Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.032307

REVIEW

Ensuring User Privacy and Model Security via Machine Unlearning: A Review

Yonghao Tang[1], Zhiping Cai[1,*], Qiang Liu[1], Tongqing Zhou[1] and Qiang Ni[2]

[1]College of Computer, National University of Defense Technology, Changsha, 410073, China

[2]School of Computing and Communications, Lancaster University, England, B23, UK

*Corresponding Author: Zhiping Cai. Email: zpcai@nudt.edu.cn

Received: 16 January 2021 Accepted: 30 October 2021 Published: 29 November 2023

ABSTRACT

As an emerging discipline, machine learning has been widely used in artificial intelligence, education, meteorology and other fields. In the training of machine learning models, trainers need to use a large amount of practical data, which inevitably involves user privacy. Besides, by polluting the training data, a malicious adversary can poison the model, thus compromising model security. The data provider hopes that the model trainer can prove to them the confidentiality of the model. Trainer will be required to withdraw data when the trust collapses. In the meantime, trainers hope to forget the injected data to regain security when finding crafted poisoned data after the model training. Therefore, we focus on forgetting systems, the process of which we call machine unlearning, capable of forgetting specific data entirely and efficiently. In this paper, we present the first comprehensive survey of this realm. We summarize and categorize existing machine unlearning methods based on their characteristics and analyze the relation between machine unlearning and relevant fields (e.g., inference attacks and data poisoning attacks). Finally, we briefly conclude the existing research directions.

KEYWORDS

Machine learning; machine unlearning; privacy protection; trusted data deletion

1 Introduction

Machine Learning (ML) requires the use of massive amounts of practical data—usually those that contain sensitive information provided by users to train algorithm models. Moreover, following the online learning paradigm, new data is collected regularly and incrementally used to refine existing models further. Conversely, data may also need to be deleted. There are many reasons that users want systems to forget specific data.

From a privacy perspective, the data provider hopes to ensure the security of the data provided to the trainer, including that the data is not used for violating purposes, and the trained model will not leak any sensitive information about trained data when it is attacked (e.g., inference attack [1]). The trainer of the model shall preserve the data's confidentiality and verify the data's security to the user. This requires skillful authentication measures because, usually, users do not have the right to read the source code of the machine learning model directly and can only indirectly interact with the model. Generally speaking, the right to use the model for a limited time, to input the model, and

get feedback from the model. Take sentence completion model training as an example. We expect the trained model to output completed sentences with incomplete input. The data provider can query whether the sensitive data has been leaked by entering keywords, similar to 'User password is:' or 'Professional experience:' [2]. Therefore, users unsatisfied with these rising risks would want their data and its effects on the models and statistics to be forgotten entirely. Moreover, the European Union's General Data Protection Regulation (GDPR) and the former Right to Be Forgotten [3,4] both mandate that companies and organizations take reasonable steps to withdraw users' consent to their data at any time under certain circumstances. Taking the United Kingdom as an example, the email sent by the British Biobank to the researcher stated that the data provider has the right to withdraw the provided data at any time; the British legal department is still arguing about the responsibility of the trained model for the data it uses, and potential legal disputes. The Information Commissioner's Office (UK) pessimistically stated in 2020 that if users request data to be retrieved, the ongoing machine learning model may be forced to retrain or even be wholly suspended [5].

From a security perspective, users concerned about the system's future privacy risks would tend to force the system to forget about their data. Consider an E-mail sorting system. The system's security depends on the model of normal behaviors extracted from the training data. Suppose an attacker contaminates the sorting system by injecting specific designed into the training dataset. In this case, large amounts of spam will be sent to receivers, which will seriously compromise the security of the model. This type of attack, known as a data poisoning attack [6], has received widespread attention in academia and industry, where it has caused severe damage. For example, Microsoft Tay, a chatbot designed to talk to Twitter users, was shut down after just 16 h after releasing. As it started making racist comments after the poisoning attack. Such attacks make us reflect the security of machine learning models. Once the model is poisoned, the service provider must completely forget about the data to regain security.

Ideally, a model with part of the data forgotten will behave as if it was trained without those data. An intuitive way to make such models demonstrably forgettable is to retrain them from scratch. To avoid the significant computational and time overhead associated with fully retraining models affected by data deletion requests, the system must be designed with the core principle of complete and rapid forgetting of training data in such a way as to restore privacy and security. Such forgetting systems must assure the user that the systems will no longer be trained using data that the user has chosen to unlearn. In the meantime, they let users designate the data to be forgotten at different degrees of granularity. For example, a conscious privacy user who accidentally searches for a series of previously posted social photos that reveal sensitive information about the user can ask the search engine to forget that particular data. These systems then delete the data and restore its effects so that all future operations operate as if the data had never existed. Further, in a distributed learning scenario, users collaborate to forget the data. This collaborative forgetting has the potential to extend to the entire network. Users trust the forgetting system to comply with forgetting requests because the service providers mentioned above have a strong incentive to comply.

Cao et al. [7] first introduced the concept of unlearning, as a dual to ML, removing the impact of a data point on the model obtained upon completion of training, where data deletion is efficient and exact. In general, MU (Machine Unlearning) aims to guarantee that unlearning part of the training data will produce the same distribution of models that have not been trained on those data. Then, Ginart et al. [8] formalized the problem of efficient data forgetting and provided engineering principles for designing forgetting algorithms. However, these methods only behave well for non-adaptive (later training does not depend on earlier training) machine learning models, such as k-means clustering [9]. For that, Bourtoule et al. [10] proposed SISA (sharded, isolated, sliced, and

aggregated), a model-independent method, which divides the training set into disjoint slices. They train an ensemble of models and save snapshots of each model for every slice. This allows perfect unlearning but incurs heavy storage costs as multiple gradients need to be stored. Accordingly, Golatkar et al. [11] distributed a "readout" function to remove weights from the model corresponding to training data that need to be forgotten. This method does not require retraining. Currently, considering different assumptions, a wide variety of methods are emerging. For instance, in the context of model interpretability and cross-validation, researchers provided various "influence" functions for estimating the impact after removing a training point on the model [12,13]. Besides, to effectively erase the corresponding "influence" from the ML model, various forgetting techniques such as weight removal [11,14,15], linear replacement [16–18] and gradient updating [19,20] have been proposed. Except by directly measuring unlearning in different dimensional spaces to achieve approximate forgetting [1,21], current MU method also provides strict mathematical guarantees [22–24] under linear constraints and ensures that the data is certified removal in the DNN (Deep Neural Networks) model [25–27]. In other words, machine unlearning has evolved into a broad field of study, and a comprehensive introduction is beneficial for newcomers to the field.

The rest of this paper is organized as follows. Section 2 briefly describes the common threats of attacks on machine learning. Section 3 provides an overview of existing machine unlearning methods. The conclusion is provided in Section 4 at the end.

2 Weaknesses of Machine Learning Models

Machine learning models, instense learning models, have achieved fruitful results in recent decades, producing research on how to attack machine learning models. This section will discuss several kinds of attacks on machine learning models.

2.1 Inference Attacks

Liu et al. [1] summarized four inference attacks on machine learning models, including membership inference, model inversion, attribute inference, and model stealing. The differences between the various attacks can be seen in "Table 1: Inference Attacks on Machine Learning Models".

1) *Membership inference:* Membership inference refers to malicious trainers in the training party who intervene in the training model, which will finally force the model to malfunction in specific training data [2].
2) *Model inversion:* Refers to the existence of external enemies trying to replicate the model's training data by studying the model.
3) *Attribute inference:* Attribute inference means that the model is used for unexpected purposes. For example, a machine learning model that predicts the person's age in the picture may be used to identify the person's face in the photo.
4) *Model stealing:* Model stealing aims to reproduce a model with similar effects by evaluating the trained model's performance.

In particular, the first three methods can determine whether certain specific data belongs to a machine learning model [28,29]. This could be a double-edged sword for the confidentiality of the model. It allows users to verify that their data has not been maliciously used for model training and exposes the training data to the risk of leakage.

Table 1: Inference attacks on machine learning models

Inference attacks	Attackers	Target	Object
Membership inference	Malicious trainers	Training model	Force the trained model to malfunction in specific training data
Model inversion	External enemies	Training data	To replicate the model's training data
Attribute inference	Anyone authorized to the model	Trained model	Use the model for unexpected purposes
Model stealing	External enemies	Trained model	Reproduce a model with similar effects

2.2 Data Poisoning Attacks

Another type of machine learning attack comes from data poisoning attacks, also known as harmful data attacks or adversarial examples [30]. Data poisoning attacks include malicious label attacks, irrelevant data bombings, and reflection data attacks. Generally speaking, data poisoning attacks add maliciously tampered data to the training data of the machine learning model in order to create backdoors on the model. Then attackers use these backdoors to perform destructive work, such as creating specifically labelled faces in the face recognition model. This attack usually fulfils the following properties: the targeted model typically performs when recognizing data without a backdoor, the targeted model can correctly respond to the backdoor, and the overall structure of the targeted model is not changed.

Some malicious label attacks use data with artificial misleading labels, while others include well-labelled data that have been processed invisibly to the naked eye. Huang et al. [6] used harmful data to attack a product recommendation system. They create fake users and send them to a product recommendation system based on machine learning so that the target product can be recommended to users. As a result, even under the protection of the fake user detector, the product recommendation model is still under attack. This shows that poisoned data attacks are effective and difficult to counter. Besides, poisoned data attacks are also brutal to prevent as it is difficult for trainers to check the reliability of the data.

Liu et al. [31] described in detail how they deploy reflection attacks on a deep learning neural network. They generate a poisoned image by adding a subtitle cover of reflection to clean background images, which will create a backdoor in the training model. Although the model still functions accurately in most of the standard inputs, the attacker will control any inputs with backdoor patterns. Their experiment attacked a machine learning model that recognizes road signs. Their attack made the model's correct rate of identifying stop warning signs and speed limit signs plummet by 25 per cent. Even after the machine learning model was retrained, the effects of harmful data attacks still existed.

Recently, an opportunistic backdoor attack approach has been proposed in speech recognition which is characterized by passively triggering and opportunistically invoking [32]. Unlike the current backdoor design in CV (Computer Vision) and SR (Speech Recognition) systems require an adversary to put on a mask or play some audios in the field to trigger the backdoor in the poisoned model. In contrast, the opportunistic attack is plug-and-conditionally-run, which avoids unrealistic presence requirements and provides more possibilities for the attacking scenarios (e.g., indoor).

2.3 Other Attacks

Gupta et al. [33] discussed the machine learning model in the case of adaptive, that is, the machine model under the request of the user to withdraw specific data due to how the models behave. They mainly focused on the model's situation relying on a convex optimization problem. They designed an attacking stratagem and finished an experiment which proved that the machine model would be vulnerable to attack after the user withdraws the data. In the non-convex machine model, the model's safety is difficult to guarantee after users request to delete specific data. In particular, they proved that adding a little noise to the model can meet the needs of adaptive data deletion.

Customary computation model, such as MapReduce, requires weeks or even months to train with high hardware costs [34]. Some model trainers would like to train their models based on pre-trained models acquired from outsourced code societies. An adversary may spread technically modified models through the Internet [35]. When a model trainer uses these modified models as samples of their model, the robustness and security of machine learning models may be compromised. Gu et al. [36] showed that this attack is similar to inference attack, but still has special qualities when referring to fully outsourced trained and transfer training.

3 Machine Unlearning

In the past seven years, many machine unlearning (data forgotten) methods were proposed. In this section, we formally define unlearning focus on machine learning and give the corresponding evaluation metrics. After that, we summarize and categorize existing MU approaches in detail based on their characteristics, as shown in Fig. 1. Unlearning for other tasks or paradigms is also discussed at the end. We next discuss the definition of MU, its metrics and various well-established MU.

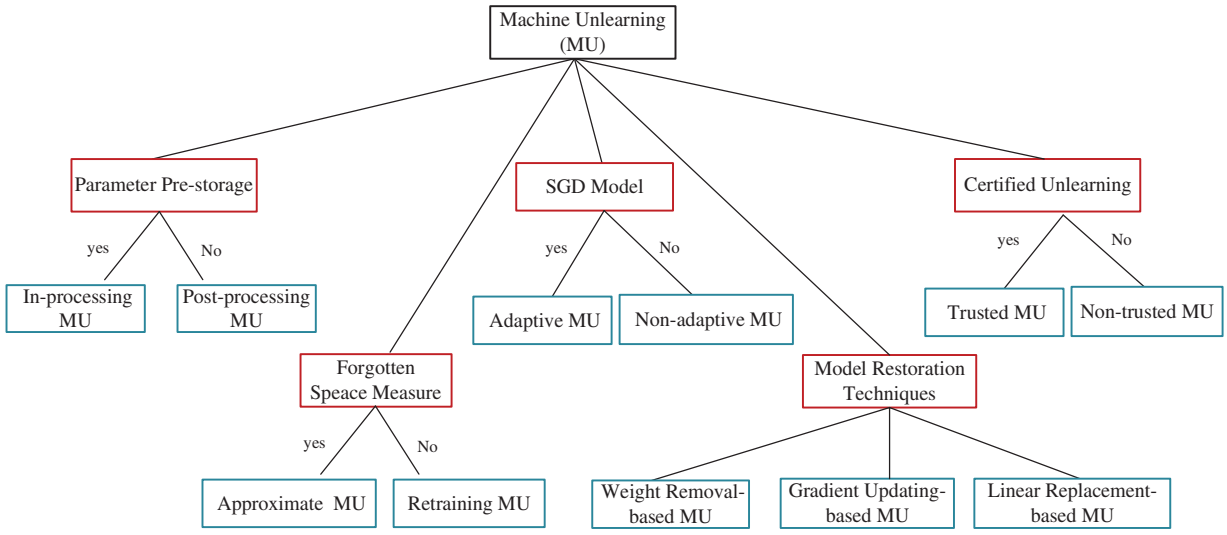

Figure 1: Taxonomy of machine unlearning with different categorization criteria. In this figure, the red boxes represent categorization criteria, while the blue boxes indicate MU subtypes. The categories are parallel to each other and intersect with each other

3.1 The Definition of Unlearning

Let $\mathcal{U}_D$ define the distribution of models a training rule could return when trained on a dataset $\mathcal{D}$. Let $\mathcal{D}' = \mathcal{D}/x^*$, where $x^* \in \mathcal{D}$ is the datapoint that would be unlearned. Similarly, $\mathcal{U}'_D$ denote the distribution of models learned using same training rule on $\mathcal{D}'$. Lastly, we define the mechanism (i.e., some randomized or deterministic process) $\mathcal{F}: \mathcal{U} \rightarrow \mathcal{S}$, where $\mathcal{S} = \mathcal{F}(\mathcal{U}_D, x^*)$ represent the distribution of output model after the transformation by $\mathcal{F}$ on x^*. Now if $\mathcal{S} = \mathcal{U}'_D$, we say the $\mathcal{F}$ is the *exact unlearning mechanism.* As such, naively retraining without x^* as the unlearning mechanism $\mathcal{F}$ can guarantee the above definition. However, the issue with naively retraining is the sizeable computational overhead associated with it. *Approximate unlearning mechanism* tries to alleviate these cost-related concerns. Instead of retraining, researchers attempt to execute computationally less expensive operations to measure the distance between $\mathcal{S}$ and $\mathcal{U}_{\mathcal{D}'}$, using different their defined forgotten space (outputs or weights of the model).

3.2 Evaluation Metrics

To evaluate the performance of MU in the experiment, two classical metrics are usually adopted, including (1) how completely they can forget data (*completeness*) and (2) and how quickly they can do so (*timeliness*). The higher these metrics, the better the algorithm is at restoring privacy and security. Besides, for some Pre-processing MU, we are still required to consider the computational overhead rather than merely evaluating forgetting efficiency [37].

3.3 Machine Unlearning for Removing Data

1) In-processing MU: The core idea of In-processing MU is to store some of the training parameters during the training of the model, and when a requirement to unlearn the training points exists, we simply retrain the part of the affected model, and since the shards are smaller than the whole training set, this reduces the retraining time to achieve unlearning. This typical approach is SISA [38], which we have described in Section 1. Similarly, Wu et al. [39] splited the data into multiple subsets and trains the models separately based on the combination of these subsets. The advantage of the above approaches is that it provides a strong demonstration of why the new model is not influenced in any way by the point to be learned since it has not been trained on that point. Accordingly, Brophy et al. [22] proposed two unlearning algorithms for random forest: (1) cache data statistics at each node and training data at each leaf so that only the necessary subtrees are retrained. (2) randomly select split variables at the upper level of each tree so that the selection is entirely independent of the data and no changes are required. At lower levels, split variables are selected to greedily maximize splitting conditions such as the Gini index or mutual information. Other methods, including [8,40–42], require parameters to be stored during training, the dwelling of note as their other more salient details we will detail subsequently in other parts.

2) Approximate MU: Approximate MU is a kind of post-hoc (post-training) approximate (avoiding retraining) unlearning method. Approximate MU has the advantage of being more computationally efficient than retraining. Still, at the cost of weaker guarantees: the learned model may not be completely unaffected by unlearned data points. Underneath the ambiguity of various approximate statements about forgetting, there are various methods to be considered for this forgetting criterion and their associated metrics: each type has their metric to measure forgetting, but it is not clear how to compare the statements of the different metrics. Thudi et al. [43] introduced an inexpensive unlearning mechanism called single-gradient forgetting, proposing a Standard Deviation (SD) loss metric for the unlearning space. This makes it possible to use single-gradient cancellation to cancel a

point from a model trained with SD loss, effectively reducing this unlearning error. Mahadevan and Mathioudakis [44] proposed a similar approach to differential privacy. It uses differential privacy and full information to give a general reduction from deletion guarantees for adaptive sequences to deletion guarantees for non-adaptive sequences. It is worth noting that it essentially defines forgetting in terms of Logits and measures the degree of unlearning through the distribution of Logits or membership inference [2]. Except by directly measuring unlearning, researchers also utilize the "influence" function to measure the impact of data points in model predictions [12,13,16]. A hessian-based method is provided for estimating the influence of a training point on the model predictions [16]. In all, how to definite the unlearning and design its measured metric is still an important open question.

3) Non-adaptive MU: MU was first proposed in [7], which is a typical non-adaptive MU, and also including [8,45], which relies on the notion of model stability, arguing that removing a sufficiently small proportion of data should not lead to significant changes in the learned model. These studies aim to remove precisely one or more training samples from a training model: their measure of success is near-optimal parameters or objective values and the distinguishing feature is that they are specific to a particular model and can provide tight mathematical guarantees under some constraint settings. Based on the same idea, Fu et al. [23] proposed a Bayesian Inference Forgetting (BIF) framework and develops unlearning algorithms for variational inference and Markov chain Monte Carlo algorithms. It is also shown that BIF can demonstrate the elimination of the effect of a single benchmark on a learning model. Accordingly, a similarly novel unlearning approach proposed by Nguyen in [24] argues that the Kullback-Leibler scatters between the approximate posterior beliefs of the model parameters after direct removal from the erased data and the exact posterior beliefs after retraining with the remaining data is minimized. Despite their not insignificant limitations in practical application, empirically, we cannot deny that they have laid a solid foundation for MU and effectively contributed to the development of adaptive MU (most of the MU methods [33,40] in this paper fall into the adaptive category. We will not expand on their specific description here).

4) Weight Removal-based MU: To effectively erase the corresponding "influence" from the ML model, Golatkar et al. [11] proposed to use a "readout" function to remove weights from the model. Instead, the weights are modified so that any probe function of the weights is indistinguishable from the same function applied to the network weights, and the network training data is not forgotten. This condition is a generalized, weaker form of differential privacy. Afterwards, Golatkar et al. [18] proposed a method that improves and generalizes previous approaches [11] to different read-out functions and can be extended to ensure unlearning when the network is eventually activated. Introducing a new constraint condition can effectively remove the upper limit of information from the forgetting queue per query in a black-box setting (i.e., only observes input-output behaviour). Besides, in a white box setting (i.e., with complete control over the model), their proposed unlearning process has a deterministic part derived from a linearized version of the differential equations of the model and a stochastic part that ensures information destruction by adding noise tailored to the geometry of the loss landscape, thus effectively removing the weight of the unlearn data. Accordingly, another similar method redefines unlearning and further provides for approximate unlearning by removing weight [15]. This attempt is an essential step towards successful machine unlearning for model restoration.

5) Gradient Updating-based MU: Kamalika et al. [19] provided a method based on residual projection updates that use synthetic data points to remove data points based on linear regression models. It computes the projection of the exact parameter update vector to a specific low-dimensional subspace. Its feature is that the residual projection update has a runtime that only scales linearly in the dimensionality of the data. Other methods, such as [8], have a quadratic or higher dependence on dimensionality. Reference [20] proposed a mask gradient unlearning algorithm and a forgetting

rate indicator based on affiliation inference. The core idea of being unlearned is to mask the neurons of the target model with gradients (called mask gradients) that are trained to eliminate the memory of some of the training samples in the specific model. For the convex risk minimization problem, Sekhari designed noisy Stochastic Gradient Descent (SGD) based TV stabilisation algorithms [21]. Their main contribution is the design of corresponding efficient cancellation learning algorithms based on constructing a (maximally) coupled Markov chain to the noisy SGD process. Since this work is well suited to exist adaptive MU, we believe it is very malicious and worth further exploration.

6) Linear Replacement MU: Baumhauer et al. proposed a logarithm-based method for unlearning classification models, where the output algorithm is linearly transformed, but the information in the weights is not removed [17]. Essentially, this is a filtering technique for output results, which can be used to prevent privacy breaches. Reference [38] pre-trained a nonconvex model on data that will never be deleted and then does convex fine-tuning on user data. To effectively remove all the information contained in the non-core data (i.e., user data), by replacing the standard deep network with a suitable linear approximation, with appropriate changes to the network structure and training process, it has shown that this linear approximation can achieve comparable performance to the original network and that the forgetting problem becomes quadratic and can even be effectively solved for large models. Other methods like [16] have a similar mindset.

7) Trusted MU: Different from *previous* standard MU methods, the issue of users determining whether their data has been deleted in ML is crucial to the field of MU, and this has a degree of influence on whether MU can be used in commercial systems. Therefore, unlike the previous categories, we will give the necessary emphasis to the relevant context in this section.

(1) Relevant Background: The user may request the machine learning model trainer to recover the provided data due to his confidentiality requirements, or simply distrust, which leads to the concept of trusted data deletion. The birth of trusted data deletion predates machine learning. The initial research in this period was based on the physical level. Some data stored in computer hardware urgently needs to be destroyed, and it is necessary to ensure that no one can restore the data after the deletion is completed. The primitive method uses physical media that need to be continuously rewritten or cannot be stored for a long time, such as flash memory. However, machine learning is an algorithm model based on experience, and its iterative update characteristics are destined to require reliable preservation [46].

There are several points to pay attention to when removing data. The first point is authentication, which means the data owner could verify whether their data is indeed deleted from the model. Yang et al. [47] gave a data deletion method based on the cloud server, without a third party, allowing the holder of the data to delete and verify the result of the deletion. Their algorithm is based on vector authentication, which can prevent external attackers from stealing data, preventing cloud servers from tampering with data, and preventing cloud servers from maliciously backing up data or transferring data. Hua et al. [48] produced another survey about data deletion in cloud server. The confidentiality, data integrity, authenticity, and accountability of data users proposed by them are similar to the definitions of data protection by many legislative bodies.

(2) Existing Methods: Based on a linear classifier, Guo et al. [25] proposed a MU mechanism that supports authentication. The mechanism is based on differential privacy technique, and an algorithm for handling convex problems based on second-order Newtonian updates is given. To ensure that an adversary cannot extract information from the small residuals (i.e., proof removal), randomly interfering with training losses is used to mask the residuals. Furthermore, Sommer et al. [26] proposed a verifiable probabilistic MU algorithm based on a backdoor attack, querying whether the output of

the backdoor data is injected (specified) label by the user in advance to confirm whether the model truly deletes the data. This is a general trusted MU method but has a disadvantage that it does not allow exact forgetting (i.e., forgetting specific individual data). Similarly, Reference [49] proposed a trusted ML method based on membership inference. Constructing a model with honeypots, which can infer whether those adversary data still existed in the training set, thus guaranteeing that the MU could be trusted. Currently, only a few works are focusing on this area. Leom et al. [50] researched the remote wiping problem in mobile devices. They assumed that the problem occurred when the device was stolen and the user sent a data deletion instruction to the device. Ullah et al. [51] identified a notion of algorithmic stability. Their work propose MU on smooth convex empirical risk minimization problems. What's more, their algorithm also fulfills differentially private.

3.4 Machine Unlearning for Other Fields or Paradigms

Currently, most existing MU methods are still focusing on a monolithic form of data or form of application [52]. However, the vast differences between different tasks and paradigms might make the similar methods to the entirely different results. For example, Gradient Updating-based MU may fail in federal scenarios, as the federal averaging algorithm will vastly reduce the impact of gradients. As such, each MU algorithm against other tasks or paradigms faces new challenges. Reference [53] provided the first framework for quick data summarization with data deletion using robust streaming submodular optimization. Accordingly, an exact MU method is proposed under the assumption that learning takes place in federated learning [38]. In federated learning, independent ML models are trained on different data partitions, and their predictions are aggregated during the inference process. In graph neural networks, due to the directly applying of SISA in the graph data can severely damage the graph structural information, Chen et al. [41] proposed two novel graph partitioning algorithms and a learning-based aggregation method. Generally speaking, MU can be used for good privacy protection. However, a new privacy threat is revealed: MU may leave some data imprints in the ML model [5]. Using the original model and the post-deletion model, ordinary member inference attacks can infer the deletion of a user's private information. More interestingly, Marchant et al. [54] argued that current approximation MU and retraining training do not set practical bounds for computation and proposed a poisoning attack against MU which can effectively increase the computational cost of data forgetting.

4 Conclusion

Machine Unlearning, including general MU and trusted MU, is a critical and booming research area. In this survey, we first summarize and categorize major ML attacks and existing machine unlearning. Specifically, we detail inference and poisoning attacks, which are threats encountered by MU's antagonist, ML, and where MU attempts to mitigate. In addition, we present a comprehensive overview of MU from five perspectives: data storage, unlearning metrics, model properties, means of unlearning and authentication of unlearning. We hope that this paper could remind researchers of the ML attack threat, and the significance of MU and provide a timely view. It would be an essential step towards trustworthy deep learning.

Acknowledgement: None.

Funding Statement: This work is supported by the National Key Research and Development Program of China (2020YFC2003404), the National Natura Science Foundation of China (No. 62072465, 62172155, 62102425, 62102429), the Science and Technology Innovation Program of Hunan Province

(Nos. 2022RC3061, 2021RC2071), and the Natural Science Foundation of Hunan Province (No. 2022JJ40564).

Author Contribution: Yonghao Tang: Wrote the initial draft of the manuscript, reviewed and edited the manuscript. **Qiang Liu:** Wrote the paper, Reviewed and edited the manuscript. **Zhiping Cai:** Performed the project administration, reviewed and edited the manuscript. **Tongqing Zhou:** Reviewed and edited the manuscript. **Qiang Ni:** Reviewed and edited the manuscript.

Availability of Data and Materials: Data and materials availability is not applicable to this article as no new data or material were created or analyzed in this study.

Conflicts of Interest: The authors declare they have no conflicts of interest to report regarding the present study.

References

[1] Y. Liu, R. Wen, X. He, A. Salem, Z. Zhang *et al.,* "ML-Doctor: Holistic risk assessment of inference attacks against machine learning models," arXiv preprint arXiv:2102.02551, 2021.

[2] R. Shokri, M. Stronati, C. Song and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symp. on Security and Privacy (SP)*, San Jose, CA, USA, IEEE, pp. 3–18, 2017.

[3] Council of European Union. Council regulation (EU) no 2012/0011, 2014. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52012PC0011 (accessed on 25/01/2012).

[4] Council of European Union. Council regulation (EU) no 2016/678, 2014. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj (accessed on 27/04/2016).

[5] M. Chen, Z. Zhang, T. Wang, M. Backes, M. Humbert *et al.,* "When machine unlearning jeopardizes privacy," in *Proc. of the 2021 ACM SIGSAC Conf. on Computer and Communications Security*, New York, NY, USA, pp. 896–911, 2021.

[6] H. Huang, J. Mu, N. Z. Gong, Q. Li, B. Liu *et al.,* "Data poisoning attacks to deep learning based recommender systems," arXiv preprint arXiv:2101.02644, 2021.

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

[8] A. Ginart, M. Guan, G. Valiant and J. Y. Zou, "Making ai forget you: Data deletion in machine learning," in *33rd Conf. on Neural Information Processing Systems*, Vancouver, BC, Canada, pp. 3518–3531, 2019.

[9] Q. Zhang, S. Jia, B. Chang and B. Chen, "Ensuring data confidentiality via plausibly deniable encryption and secure deletion–a survey," *Cybersecurity*, vol. 1, no. 1, pp. 1–20, 2018.

[10] L. Bourtoule, V. Chandrasekaran, C. Choquette-Choo, H. Jia, A. Travers *et al.,* "Machine unlearning," in *2021 IEEE Symp. on Security and Privacy*, San Francisco, CA, USA, pp. 141–159, 2021.

[11] A. Golatkar, A. Achille and S. Soatto, "Eternal sunshine of the spotless net: Selective forgetting in deep networks," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, CA, USA, pp. 9304–9312, 2020.

[12] S. Shintre and J. Dhaliwal, "Verifying that the influence of a user data point has been removed from a machine learning classifier," *Justia Patents*, Patent # 10,225,277, 2019.

[13] S. Garg, S. Goldwasser and P. N. Vasudevan, "Formalizing data deletion in the context of the right to be forgotten," in *Advances in Cryptology–EUROCRYPT 2020*, Zagreb, Croatia, pp. 373–402, 2020.

[14] A. Golatkar, A. Achille and S. Soatto, "Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations," in *European Conf. on Computer Vision*, Glasgow US, Springer, Cham, pp. 383–398, 2020.

[15] A. Thudi, H. Jia, I. Shumailov and N. Papernot, "On the necessity of auditable algorithmic definitions for machine unlearning," in *Proc. of the 31st USENIX Security Symp.*, Boston, MA, USA, 2021.

[16] R. Giordano, W. Stephenson, R. Liu, M. Jordan and T. Broderick, "A swiss army infinitesimal jackknife," in *22nd Int. Conf. on Artificial Intelligence and Statistics*, Naha, Okinawa, Japan, pp. 1139–1147, 2019.
[17] T. Baumhauer, P. Schöttle and M. Zeppelzauer, "Machine unlearning: Linear filtration for logit-based classifiers," arXiv preprint arXiv:2002.02730, 2020.
[18] A. Golatkar, A. Achille, A. Ravichandran, M. Polito and S. Soatto, "Mixed-privacy forgetting in deep networks," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Nashville, TN, USA, pp. 792–801, 2021.
[19] Z. I. M. A. S. Kamalika and C. J. Zou, "Approximate data deletion from machine learning models: Algorithms and evaluations," in *Int. Conf. on Artificial Intelligence and Statistics*, San Diego, California, USA, pp. 2008–2016, 2021.
[20] Y. Liu, Z. Ma, X. Liu, J. Liu, Z. Jiang *et al.,* "Learn to forget: Machine unlearning via neuron masking," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 1–14, 2020.
[21] A. Sekhari, J. Acharya, G. Kamath and A. T. Suresh, "Remember what you want to forget: Algorithms for machine unlearning," in *Thirty-fifth Conf. on Neural Information Processing Systems*, Red Hook, NY, USA, vol. 34, 2021.
[22] J. Brophy and D. Lowd, "DART: Data addition and removal trees," arXiv preprint arXiv:2009.05567, 2020.
[23] S. Fu, F. He, Y. Xu and D. Tao, "Bayesian inference forgetting," arXiv preprint arXiv:2101.06417, 2021.
[24] Q. P. Nguyen, B. K. H. Low and P. Jaillet, "Variational bayesian unlearning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16025–16036, 2020.
[25] C. Guo, T. Goldstein, A. Hannun and L. van der Maaten, "Certified data removal from machine learning models," arXiv preprint arXiv:1911.03030, 2019.
[26] D. M. Sommer, L. Song, S. Wagh and P. Mittal, "Towards probabilistic verification of machine unlearning," arXiv preprint arXiv:2003.04247, 2020.
[27] X. Liu and S. A. Tsaftaris, "Have you forgotten? A method to assess if machine learning models have forgotten data," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*, Lima, Peru, Springer, pp. 95–105, 2020.
[28] C. Song and V. Shmatikov, "Auditing data provenance in text-generation models," in *Proc. of the 25th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, Anchorage, AK, USA, pp. 196–206, 2019.
[29] S. Zhao, M. Hu, Z. Cai, Z. Zhang, T. Zhou *et al.,* "Enhancing Chinese character representation with lattice-aligned attention," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 1, no. 1, pp. 1–10, 2021.
[30] J. Zhang and J. Wang. "A survey on adversarial example," *Journal of Information Hiding and Privacy Protection*, vol. 2, no. 1, pp. 47–57, 2020.
[31] Y. Liu, X. Ma, J. Bailey and F. Lu, "Reflection backdoor: A natural backdoor attack on deep neural networks," in *European Conf. on Computer Vision*, Edinburgh, UK, pp. 182–199, 2020.
[32] Q. Liu, T. Q. Zhou, Z. P. Cai and Y. H. Tang, "Opportunistic backdoor attacks: Exploring human-imperceptible vulnerabilities on speech recognition systems," in *Proc. of the 30th ACM Int. Conf. on Multimedia*, Lisbon, Portugal, pp. 1–9, 2022.
[33] V. Gupta, C. Jung, S. Neel, A. Roth, S. Sharifi-Malvajerdi *et al.,* "Adaptive machine unlearning," in *Thirty-fifth Conf. on Neural Information Processing Systems*, Red Hook, NY, USA, vol. 34, pp. 16319–16330, 2021.
[34] U. Selvi and S. Pushpa, "Machine learning privacy aware anonymization using mapreduce based neural network," *Intelligent Automation & Soft Computing*, vol. 31, no. 2, pp. 1185–1196, 2022.
[35] M. Abu-Alhaija and N. M. Turab, "Automated learning of ECG streaming data through machine learning Internet of Things," *Intelligent Automation & Soft Computing*, vol. 32, no. 1, pp. 45–53, 2022.
[36] T. Gu, K. Liu, B. Dolan-Gavitt and S. Garg, "BadNets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47230–47244, 2019.
[37] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," in *28th USENIX Security Symp. (USENIX Security 19)*, Santa Clara, CA, USA, pp. 267–284, 2019.

[38] G. Liu, X. Ma, Y. Yang, C. Wang and J. Liu, "Federated unlearning," arXiv preprint arXiv:2012.13891, 2020.
[39] Y. Wu, E. Dobriban and S. Davidson, "Delta-grad: Rapid retraining of machine learning models," in *Int. Conf. on Machine Learning*, Vienna, Austria, PMLR, pp. 10355–10366, 2020.
[40] S. Neel, A. Roth and S. Sharifi-Malvajerdi, "Descent-to-delete: Gradient-based methods for machine unlearning," in *Algorithmic Learning Theory*, Paris, France: Springer, pp. 931–962, 2021.
[41] M. Chen, Z. Zhang, T. Wang, M. Backes, M. Humbert *et al.,* "Graph unlearning," arXiv preprint arXiv:2103.14991, 2021.
[42] A. Warnecke, L. Pirch, C. Wressnegger and K. Rieck, "Machine unlearning of features and labels," arXiv preprint arXiv:2108.11577, 2021.
[43] A. Thudi, G. Deza, V. Chandrasekaran and N. Papernot, "Unrolling sgd: Understanding factors influencing machine unlearning," arXiv preprint arXiv:2109.13398, 2021.
[44] A. Mahadevan and M. Mathioudakis, "Certifiable machine unlearning for linear models," arXiv preprint arXiv:2106.15093, 2021.
[45] S. Schelter, "Amnesia—A selection of machine learning models that can forget user data very fast," *Suicide*, vol. 8364, no. 44035, pp. 46992, 2020.
[46] M. Jegorova, C. Kaul, C. Mayor, A. Q. O'Neil, A. Weir *et al.,* "Survey: Leakage and privacy at inference time," arXiv preprint arXiv:2107.01614, 2021.
[47] C. Yang, X. Tao and F. Zhao, "Publicly verifiable data transfer and deletion scheme for cloud storage," *International Journal of Distributed Sensor Networks*, vol. 15, no. 10, pp. 1550147719878999, 2019.
[48] M. Hua, Y. Zhao and T. Jiang, "Secure data deletion in cloud storage: A survey," *International Journal of Embedded Systems*, vol. 12, no. 2, pp. 253–265, 2020.
[49] S. Shan, E. Wenger, B. Wang, B. Li, H. Zheng *et al.,* "Gotta catch'em all: Using honeypots to catch adversarial attacks on neural networks," *Proc. of the 2020 ACM SIGSAC Conf. on Computer and Communications Security (CCS '20)*, New York, USA, pp. 67–83, 2020.
[50] M. D. Leom, K. K. R. Choo and R. Hunt, "Remote wiping and secure deletion on mobile devices: A review," *Journal of Forensic Sciences*, vol. 61, no. 6, pp. 1473–1492, 2016.
[51] E. Ullah, T. Mai, A. Rao, R. A. Rossi and R. Arora, "Machine unlearning via algorithmic stability," in *Conf. on Learning Theory*, Boulder, Colorado, pp. 4126–4142, 2021.
[52] M. Du, Z. Chen, C. Liu, R. Oak and D. Song, "Lifelong anomaly detection through unlearning," in *Proc. of the 2019 ACM SIGSAC Conf. on Computer and Communications Security*, London, UK, pp. 1283–1297, 2019.
[53] B. Mirzasoleiman, A. Karbasi and A. Krause, "Deletion-robust submodular maximization: Data summarization with "the right to be forgotten"," in *Int. Conf. on Machine Learning*, Sydney, Australia, pp. 2449–2458, 2017.
[54] N. G. Marchant, B. I. Rubinstein and S. Alfeld, "Hard to forget: Poisoning attacks on certified machine unlearning," in *Proc. of the AAAI Conf. on Artificial Intelligence*, Palo Alto, California, USA, vol. 36, no. 7, pp. 7691–7700, 2022.