internet users interact with various applications almost every day. Enterprises and developers use data mining, big data analytics, and machine learning' techniques to extract useful information from the vast database. These data contain more or less sensitive information about users, such as their identity and passwords. Hence, machine learning plays an important role.

Machine learning is a branch of artificial intelligence that automatically enables computers to learn from experience through human intervention. The whole concept of machine learning starts around determining the answer to an obstacle without human interference, which begins with understanding data from examples or direct experience, analyzing data patterns and making better decisions based on inferences. It is best used for problem-solving when large amounts of data and variables exist without using the existing algorithms. For example, Google tends to optimize search results and pop-up ads for products similar to users' tastes or websites they have visited before. It studies the user's behavior and displays the results accordingly.

Machine learning is an integral part of big data analytics. Big data analytics includes big data, data learning, statistical information, etc. Machine learning uses programming and computational algorithms to conclude, while big data analytics uses numbers and statistics to draw results.

The General Data Protection Regulation

Recently, an increasing number of laws have governed the usability of users' privacy. For example, Article 17 of the General Data Protection Regulation (GDPR), *the right to be forgotten*, requires entities to remove a portion of data from a dataset if the user makes such a request [5, 6]. Furthermore, it maintains the user's right to use their privacy from a privacy protection perspective [4]. The GDPR is a new EU privacy and data protection regulation. It requires more granular privacy protections in company systems, more detailed data protection agreements, and more user-friendly and detailed disclosures about company privacy and data protection practices.

The GDPR has direct legal implications for all EU member states, i.e., it is binding without having to be transposed into the national laws of EU member states. This will enhance the consistency and harmonization of the implementation of EU law.

From its initial draft in 2012 to becoming official EU law in 2016, *the right to be forgotten* was initially intended to bind Internet search engines, such as Google and Yahoo, in their use of users' privacy. Under Article 17, if a user requests the deletion of any private data, the search engine shall immediately execute and is not allowed to refuse. However, implementing this law also raises considerations about the current hot topic, machine learning technology, and overusing users' private data. For example, how should machine learning platforms respond if data holders request to delete specific data used for training purposes?

In the context of machine learning, *the right to be forgotten* requires the machine learning applications to be able to readily accommodate requests from data owners who wish to delete any data [7]. This process is called machine unlearning. The machine learning application needs to remove the requested data from the training dataset and retrain the machine learning model from scratch.

The appearance of Article 17 has primarily limited the undesirable phenomenon of misused and unprotected user privacy in the current fast-growing Internet and big data environment. However, privacy protection should be carried out from both the perspective of the data controller and the data holder.

The right to be forgotten can be regulated from the perspective of data controllers, but it does not work from the perspective of data holders. That is, how data holders become aware of the violation of their privacy and when they request the deletion of their private data. These cannot be regulated by the regulation and require data holders to raise their awareness of privacy protection under the guidance of social engineering.

Security Concerns

Machine Learning is Still Weak

In the era of big data and artificial intelligence, people can access information more quickly and efficiently. However, while gaining convenience, our behavior is being recorded, learned and used all the time. If we ignore privacy protection in the application, it will be challenging to prevent personal information from being used for illegal purposes.

Due to the vulnerability problem of machine learning models themselves, attackers can attack machine learning models by sending many malicious requests, exposing machine learning services to various potential security risks [8].

- Data privacy leakage risk: attackers can exploit the model vulnerability to obtain data information for training models by invoking machine learning services.
- Model theft risk: model parameter information in machine learning services due to the model's vulnerability issues, making it risky for attackers to speculate and restore model parameter information by frequently invoking the service.
- Data poisoning: the attacker can mix specific malicious data in the request process, which can affect the model training and subsequent model inference through the feedback of the service process to achieve the effect of interfering with the model [9, 10].
- Evasion: attackers can make machine learning services make the wrong judgments by adding a small amount of noise and perturbation to typical requests.

Usually, when designing machine learning systems, developers consider specific threat models to ensure that the designed system is secure and trustworthy. So far, most of the existing machine learning models have been designed and implemented for a fragile threat model without much consideration of the attackers [11]. Although these models can perform very well in the face of natural inputs, in a realistic setting, these machine learning models encounter many malicious users and even attackers.

Toreini et al. [12] provide a systematic approach to relate considerations about trust from the social sciences to trustworthiness technologies proposed for AI-based services and products. For example, attackers have different degrees of ability to maliciously modify the inputs and outputs during the model's training and prediction phases. Even they can access the internal structure of the model by some means and steal the parameters, thus destroying the confidentiality, integrity and usability of the models.

CIA Triad in Machine Learning

The CIA triad is a common assessment model that forms the basis for developing security systems and policies. The CIA

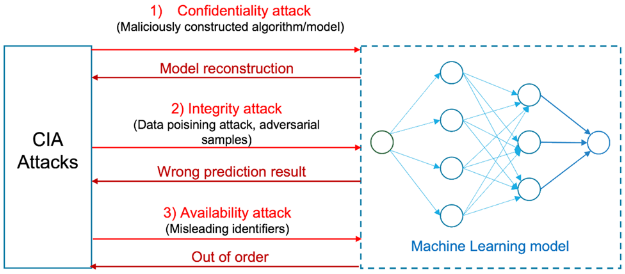


Fig. 2 CIA triad in machine learning

refers to confidentiality, integrity, and availability. The CIA triad identifies system vulnerabilities and methods to address problems and create effective solutions.

Attacks against machine learning models can impact Confidentiality, Integrity, and Availability [13]. Figure 2 describes how the CIA triad can be applied to the machine learning model.

- **Confidentiality attacks** mean that machine learning systems must ensure that unauthorized users do not have access to the information. While most machine learning platforms are professional and secure, the algorithms provided by machine learning model providers are not necessarily reliable [14]. When data holders use MLaaS (Machine Learning as a Service) to train their predictive models, they may select a malicious model carefully constructed by an attacker. In such models, the attacker encodes the data holder's private data into the parameters of the model and finally steals the user's private data by decoding the parameters of the model [15].
- Machine learning models are most vulnerable to **integrity attacks**, occurring both in the learning and prediction phases. If the attacker disrupts the model's integrity, then the model's prediction results will deviate from expectations. The attacker can modify the existing training set or add additional malicious data to compromise the integrity of the model to reduce the accuracy in the prediction phase [16]. When the model is trained and used for prediction, the attacker only needs to add a small perturbation to the sample to be predicted, which is unrecognizable to the human eye but sufficient to make the model classification wrong.
- The **availability** of machine learning models can also be a target of attack. For example, in a driverless scenario, if an attacker places something complicated to identify on the side of the road where a vehicle would pass, it could potentially force a self-driving car to go into safety protection mode and then stop at the side of the road.

Privacy-Preserving

In recent years, more and more people have started to pay attention to data privacy and pay more attention to privacy terms when choosing to use client software (apps). Some studies have shown that the protection of privacy can increase the usage rate of users [17].

As research evolves, machine learning models become more powerful and require more training data. For example, some training models in the industry need to use hundreds of gigabytes of data to train billions of parameters. Unfortunately, in many professional fields such as healthcare and financial fraud prevention, data are divided into silos due to privacy or interests, making machine learning face the problem of insufficient valid data. Therefore, information flow and machine learning cannot be achieved without providing guarantees for data privacy.

Privacy-preserving approaches in machine learning can be divided into confidential computing, model privacy, and distributed learning [4].

- **Confidential computing** means that the transmission of data and the computation process is confidential. Current approaches to achieving confidential computing include Multi-party Secure Computation, Homomorphic Encryption, and Trusted Executive Environment. Confidential computing can be done to protect data privacy during the training process. So can the trained model cause the leakage of private training data? The answer is yes, because machine learning models are overfitted to some extent. The models themselves remember part of the training data, leading to private training data leakage by the published models.
- For **model privacy**, this includes **differential private machine learning** and **machine unlearning** algorithms. A common practice to achieve differential privacy is to add noise. Adding noise entails performance loss of the model, and differential privacy machine learning studies how to add noise more economically and how to add the least amount of noise to achieve the best performance for a given privacy loss requirement. Another hot topic of model privacy research is machine unlearning. If implementing differential privacy is viewed as actively designing algorithms to make the output model satisfy the privacy requirements, then machine forgetting is a passive solution to model privacy. It aims to implement the user's "the right to be forgotten" in machine learning models.
- The vision of **federated learning** is to perform multi-party federated machine learning without sharing data, which is essentially a distributed machine learning framework with restricted data access. Compared to classical distributed machine learning, the first layer of constraints in federated learning is data isolation—data are not shared across endpoints, are not balanced, and interactive communication is kept to a minimum.

Research on privacy-preserving machine learning has never stopped. Among the many approaches, machine unlearning is emerging and closely related to machine learning algorithms themselves. Current studies on machine unlearning cover various research approaches, such as model privacy, differential privacy, and federation learning. It also demonstrates the importance of machine unlearning in studying privacy-preserving machine learning. Therefore, this paper focuses on machine unlearning as a privacy-preserving approach.

Machine Unlearning

In this section, we explain the definitions of machine learning and machine unlearning and introduce the two primary approaches of machine unlearning, i.e., **exact unlearning** and **approximate unlearning**.

Defining Machine Learning

Machine learning is a technique that makes judgments by predicting possible outcomes. The programmer designs an initial model, trains it on a specific data set, and continuously optimizes the parameters in the model based on the prediction results obtained, which eventually leads to a mature model. Figure 3 shows the general process of a machine learning system.

The task to be learned in machine learning can be defined in a space Z of the form $X \times Y$, where X is named the sample space and Y is named the output space [3]. Taking supervised machine learning as an example, in the image classification problem, for a given data set $\mathcal{D}$ of the input–output pairs $(x, y) \in X \times Y$, the learning aims to find a model function satisfying $F: X1 \rightarrow Y$ in a continuous optimization process.

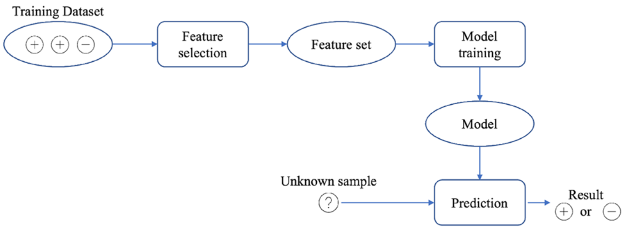


Fig. 3 The general machine learning system is consisting of three stages, i.e., feature selection, model training, and prediction

Machine Retraining vs. Machine Unlearning

The most intuitive approach to machine unlearning is to retrain the model on the training data set after deleting the specified data. However, this approach is computationally expensive, so the primary goal of machine unlearning is to reduce the computational cost. One approach is to post-process the trained model, so that the results of the machine unlearning algorithm are statistically indistinguishable from the retrained model [18]. Another approach is to design new training methods to reduce the cost of retraining. For instance, dividing the data into different blocks, training a separate sub-model for each block, and aggregating the results of the sub-models, so that only one sub-model needs to be retrained to remove a data point [3, 19].

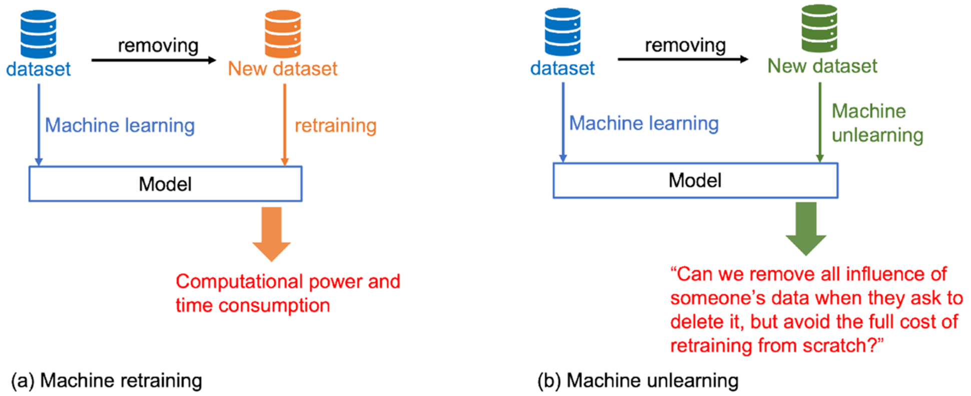
Figure 4 explains the difference between machine retraining and machine unlearning methods. As opposed to removing data from the data set and retraining the entire model, the purpose of machine unlearning is to minimize the cost of time and computational power associated with retraining.

Defining Machine Unlearning

The purpose of machine unlearning is that when the user requests to delete a part of the data, the model that has been learned needs to be retrained to generate a model distribution as if that part of the data had not been learned from the beginning.

The unlearning problem is defined as a kind of game between two parties, the service provider S, and the user population U by Bourtoule et al. [3]. The service provider S can be an organization that can collect various users' information and the collected information are stored in the form of a dataset $\mathcal{D}$. The service provider S uses these data to train and test a machine learning model M as a way to provide an intelligent service to the user U. Then, according to the GDPR, any user $u \in U$ has the right to request the removal of part of the data du from $\mathcal{D}$, and the service provider S must execute it. Thus, the service provider S must modify

Fig. 4 Machine retraining vs. machine unlearning


dataset
removing
New dataset
Machine learning
retraining
Model
Computational power and time consumption
dataset
removing
New dataset
Machine learning
Machine unlearning
Model
"Can we remove all influence of someone's data when they ask to delete it, but avoid the full cost of retraining from scratch?"

(a) Machine retraining

(b) Machine unlearning

the model M to generate a new model $M\neg du$, which represents a model without trained data *du*.

Guo et al. [21] propose a similar concept, *certified removal*, from an accuracy perspective. $\mathcal{D}$ is assumed to be a training dataset and A is the learning algorithm used to train $\mathcal{D}$, resulting in model $h \in H$, that is, $A : \mathcal{D} \rightarrow H$. When a request is made to remove sample x from $\mathcal{D}$, this results in a data removal mechanism M, one that can be applied to $A(\mathcal{D})$ and removes the effects of x. If the removal is successful, the output of M should be much close to the output of A applied on $\mathcal{D}\neg x$. Given $\epsilon > 0$, the removal mechanism M is said to perform *ϵ-certified removal* for learning algorithm A if $\forall T \subseteq H, D \subseteq X, x \in D$

$$e^{-\varepsilon} \leq \frac{P(M(A(D), D, x) \in T)}{P(A(D - x) \in T)} \leq e^{\varepsilon}.$$

The above definition states that the ratio between the likelihood of a model after the removal of sample x and a model that was never trained on sample x is close to one for all models, all possible data sets, and all removed samples.

However, some researchers have also proposed different views on defining machine unlearning. Thudi et al.[22] argue that machine unlearning should be divided into *exact unlearning* [23] and *approximate unlearning* [20]. Exact unlearning means that the model outputs after removing the sample x is the same as the one that was never trained on the removed sample x; approximate unlearning means that the model and dataset are adjusted, so that it does not need to be retrained from scratch. Current definitions of machine unlearning seek to make the output of approximate unlearning as close as possible to the output of exact unlearning. They suggest that this definition is incorrect, because the same model can be obtained even when trained on a different data set. Moreover, this definition only applies at the algorithmic level.

Figure 5 illustrates how machine unlearning algorithms can be applied to machine learning models and the essential difference between **exact unlearning** and **approximate unlearning** by defining a typical machine learning pipeline [20] with the three phases of model training, inference, and data unlearning. Finally, we summarize reviewed studies relatively to exact and approximate unlearning, as shown in Table 1.

Exact Unlearning

Exact unlearning [23] means that in the case of direct use of user data to build a machine learning model, such as a prediction task, a reasonable criterion is that the state of the system is adjusted to what it would be in the complete absence of user data.

Ullah et al. proposed an efficient machine unlearning algorithm, **total variation stability**, for the convex risk minimization problem, provided that the following three properties are satisfied.

- In the stream, at every time point, the output model should be indistinguishable from what we would have obtained if trained on the updated dataset.

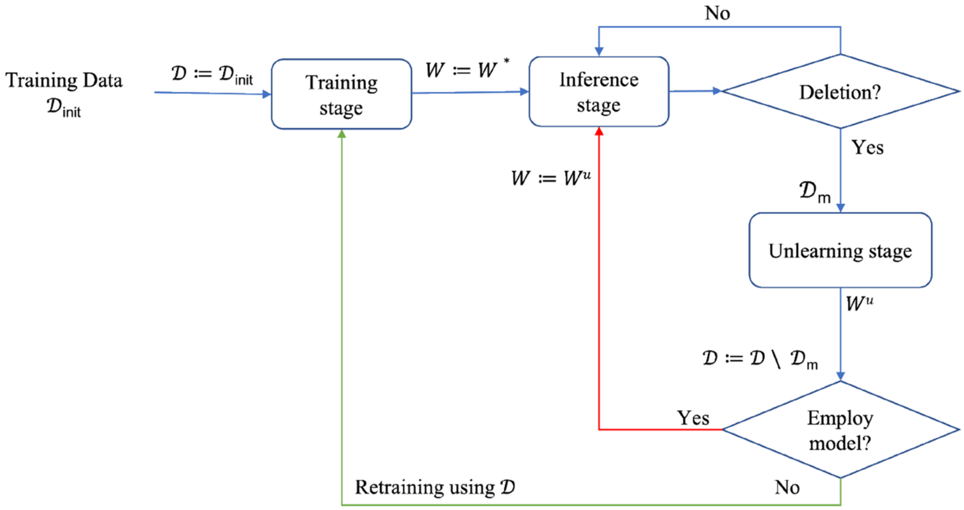

No
Training Data
$\mathcal{D}_{\text{init}}$
$\mathcal{D} \coloneqq \mathcal{D}_{\text{init}}$
Training stage
$W \coloneqq W^*$
Inference stage
Deletion?
Yes
$W \coloneqq W^u$
$\mathcal{D}_{\text{m}}$
Unlearning stage
W^u
$\mathcal{D} \coloneqq \mathcal{D} \setminus \mathcal{D}_{\text{m}}$
Yes
Employ model?
Retraining using $\mathcal{D}$
No

Fig. 5 A typical machine learning pipeline consists of three primary stages, i.e., training, inference, and unlearning. First, the initial model W^* is trained on the initial dataset $\mathcal{D}init$, and the output is used in the inference stage; afterward, once a request to delete the data $\mathcal{D}m$ is received, the updated model Wu can be obtained through the unlearning stage, when the data set becomes $\mathcal{D}\backslash\mathcal{D}m$. The process pointed by the red arrow is to apply the updated model Wu directly to the inference stage, i.e., approximate unlearning; the process pointed by the green arrow is to start retraining the initial model $W.^*$ on the new data set $\mathcal{D}\backslash\mathcal{D}m$ from scratch, i.e., exact unlearning [20]

Table 1 Summary of reviewed studies relatively to exact and approximate unlearning

Author	Year	Unlearning type		Approach	Ref.
		Exact unlearning	Approximate unlearning		
Cao & Yang	2015	✓		Summations following SQ learning	[24]
Cao et al.	2018	✓		Causal unlearning	[25]
Ullah et al.	2021	✓		Total variation stability	[23]
Kashef	2021	✓		Decremental unlearning	[26]
Schelter	2021	✓		Incremental maintenance	[5]
Jose & Simeone	2021	✓		Pac-Bayesian	[27]
Bourtoule	2021	✓		Federated learning (SISA)	[3]
Liu et al.	2021	✓		Federated unlearning	[28]
Brophy & Lowd	2021	✓		Random forests	[29]
Wu et al.	2022	✓	✓	Federated unlearning	[30]
Guo et al.	2019		✓	Newton method	[21]
Du et al.	2019		✓	Exploding loss and catastrophic forgetting	[31]
Baumhauer et al.	2020		✓	Linear filtration	[32]
Golatkar et al.	2020		✓	Differential privacy	[33]
Wu et al.	2020		✓	Rapid retraining by storing training data	[34]
Graves et al.	2020		✓	Amnesiac unlearning	[6]
Golatkar et al.	2021		✓	Mixed-privacy setting	[35]
Izzo et al.	2021		✓	Influence method	[36]
Neel et al.	2021		✓	Gradient-based method	[37]
Thudi et al.	2021		✓	Verification unlearn-error	[38]
Warnecke et al.	2021		✓	Parameters updates	[39]
He et al.	2021		✓	Intermediate models	[40]
Gong et al.	2021		✓	Particle-based Bayesian federated unlearning	[41]
Guo et al.	2022			Vertical unlearning	[42]

- The run time of unlearning method should be small.
- The output model should be effective in terms of accuracy.

There exist several exact unlearning approaches, for example, in support vector machines [26, 43–45], naïve Bayes [5, 24, 27], collaborative filtering, and ridge regression. This subsection will introduce and analyze several representative *exact unlearning* approaches.

Machine Unlearning's First Proposed

Cao and Yang first introduced the concept of machine unlearning in [24]. They present an unlearning method by transforming the model learning algorithm into a summation form that follows the statistical query (SQ) learning [46]. The unlearning method is performed by simply updating a small number of summations from the training dataset. The small number of summations is set in a layer between the machine learning algorithm and the model's training data to break down the dependencies. The learning algorithm only depends on the summations.

The authors implemented the unlearning method based on non-adaptive SQ learning (i.e., all SQs are determined upfront before the algorithm starts) and adaptive SQ learning (i.e., the later SQs may depend on the earlier SQ results). In this case, their summation forms can be implemented in many machine learning models and all stages.

SISA Training Approach

Bourtoule et al. [3] introduce the **SISA** training approach, short for Sharded, Isolated, Sliced, and Aggregated training. This framework expedites the unlearning process by strategically limiting the influence of a data point in the training procedure, which is illustrated in Fig. 6.

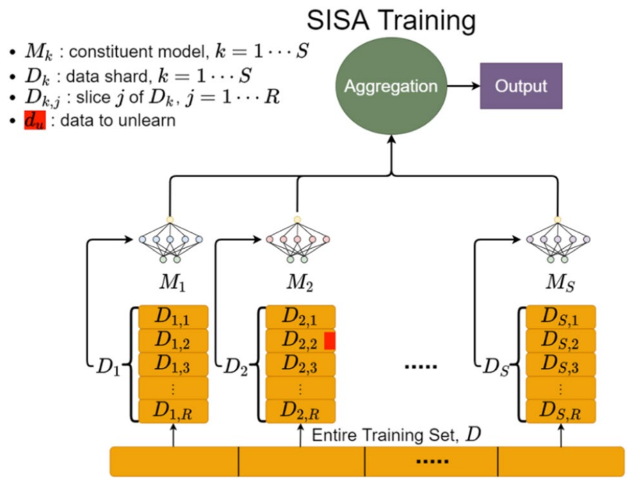


Fig. 6 The SISA approach is presented in the form of federated learning. In the SISA structure, separated machine learning models are trained on separated data blocks and the outputs are aggregated in the final inference stage [3]

In this model, the authors slice the original dataset $\mathcal{D}$ into s sub-datasets $\mathcal{D}1$ to $\mathcal{D}s$, and the machine learning network into s sub-networks $M1$ to Ms. Each sub-dataset $\mathcal{D}k$ is trained by the corresponding sub-network Mk, and the final training results are integrated by the aggregation algorithm. In this series, if a portion of data is requested to be deleted, it is only necessary to remove it from the sub-dataset and retrain it. Finally, the training results are reintegrated to obtain the new training results. This approach reduces the unnecessary data and model training process and dramatically reduces the time and computational power consumed by machine retraining.

In the paper, the authors illustrate that for simple learning tasks, the **SISA** training approach can accomplish the unlearning requests quickly without affecting the accuracy of the model. However, for complex learning tasks, the **SISA** training approach needs to be combined with other learning methods, such as transfer learning, to reduce the impact on model accuracy and to complete the unlearning requests quickly.

Approximate Unlearning

Approximate unlearning is a method for approximating the effect of model retraining by adjusting machine learning models and data sets. Mahadevan et al. [20] summarize that approximate unlearning methods can be roughly divided into three groups, and this subsection will introduce several methods on this basis.

First Group

The first group updates the machine learning model by retraining it with the remaining data and injecting optimal noise based on the principle of the **Fisher information matrix** [47] to control the certifiability.

Differential privacy [48, 49] can guarantee that the parameters of the trained model do not leak any individual information. Golatkar et al. [33, 50] proposed a method to selectively unlearn the dataset and update the machine learning models based on differential privacy methods. They propose a method for "scrubbing" the weights to remove specific training data used to train deep neural networks. This method does not require retraining from scratch or accessing the data initially used for training. Instead, this method modifies the weights of the model, so that any probe function of the weights approximates the same function as the weights of a network that has not been trained on these particular data.

In [35], Golatkar et al. introduce a new concept for machine unlearning, a mixed-privacy setting, based on their previous research. According to this method, a "core" subset of the training samples needs not be unlearned. Similar to [33, 50], this method allows to effectively remove all the information contained in the non-core data by simply setting a subset of the weights to zero with minimal performance loss. They demonstrate that this method yields significant improvements in unlearning accuracy and guarantees a large-scale vision classification task.

Second Group

The second group updates the machine learning model with the deleted data during the unlearning, they perform **Newton's method** [51] to estimate the impact of the deleted data on the model and remove it. The work [21] attempted approximate retraining by taking a single Newton's step. This can be formed as

$$\theta_{Newton} = \theta^{full} - \left[\nabla_{\theta}^{2} L^{\setminus k}\left(\theta^{full}\right)\right]^{-1} \nabla_{\theta} L^{\setminus k}\left(\theta^{full}\right),$$

where $\theta \in \mathbb{R}^d$ denotes the model parameters, k denotes the number of data points to be deleted from the model, $\theta^{full} = argmin\theta L^{full}(\theta)$ are the model parameters when fitted to the full dataset, $L^{\setminus k}(\theta)$ is the loss on the LKO dataset. When the loss function is quadratic, the approximation to $L^{\setminus k}$ is just $L^{\setminus k}$ itself which means that Newton's method is effective for solving this issue.

Izzo et al. [36] introduce the **Influence method** [51, 52] to estimate the influence of a particular training point on the model's predictions. The influence method can be formed under suitable assumptions on the loss function.l: $\theta(w) \equiv argmin_\theta \sum_{i=1}^{n} w_i l(x_i, y_i;\theta)$, where n denotes the total number of training points, $X = [x_i, \dots, x_n]^\top \in \mathbb{R}^{n\times d}$ is the data matrix for the full set of training data $\mathcal{D}^{full}$, $Y = [y_i, \dots, y_n]^\top \in \mathbb{R}^n$ is the response vector for $\mathcal{D}^{full}$, d denotes the data dimension. In this setting, $\theta^{full} = \theta(1)$ where 1 is the all 1 s vector and $\theta^{\backslash k} = \theta\big((\theta, \dots, 1, \dots)^\top\big)$. The influence function $kn - k$ approach uses the linear approximation [51, 52]

$$\theta_{inf} = \theta^{full} - \left[\nabla_\theta^2 L^{full}\left(\theta^{full}\right)\right]^{-1} \nabla_\theta L^{\backslash k}\left(\theta^{full}\right).$$

To $\theta(w)$, about $w = 1$ to estimate $\theta^{\backslash k}$. Therefore, they propose an unlearn method based on the influence method principle that the computational cost is linearly related to the feature dimension d, i.e., $O(d^2)$, and is independent of the number of training data n. And this method is applicable to both linear regression and logistic regression models.

The influence method explains the principle of machine unlearning at a higher level. In other words, the influence method-based unlearning can compute the impact of the deleted data relative to the parameters of the trained model for removing the influence and updating the parameters.

Third Group

The third group stores the data and related information during the machine learning model training and uses them to update the model when a request to delete the data is made [34, 37].

Graves et al. [6] proposed the concept of **Amnesiac Unlearning**, where the model owner stores the sensitive data and parameters in the form of batches during the training process. When a request for deleting the data is made, the model owner does not perform the parameter update of the batches containing the deleted data. This process can also be interpreted as selectively undoing specific machine learning steps containing sensitive data. The model training can be regarded as a series of parameter updates to the initial model parameters. The model parameters can be expressed as

$$\theta_M = \theta_{initial} + \sum_{e=1}^{E}\sum_{b=1}^{B} \triangle_{\theta_{e,b}},$$

where $\theta_{initial}$ is the initial model parameters, model M is trained for E epochs, each epoch consists of B batches. The model parameters are updated after each batch by an amount $\Delta_{\theta_{e,b}}$. During training, the model owner stores a list, which refers to the batches containing the sensitive data. When the request of removing data s belonging to the batch b, where$sb \in SB$, is received, the amnesiac unlearning method can simply remove the parameter updates from the learning parameters θM to get the $\theta M'$

$$\theta M' = \theta_{initial} + \sum_{e=1}^{E}\sum_{b=1}^{B} \triangle_{\theta_{e,b}} - \sum_{sb=1}^{SB} \triangle_{\theta_{e,b}} = \theta_M - \sum_{sb=1}^{SB} \triangle_{\theta_{e,b}} .$$

This approach has a potential drawback in that the model owner needs a large amount of storage space for storing sensitive data and related parameters. However, the authors argue that this space cost is much less than the computational and time cost of the exact unlearning methods.

Evaluation Metrics

For approximate unlearning, in addition to designing an effective and fast algorithm for data deletion, it is a significant challenge to evaluate the quality of an approximate unlearning method properly. As a result, many researchers proposed effective evaluation metrics for their algorithms.

In [20], the authors defined three evaluation metrics to measure the performance of different unlearning methods in terms of effectiveness, certifiability, and efficiency on the basis of the *Symmetric Absolute Percentage Error* (SAPE) defined as

$$SAPE(a, b) = \frac{|b - a|}{|b| + |a|} 100\%.$$

- **Effectiveness** is used to measure the prediction accuracy of a machine learning model. The error in test accuracy *AccErr* of the updated model *wu* is defined as

  $$Acc_{Err} = SAPE\left(Acc^*_{test}, Acc^u_{test}\right),$$

  where Acc^u_{test} denotes the accuracy of the updated model w^u on the test dataset $\mathcal{D}_{test}$, and Acc^*_{test} denotes the optimal accuracy of the regression model on the same dataset. The lower value of Acc_{Err} means that the prediction accuracy of w^{u} is closer to the accuracy of the initial model (in which the noise value $\theta = 0$), i.e., *wu* is more effective.
- **Certifiability** is used to measure how well the updated model w^u has unlearned the delated data. For the certifiability, both the updated model and the fully retrained model are considered, and the disparity in accuracy of the two models *AccDis* is defined as

  $$Acc_{Dis} = SAPE\left(Acc^*_{del}, Acc^u_{del}\right),$$

  where Acc^u_{del} denotes the accuracy on the deleted data D_{del} for the updated model w^u, and $Acc^\star_{del}$ denotes the accuracy on the deleted data for the fully retrained model. The lower value of Acc_{Dis} means that the updated model is more similar to the fully retrained model, i.e., the updated model *wu* has higher certifiability.

- **Efficiency** is used to measure the speed-up performance of running the algorithm to obtain the updated model *wu* and the fully retrained model $w*$

$$speed - up = \frac{timetaken\ to\ obtain\, w*}{timetaken\ to\ obtain\, wu}x.$$

Izzo et al. [36] introduced two metrics, L^2 **distance** and **Feature injection test**, to evaluate the effectiveness of an approximate data deletion method.

- L^2 **distance** between the updated model and the fully retrained model is a relatively common method used to measure the accuracy of approximate unlearning. A lower value of L^2 distance indicates that the predictive ability of the updated model is closer to that of the fully retrained model.
- **Feature injection test** is injected as a strong signal (an extra feature) into the remaining dataset, which the model (updated and fully retrained model) expects to learn. The authors measure the effectiveness of the approximate deletion method by observing the performance of the model's learned parameters before and after the removal of this particular feature.

Discussion on Data Lineage

In the process of protecting the privacy of machine learning models, the tracking of data flow is an essential part of the process [1, 10, 53–55]. Therefore, this section discusses the role of data lineage management techniques in the privacy protection of machine learning models.

Data lineage tracks data movement over time from the source system to different forms of persistence and transformations and ultimately to data consumption by an application or analytics model. The data lineage management system can monitor any data changes in the machine learning model that occur at any point in time [56]. Therefore, the combination with the data lineage management system can effectively enhance the security protection of machine learning models.

Data lineage management can be applied to defend against particular cyberattacks, such as data poisoning attacks, which can be viewed as integrity attacks. The attacker affects the model's prediction of the correct output by tampering with the training data. Even the attacker's goal is to have their input accepted as the model's training data.

Baracaldo et al. [1] proposed to identify poisonous data using the system's' lineage about the sources, transformation, and destinations of data points in the training dataset of a machine learning model as part of a filtering algorithm, which is also known as a method for detecting causative attacks. With this approach, online and periodically retrained machine learning systems can discriminate between data sources in a potentially adversarial environment. Subsequently, they applied this approach to identify poisonous data injection in the Internet of Things environment as well [10].

The Tensorflow team of Google developed a version control platform for machine learning data lineage management, Machine Learning Metadata (MLMD). MLMD can be viewed as a library to track the complete data lineage of the entire machine learning workflow, including the metadata, data preprocessing, feature selection, model training, prediction, evaluation, deployment, and so on. This work aims to answer questions like

- What hyperparameters were this model used?
- Which dataset was this model trained on?
- Which version of libraries were used to build this model?
- Which pipeline was used to build this model?
- Which version of this model was last deployed?
- What is the reason for this model's failure?

MLMD can be implemented in various machine learning pipelines to record and control all the data generated during model training. It can help developers analyze all model data transformations, including parameter updates and debugging of errors. Furthermore, from the security perspective, this metadata platform also provides ideas for the research of combining machine unlearning with data lineage.

Figure 7 explains how the data lineage management technique works in a machine learning system. With the machine unlearning approach, data lineage can still play an important role. For example, developers use machine unlearning when a user wants to withdraw sensitive personal information used to train a machine learning model or when a malicious data injection attack is detected. A part of the training data needs to be removed from the dataset to retrain the model. This process is also recorded in the data lineage management system without reservation.

Challenges

This paper describes the security risks and privacy protection issues associated with machine learning models. Both machine unlearning and data lineage management systems can play a role in addressing these issues. However, research in this area is just beginning, and researchers still face many challenges. For example, how machine unlearning can efficiently handle large amounts of data deletion tasks in a big

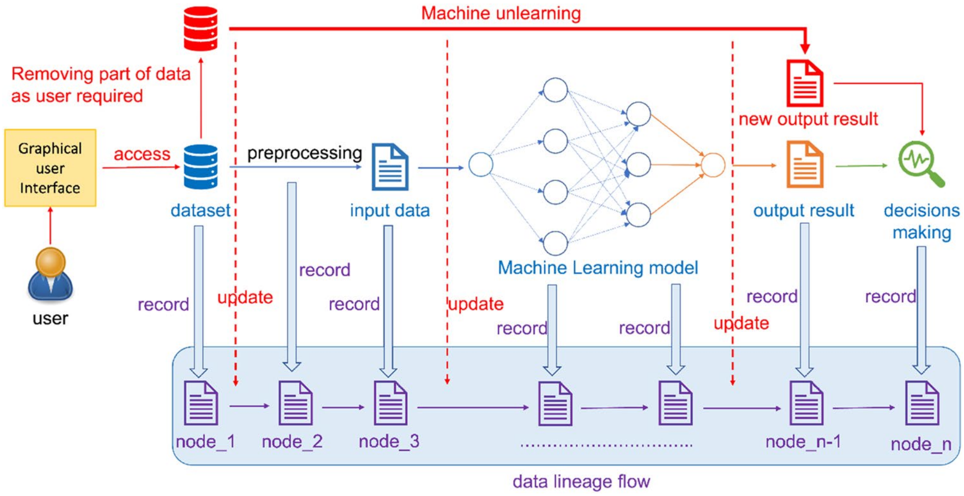


Fig. 7 Data lineage management for machine learning data flows recording and machine unlearning updating. For each step in the machine learning system, from the original dataset to optimizing each parameter in the training process to getting the final training results and analyzing the results, all data and changes will be recorded in the data lineage management system. This series of records of data flow characteristics and changes allow developers to control and track any subtle differences in the model learning process at any time

data environment; how to quickly respond to privacy theft when machine learning platforms encounter it; and how data lineage management systems can make the most of the privacy-preserving aspects of machine learning.

Machine Unlearning Algorithms

There are not many machine unlearning algorithms designed for the privacy preservation of machine learning models. Instead, algorithms with high adaptability are necessary for different user needs or data diversity. For example, the superiority of the SISA algorithm can be demonstrated when the amount of data requested for deletion are small. However, when the amount of data requested for deletion is large, the retraining approach becomes more applicable.

Active and Passive Unlearning

The machine unlearning methods we are discussing are all based on the active unlearning at the will of the data holder. However, passive unlearning is also a good option for the CIA property of machine learning models. When an attacker performs a CIA attack on a machine learning model, the data holder or the machine learning platform does not discover this attacker's behavior in time, which leads to the private data being compromised before taking countermeasures (such as machine unlearning methods). In this case, the passive model unlearning method can delete the data in time when the machine learning system is attacked, thus minimizing the loss of the data holder.

Privacy Risks of Machine Unlearning

The original intent of machine unlearning was to prevent privacy leaks caused by machine learning. However, some researchers have questioned the privacy-preserving effects of machine unlearning in recent years. Chen et al. [7] propose that machine unlearning methods can also be attacked and leak the privacy of models in specific scenarios, such as membership inference attacks [57–60]. They designed novel membership attacks and conducted experimental evaluations against two machine unlearning approaches, retraining the machine learning model from scratch and the SISA approach. The experimental results show that their attack methods significantly impact unlearning methods that handle tedious tasks, i.e., retraining from scratch. In contrast, they have less impact on distributed unlearning models like SISA.

Working with Data Lineage

At the level of security and privacy protection for machine learning, the data lineage management system can trace all data and changes in the model. With the introduction of the machine unlearning approach as a protection mechanism for machine learning models, it is imperative to use it in conjunction with a data lineage management system. The machine unlearning approach can be considered another model independent of the machine learning model used for training in the same environment. Any data changes in the machine unlearning model directly affect the security of the machine learning model used for training and, therefore, should be recorded by the data lineage management system.

Conclusion

This paper starts with the right to be forgotten about the GDPR regulations. Then, it discusses the security concerns in machine learning models and the possible privacy breaches to the data holders used for training. In this process, machine unlearning methods and data lineage management play an essential role in machine learning privacy protection. Furthermore, the challenges this research area may encounter in the future are elaborated. More and more machine learning models appear in our lives at a swift pace. While we enjoy the convenience of technological development, we cannot let down our guard on the potential security threats that may exist.

Acknowledgements This research was partially supported by the Japan Science and Technology Agency (JST) Strategic International Collaborative Research Program (SICORP). The first author was supported by JST SPRING, under Grant No. JPMJSP2136.

Funding The work of Haibo Zhang was supported by the JST-Mirai Program, JPMJSP2136.

Data availability Interested parties can obtain the anonymised datasets that support the findings of this study from the corresponding author upon reasonable request.

Declarations

Conflict of Interest On behalf of all the authors, the corresponding author states that there is no conflict of interest.

References

1. Baracaldo N, Chen B, Ludwig H, Safavi JA. Mitigating poisoning attacks on machine learning models: a data provenance based approach. In: Proceedings of the 10th ACM workshop on artificial intelligence and security. 2017;103–110
2. Liu Y, Fan M, Chen C, Liu X, Ma Z, Wang L, Ma J. Backdoor defense with machine unlearning. arXiv. 2022. https://doi.org/10.48550/arXiv.2201.09538.
3. Bourtoule L, Chandrasekaran V, Choquette-Choo CA, Jia H, Travers A, Zhang B, Lie D, Papernot N. Machine unlearning. In: 2021 IEEE symposium on security and privacy (SP). IEEE. 2021;141–159
4. Al-Rubaie M, Chang JM. Privacy-preserving machine learning: threats and solutions. IEEE Secur Priv. 2019;17(2):49–58.
5. Schelter S. Towards efficient machine unlearning via incremental view maintenance.
6. Graves L, Nagisetty V, Ganesh V. Amnesiac machine learning. arXiv. 2020. https://doi.org/10.48550/arXiv.2010.10981.
7. Chen M, Zhang Z, Wang T, Backes M, Humbert M, Zhang Y. When machine unlearning jeopardizes privacy. In: Proceedings of the 2021 ACM SIGSAC conference on computer and communications security. 2021;896–911
8. Gao J, Garg S, Mahmoody M, Vasudevan PN. Deletion inference, reconstruction, and compliance in machine (un) learning. arXiv. 2022. https://doi.org/10.48550/arXiv.2202.03460.
9. Marchant NG, Rubinstein BI, Alfeld S. Hard to forget: poisoning attacks on certified machine unlearning. arXiv preprint arXiv:2109.08266. 2021
10. Baracaldo N, Chen B, Ludwig H, Safavi A, Zhang R. Detecting poisoning attacks on machine learning in iot environments. In: 2018 IEEE international congress on internet of things (ICIOT). IEEE 2018;57–64.
11. Chundawat VS, Tarun AK, Mandal M, Kankanhalli M. Zero-shot machine unlearning. arXiv. 2022. https://doi.org/10.48550/arXiv.2201.05629.
12. Toreini E, Aitken M, Coopamootoo K, Elliott K, Zelaya CG, Van Moorsel A. The relationship between trust in ai and trustworthy machine learning technologies. In: Proceedings of the 2020 conference on fairness, accountability, and transparency. 2020;272–283.
13. Surma J. Hacking machine learning: towards the comprehensive taxonomy of attacks against machine learning systems. In: Proceedings of the 2020 the 4th international conference on innovation in artificial intelligence. 2020;1–4.
14. Tramèr F, Zhang F, Juels A, Reiter MK, Ristenpart T. Stealing machine learning models via prediction APIs. USENIX Secur Symp. 2016;16:601–18.
15. Song C, Ristenpart T, Shmatikov V. Machine learning models that remember too much. In: Proceedings of the 2017 ACM SIGSAC confer-ence on computer and communications security. 2017;587–601.
16. Shen S, Tople S, Saxena P. Auror: Defending against poisoning attacks in collaborative deep learning systems. In: Proceedings of the 32nd annual conference on computer security applications. 2016;508–519.
17. Alsdurf H, Belliveau E, Bengio Y, Deleu T, Gupta P, Ippolito D, Janda R, Jarvie M, Kolody T, Krastev S, et al. Covi white paper. arXiv. 2020. https://doi.org/10.48550/arXiv.2005.08502.
18. Ginart A, Guan MY, Valiant G, Zou J. Making ai forget you: data deletion in machine learning. arXiv. 2019. https://doi.org/10.48550/arXiv.1907.05012.
19. Fredrikson M, Jha S, Ristenpart T. Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. 2015;1322–1333.
20. Mahadevan A, Mathioudakis M. Certifiable machine unlearning for linear models. arXiv. 2021. https://doi.org/10.48550/arXiv.2106.15093.
21. Guo C, Goldstein T, Hannun A, Van Der Maaten L. Certified data removal from machine learning models. arXiv. 2019. https://doi.org/10.48550/arXiv.1911.03030.
22. Thudi A, Jia H, Shumailov I, Papernot N. On the necessity of auditable algorithmic definitions for machine unlearning. arXiv. 2021. https://doi.org/10.48550/arXiv.2110.11891.
23. Ullah E, Mai T, Rao A, Rossi RA, Arora R. Machine unlearning via algorithmic stability. In: conference on learning theory. PMLR. 2021;4126–4142.
24. Cao Y, Yang J. Towards making systems forget with machine unlearning. In: 2015 IEEE Symposium on Security and Privacy. IEEE. 2015;463–480.
25. Cao Y, Yu AF, Aday A, Stahl E, Merwine J, Yang J. Efficient repair of polluted machine learning systems via causal unlearning. In: Proceedings of the 2018 on Asia conference on computer and communications security. 2018;735–747.
26. Kashef R. A boosted svm classifier trained by incremental learning and decremental unlearning approach. Expert Syst Appl. 2021;167:114154.
27. Jose ST, Simeone O. A unified pac-bayesian framework for machine unlearning via information risk minimization. In: 2021 IEEE 3 1st international workshop on machine learning for signal processing (MLSP). IEEE. 2021;1–6.

28. Liu G, Ma X, Yang Y, Wang C, Liu J. Federaser: enabling efficient client-level data removal from federated learning models. In: 2021 IEEE/ACM 29th international symposium on quality of service (IWQOS). IEEE. 2021;1–10.
29. Brophy J, Lowd D. Machine unlearning for random forests. In: International Conference on Machine Learning. PMLR. 2021;1092–1104.
30. Wu C, Zhu S, Mitra P. Federated unlearning with knowledge distillation. arXiv. 2022. https://doi.org/10.48550/arXiv.2201.09441.
31. Du M, Chen Z, Liu C, Oak R, Song D. Lifelong anomaly detection through unlearning. In: Proceedings of the 2019 ACM SIGSAC conference on computer and communications security. 2019;1283–1297.
32. Baumhauer T, Sch¨ottle P, Zeppelzauer M. Machine unlearning: linear filtration for logit-based classifiers. arXiv. 2020. https://doi.org/10.48550/arXiv.2002.02730.
33. Golatkar A, Achille A, Soatto S. Eternal sunshine of the spotless net: selective forgetting in deep networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020;9304–9312.
34. Wu Y, Dobriban E, Davidson S. Deltagrad: rapid retraining of machine learning models. In: International conference on machine learning. PMLR. 2021;10355–10366.
35. Golatkar A, Achille A, Ravichandran A, Polito M, Soatto S. Mixed—privacy forgetting in deep networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021;792–801.
36. Izzo Z, Smart MA, Chaudhuri K, Zou J. Approximate data deletion from machine learning models. In: International conference on artificial intelligence and statistics. PMLR. 2021;2008–2016.
37. Neel S, Roth A, Sharifi-Malvajerdi S. Descent-to-delete: gradient-based methods for machine unlearning. In: Algorithmic learning theory. PMLR. 2021;931–962.
38. Thudi A, Deza G, Chandrasekaran V, Papernot N. Unrolling sgd: understanding factors influencing machine unlearning. arXiv. 2021. https://doi.org/10.48550/arXiv.2109.13398.
39. Warnecke A, Pirch L, Wressnegger C, Rieck K. Machine unlearning of features and labels. arXiv. 2021. https://doi.org/10.48550/arXiv.2108.11577.
40. He Y, Meng G, Chen K, He J, Hu X. Deepobliviate: a powerful charm for erasing data residual memory in deep neural networks. arXiv. 2021. https://doi.org/10.48550/arXiv.2105.06209.
41. Gong J, Simeone O, Kassab R, Kang J. Forget-svgd: Particle-based bayesian federated unlearning. arXiv. 2021. https://doi.org/10.48550/arXiv.2111.12056.
42. Guo T, Guo S, Zhang J, Xu W, Wang J. Vertical machine unlearning: Selectively removing sensitive information from latent feature space. arXiv. 2022. https://doi.org/10.48550/arXiv.2202.13295.
43. Cauwenberghs G, Poggio T. Incremental and decremental support vector machine learning. advances in neural information processing systems. 2000;13.
44. Tsai C-H, Lin C-Y, Lin C-J. Incremental and decremental training for linear classification. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining. 2014;343–352.
45. Karasuyama M, Takeuchi I. Multiple incremental decremental learning of support vector machines. IEEE Trans Neural Networks. 2010;21(7):1048–59.
46. Kearns M. Efficient noise-tolerant learning from statistical queries. J ACM. 1998;45(6):983–1006.
47. Martens J. New insights and perspectives on the natural gradient method. arXiv. 2014. https://doi.org/10.48550/arXiv.1412.1193.
48. Dwork C, Roth A, et al. The algorithmic foundations of differential privacy. Found Trends Theor Comput Sci. 2014;9(3–4):211–407.
49. Chaudhuri K, Monteleoni C. Privacy-preserving logistic regression. advances in neural information processing systems. 2008;21.
50. Golatkar A, Achille A, Soatto S. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations Europea conference on computer vision. Cham: Springer; 2020. p. 383–98.
51. Koh PW, Liang P. Understanding black-box predictions via influence functions. In: International conference on machine learning. PMLR. 2017;1885–1894.
52. Giordano R, Stephenson W, Liu R, Jordan M, Broderick T. A swiss army infinitesimal jackknife. In: The 22nd international conference on artificial intelligence and statistics. International conference on machine learning. PMLR. 2019;1139–1147.
53. Zhang Z, Sparks ER, Franklin MJ. Diagnosing machine learning pipelines with fine-grained lineage. In: Proceedings of the 26th international symposium on high-performance parallel and distributed computing. 2017;143–153.
54. Luo G, et al. A roadmap for automating lineage tracing to aid automatically explaining machine learning predictions for clinical decision support. JMIR Med Inform. 2021;9(5):27778.
55. Thiago RM, Souza R, Azevedo L, Soares EFDS, Santos R, Dos Santos W, De Bayser M, Cardoso MC, Moreno MF, Cerqueira R. Managing data lineage of o&g machine learning models: the sweet spot for shale use case. First EAGE Digit Conf Exhib. 2020;2020:1–5.
56. Li Y, Zheng X, Chen C, Liu J. Making recommender systems forget: learning and unlearning for erasable recommendation. arXiv. 2022. https://doi.org/10.48550/arXiv.2203.11491.
57. Shokri R, Stronati M, Song C, Shmatikov V. Membership inference attacks against machine learning models. In: 2017 IEEE Symposium on Security and Privacy (SP). IEEE. 2017;3–18.
58. Yeom S, Giacomelli I, Fredrikson M, Jha S. Privacy risk in machine learning: analyzing the connection to overfitting. In: 2018 IEEE 31st computer security foundations symposium (CSF). IEEE. 2018;268–282.
59. Sablayrolles A, Douze M, Schmid C, Ollivier Y, J´egou H. White-box vs black-box: Bayes optimal strategies for membership inference. In: International conference on machine learning. PMLR. 2019;5558–5567.
60. Hayes J, Melis L, Danezis G, De Cristofaro E. Logan: membership inference attacks against generative models. Proc Privacy Enhanc Technol De Gruyter. 2019;2019:133–52.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.