Cognitive Computation (2024) 16:482–506
https://doi.org/10.1007/s12559-023-10219-3

Machine Un-learning: An Overview of Techniques, Applications, and Future Directions

Siva Sai[1] · Uday Mittal[1] · Vinay Chamola[1] · Kaizhu Huang[1] · Indro Spinelli[1] · Simone Scardapane[1] · Zhiyuan Tan[1] · Amir Hussain[1]

Received: 13 July 2023 / Accepted: 22 October 2023 / Published online: 4 November 2023

Abstract

ML applications proliferate across various sectors. Large internet firms employ ML to train intelligent models using vast datasets, including sensitive user information. However, new regulations like GDPR require data removal by businesses. Deleting data from ML models is more complex than databases. Machine Un-learning (MUL), an emerging field, garners academic interest for selectively erasing learned data from ML models. MUL benefits multiple disciplines, enhancing privacy, security, usability, and accuracy. This article reviews MUL's significance, providing a taxonomy and summarizing key MUL algorithms. We categorize modern MUL models by criteria, including model independence, data driven, and implementation considerations. We explore MUL applications in smart devices and recommendation systems. We also identify open questions and future research areas. This work advances methods for implementing regulations like GDPR and safeguarding user privacy.

Keywords Machine unlearning · Privacy · GDPR · Data deletion

Introduction

We know that groups have access to our data, but the process for removing that data is still unclear. When ML is used to analyze our data, this problem worsens since these algorithms can store and access all the data's information. Notably, models taught using decision trees and deep learning algorithms unintentionally remember certain elements of their training material [1, 2].

As a result, a crucial issue arises: how should a company handle its ML models when a user asks for the removal of data they previously gave their authorization to share?

The significance of answering this question cannot be overstated, especially in light of current laws mandating businesses to allow customers to erase their data (such as the GDPR's Right to be Forgotten) [3–6]. Transparency in technology and data use policies must grow to comply with these rules [7]. As a result, there is a growing need for a system that would allow trained machine-learning models to choose to ignore whatever knowledge they may have picked up from certain data sets.

Pinpointing the exact adjustments needed to get the desired results is the main challenge faced in MUL. Instead, it comes from an unclear understanding of the intended changes to the outcome, which is the probability distribution of a model's outputs. One may suggest an "active unlearning" method, in which, in the case of a recommender system, the customer may be presented with a set of recommendations and asked to explicitly choose which recommendations are wrong so that the system can make corrections to itself [8, 9]. But note that this is not "correcting" the model as much as it is making it "adapt" — just like in reinforcement learning [10]. Instead of going back and undoing any previous actions, we are adding more corrective data to the dataset, which causes the machine to behave differently. This approach could lead to the desired outcomes, but it fundamentally differs from MUL.

Practical working models usually operate in non-uniform and non-convex loss landscapes with various local optima, which is another thing to consider. As a result, the same model may converge to different sets of final weights while utilizing a given dataset by simply changing the order in which data

✉ Vinay Chamola
vinay.chamola@pilani.bits-pilani.ac.in

[1] Birla Institute of Technology and Science - Pilani Campus: Birla Institute of Technology & Science Pilani, Pilani, Rajasthan, India

points are added during incremental training updates. This highlights the absence of a clear goal configuration. In most real-world situations, there is no predetermined goal distribution to achieve because of the unlearning process. As a result, it is challenging to describe MUL in terms of a particular goal. The only requirement is that, in some manner, the distribution of results matches what it would have been if the computer had not been trained on the particular data in question. The specific parameters of the dispersion are yet unknown. The only way to evaluate our performance is to train another network from scratch without deleting data and compare the unlearned distribution with the newly trained one. Another approach is discussed later, a better alternative to retaining the whole model. There is an obvious need for a thorough evaluation of the current literature, given the critical significance of MUL in modern society and how it solves issues of privacy, security, fidelity, and usability within ML models. In the current paper, we thoroughly review MUL to fill this research gap [11–13].

Organization of This Study

Initially, we establish the need for MUL in "Need for Machine Unlearning" followed by ML preliminaries in "Preliminaries" and a detailed definition of MUL in "Machine Unlearning". In "Data-Driven Unlearning," "Model Independent Unlearning," and "Model-Specific Unlearning," we classify and present the existing works in MUL based on multiple classification criteria — data-based, exactness of unlearning-based, model-independent method-based, and ML model-based. We describe the evaluation of MUL models in "Evaluation of Machine Unlearning Models" followed by an analysis of the real-world deployment of MUL models in the world in "Analyzing Machine Unlearning Models: Real-World Deployment." We examine the incorporation of MUL methods in "Incorporating Machine Unlearning Algorithms in Different Paradigms of Machine Learning."

Overview of This Survey

We put forward several applications of MUL in "Applications of Machine Unlearning" followed by crucial challenges and future directions in "Challenges and Future Prospects". Finally, we conclude our survey in "Conclusion". An overview of this review is presented in Fig. 1. The major abbreviations used in this paper are presented in Table 1.

Need for Machine Unlearning

Internet giants like Facebook and Google have gathered enormous amounts of private data on people, including their purchase habits, behavioral patterns, and lifestyle choices. Additionally, these companies use big datasets to train massive machine-learning models for various tasks. However, this data collection goes beyond what is deemed essential, which might endanger users' privacy. Existing privacy concerns have been exacerbated by recent data breaches and scams that have brought to light instances in which users' private information was sold to other parties [14, 15]. Governments have formulated multiple rules and regulations to minimize such misuse of users' data. Some of the examples are the General Data Protection Regulation (GDPR) by the European Union, giving users the "Right to be Forgotten", the California Consumer Privacy Act (CCPA) [16], and the Indian government making it compulsory to store data of Indian users in India only and not to take them out to some other nation, etc. [17–21]. Simply removing data from databases is inadequate to tackle the problem since ML models are constructed to allow them to remember and learn from user input. Because the models still have indirect access to the data, just removing the record entries is inadequate. MUL is a concept developed based on the "Right to be Forgotten," where users' data is completely forgotten from multiple institutions' various ML models [22]. Although privacy is the main reason, the users/institutions may want to delete the data for other reasons, some of which are mentioned below. A pictorial presentation of the need for MUL is presented in Fig. 2.

Security

Organizations are vulnerable to incursions in this digital age, which may lead to data breaches or corruption [23]. This is a major issue since these businesses' ML and deep learning algorithms depend on reliable and accurate data to generate predictions. ML algorithms may provide incorrect predictions in the case of a cyberattack that corrupts data [24]. This is particularly important in industries like healthcare, where making incorrect diagnoses based on contaminated data may have disastrous results and jeopardize patients' lives [25]. Whenever such attacks are detected, the organization should be capable of deleting such data from the "learning" of its ML models so that the models can remain robust.

Privacy

Everyone has the right to privacy [26]. In a democratic setup, privacy is a fundamental right, and governments, companies, and organizations have no right to infringe on users' privacy. The GDPR again reestablished this principle by giving the citizens (their data) the "Right to be Forgotten." In this right, users can instruct the companies to delete their information from their databases, including deleting data from ML models. To fulfill this, MUL is required.

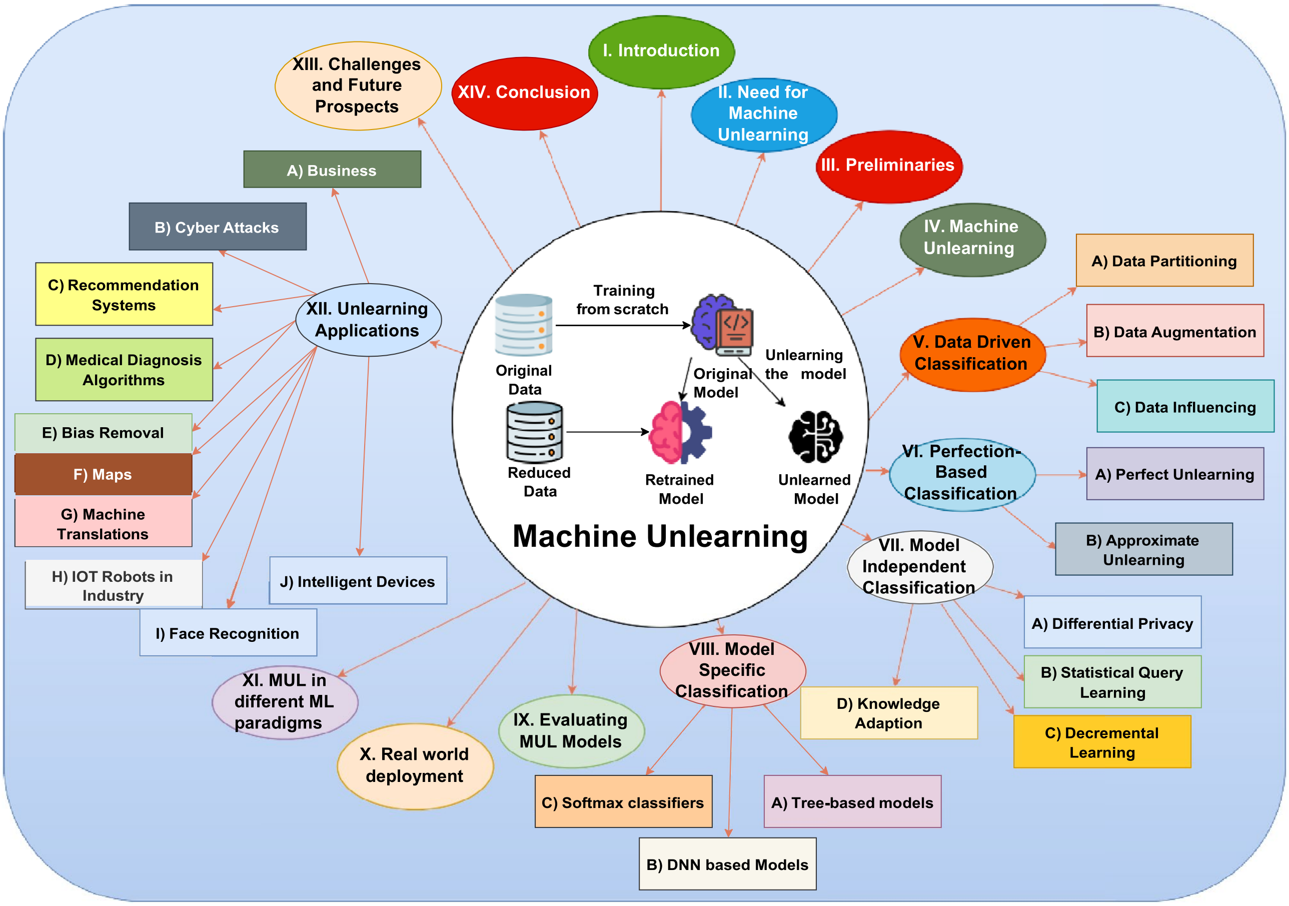

Fig. 1 Overview of this survey

Usability

Data are often accompanied by noise or other information. Businesses must clean up their databases of unnecessary data to provide consumers with a better application or platform experience. By freeing up storage space, deleting unnecessary data conserves important resources and improves the effectiveness of database administration. MUL methods must be used to satisfy these data cleansing needs. Even after many advances in this field, learning models are associated with some bias [27, 28]. The models must be revamped in such cases to remove their bias toward something. For example, it has been observed that the software used in automatic cars, like Tesla, has biasedness based on skin color in case of accidents. In the case of the choice of the accident being a white-skinned and black-skinned person, the software is biased towards hitting a dark person more than hitting a white-skinned person [29]. The biased nature of ML algorithms is often a consequence of underlying biases in the data. To increase accuracy and ensure the models' fidelity to the broader population in such situations, models must eliminate biased dataset samples. This technique makes bias less apparent, and more representative and equitable ML models are produced [30].

Preliminaries

Before discussing MUL, we would like to describe ML from the perspective of MUL so that the audience has a greater comprehension.

Machine Learning in Perspective

Computer systems may learn and adapt using ML, which is used to teach them. It comprises the system finding patterns and correlations in the data independently, enabling precise predictions of future occurrences of fresh data. Without explicit encoding for each case, the system can generate precise predictions and understand incoming data by drawing on earlier learning and training [31–33].

In terms of the type of training of the ML models, the models can be divided into two categories:

Table 1 Major abbreviations used in the survey

Notation	Meaning
AI	Artificial intelligence
ML	Machine learning
MUL	Machine unlearning
SISA	Sharded, isolated, sliced, and aggregated algorithm
GDPR	General data protection regulation
DNN	Deep neural network
FL	Federated learning
ANN	Artificial neural network
CNN	Convolutional neural network
IoT	Internet of Things
SVM	Support vector machine
RNN	Recurrent neural network
LSTM	Long short-term memory
GRU	Gated recurrent unit
RL	Reinforcement learning
MSE	Mean squared error
DP	Differential privacy
AIN	Anamnesis Index
ZRF	Zero retain forgetting
SHAP	SHapley Additive exPlanations
LIME	Local interpretable model-agnostic explanations
GPT	Generative pre-trained transformer
PII	Personally-identifying information
T5	Text-to-text transfer transformer
NLP	Natural language processing
MAE	Mean absolute error

1. Adaptive — A model is built using an adaptive training technique, and the model's past actions and learned data points decide its upcoming steps. When faced with new situations, the program chooses model modifications depending on the model's previous state. Stochastic gradient descent is a technique that leverages the existing classification model and suggests tiny incremental steps to reduce the loss [34]. For model training, the industry often uses adaptively trained algorithms [35]. However, it is possible to unlearn these models using specific techniques. Certain methodologies emphasize the whole model class's probabilistic unlearning. Another method involves starting with the original model and then optimizing it with each data point, omitting the one that has to be unlearned until convergence is reached. This process produces an unlearned model. It is important to note that the resultant unlearned model may not achieve 100% accuracy for complicated models like neural networks [36].
2. Non-adaptive — Unlike adaptive training algorithms, non-adaptive training algorithms usually use techniques like gradient descent with a single step to predetermine their learning updates [37]. Non-adaptive models are less common, although unlearning them may improve accuracy. The impact of a particular data point may be identified in non-adaptive algorithms. Unlearning requires removing the effects of the data point, which may be done using strategies such as influence functions [38], as detailed in the following parts of the article. A model is built using an adaptive training technique, and the model's past actions and learned data points decide its upcoming steps. The program chooses model modifications depending on the model's previous state when faced with new situations. Stochastic gradient descent is a technique that leverages the existing classification model and suggests tiny incremental steps to reduce the loss [39]. For model training, the industry often uses adaptively trained algorithms. However, it is possible to unlearn these models using specific techniques. Certain methodologies emphasize the complete model class's probabilistic unlearning. Another method involves starting with the original model and then optimizing it with each data point, omitting the one that has to be unlearned until convergence is reached. This process produces an unlearned model. It is important to note that the resultant unlearned model may not achieve 100% accuracy for complicated models like neural networks.

Machine Unlearning

The process by which a system eliminates the impact of a previously learned data item gathered through a machine-learning algorithm is known as MUL. In an ideal world, the model forgets or unlearns all the data points contributing to the learning process. MUL aims to eliminate or lessen the influence of specific data points on the learned model, allowing more adaptive and flexible model behavior [40]. If a particular data point is presented again in the future, the computer is technically capable of forgetting it and considering it as a new data point. An unlearning algorithm [41], the resulting unlearned model, optimization criteria [42], assessment metrics, and a verification method are all parts of the unlearning process. The unlearning algorithm creates the unlearned model, which may need further tuning for the best performance. Optimization strategies are crucial since the unlearning algorithm may harm the model's effectiveness or take a long process. The model's validity and robustness are evaluated using evaluation metrics.

To confirm the model's integrity and preservation of the relevant learning knowledge, it is then verified using test data (Fig. 3). A sample MUL framework is shown in Fig. 4. There are many types of unlearning requests that a server (which possesses and serves an already trained ML model)

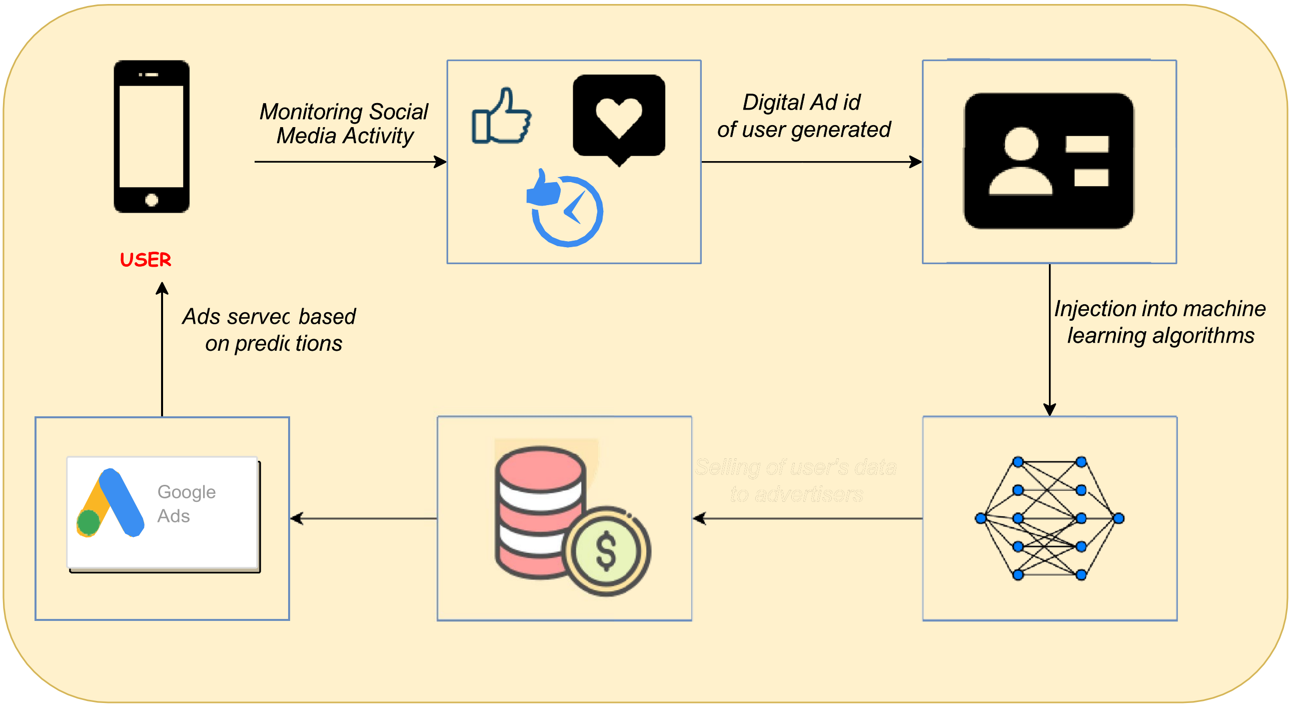

Fig. 2 Need for machine unlearning: The likes, preferences, and much more information about the user are all being fed to ML models to predict the topics of interest, which are in turn used for serving them the targeted advertisements

may receive in day-to-day functioning from its users. Some of them are described as follows:

1. Item removal: The request to remove specific samples or items from the data set. It is the most common type of request, which the model entertains [43].
2. Feature removal: Sometimes, privacy leaks originate in a data group that resembles a particular feature or label [44]. For example, with the growth of gender awareness and government rules, it is not legal to classify someone based on gender. Hence, industries have to unlearn the users' gender, which is also a feature representing a batch

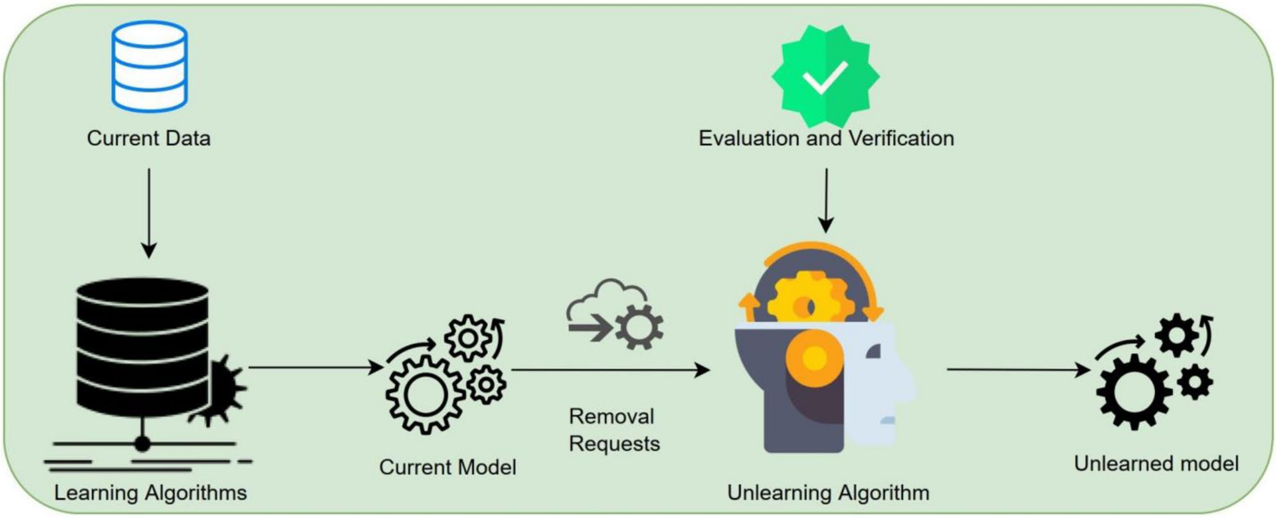

Fig. 3 A machine unlearning framework. Upon receiving the data removal requests, the unlearning algorithms unlearn the concerned data samples. The unlearned models are evaluated and verified for data deletion

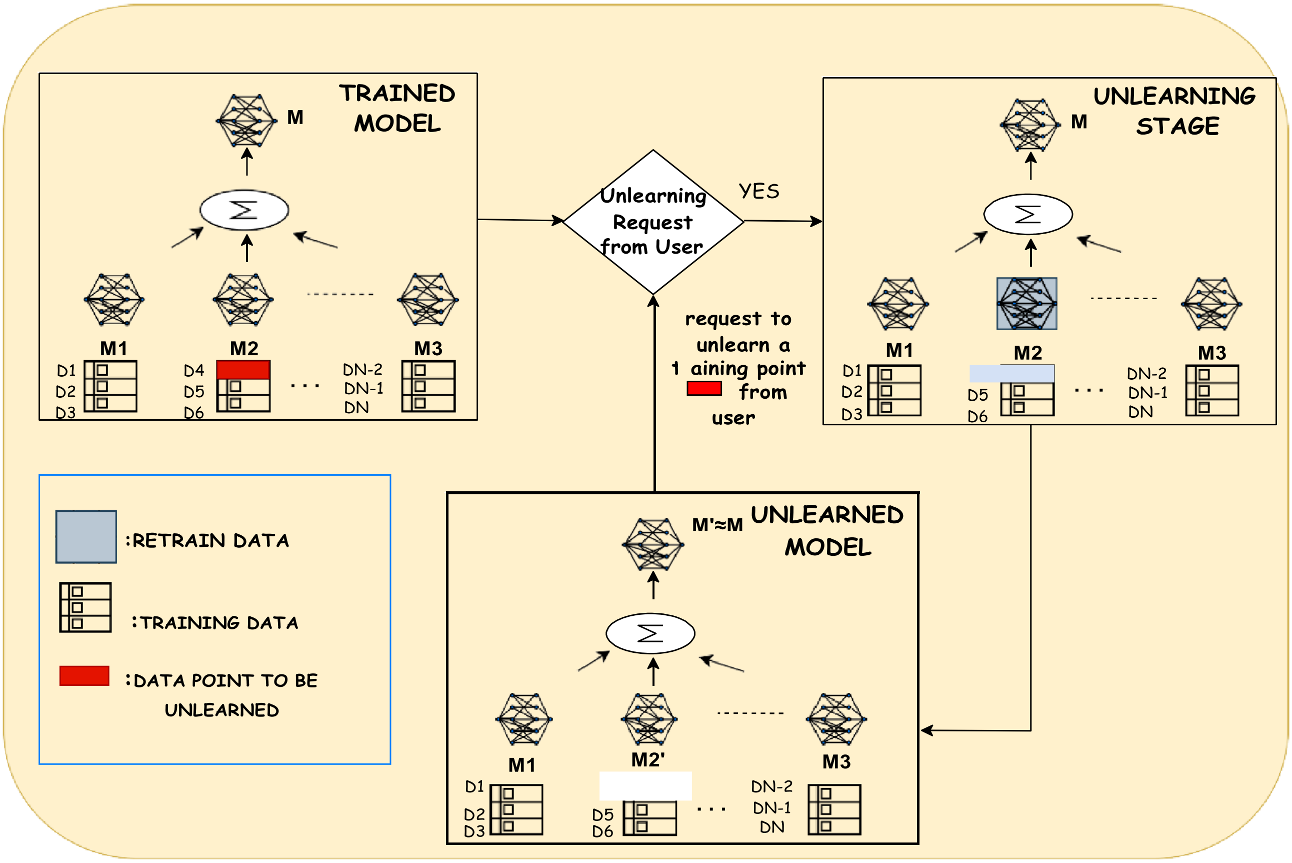

Fig. 4 SISA algorithm. With the help of data partitioning into shards and slices, the models are selectively re-trained to achieve machine unlearning

of people. In such cases, retraining is costly. At the same time, unlearning a few parts of a dataset might lead to a significant loss of understanding of the model, leading to bad performance. Therefore, a model has to be developed to recognize and delete the feature. It can be done based on the influence function. Influence functions calculate the influence of a particular data item on the model, and they are stored in the database for further usage. Hence, we could calculate the influence of a feature on the model and delete its influence on the model [45, 46].

3. Class removal: There may be scenarios when a request to delete data is pushed that belongs to single or multiple classes. For example, in face recognition applications, each face represents a class. Hence, deleting a look when the user ceases to use that application comes under this category. One of the ideas proposed is to do it through data augmentation [47]. The idea suggested is to introduce noise to maximize classification error for the target classes. The model is then updated by training it on this noise without accessing the target class(es) samples. Since doing it would change the model's weights and corrupt the performance, a repair step would be needed in which the model would be trained in small portions of the remaining data.
4. Task removal: Nowadays, ML models are trained on multiple tasks instead of being trained on single studies, which is sometimes referred to as "continual learning" or "lifelong learning," which has been influenced by the human brain [48]. By learning many tasks simultaneously, the model can correlate various tasks and overcome the data sparsity problem [49], wherein the model has to start from scratch (in case of a completely new task/problem). Although multi-task learning is good [50], sometimes, in such scenarios, due to privacy requirements, the machine must forget a previously learned task. For example, if a robot assists someone during medical treatment, it might be asked to forget the experience after the patient has recovered. In such cases, unlearning becomes a part of lifelong learning [51]; as every time the patient becomes healthy, the robot will have to unlearn itself, and as soon as a new patient arrives, the robot has to be ready to entertain him and take his care to the fullest, without being intervened by the previous experiences [52].

5. Stream removal: Due to storage constraints, it is sometimes required to forget a massive chunk of data that arrives in the system online [53]. The machine has to decide whether to ignore it or retain it. The devices involved in handling streams of data have to learn and unlearn the data continuously so that the robustness of the model is maintained and, at the same time, storage usage is also minimized. This is helpful because sometimes, adversarial data might be inserted into the systems through a data stream.

A model can achieve the end goal of MUL in multiple ways (Table 2).

In the following sections, we classify the different MUL algorithms based on other criteria — data based, exactness of unlearning-based, ML paradigm-based, and ML model-based. Table 3 summarizes the device unlearning algorithms and techniques reviewed in this work and their advantages and disadvantages.

Approaches by the Exactness of Unlearning

In this section, the unlearning algorithms are classified based on the exactness/perfection of unlearning achieved.

Perfect Unlearning or Exact Unlearning

Perfect or exact unlearning algorithms [57] would give the model, which would be the same as the model obtained if we train the model on the data set excluding a particular data point, which the model has to unlearn. This is the ideal situation, and it is generally challenging to obtain such models. Due to this, retraining is considered the only exact unlearning algorithm. However, it is difficult to retrain, as sometimes the data set could have billions of data points, and entertaining every data deletion request will consume time. If the frequency of data deletion requests is high, it would be impossible to retrain the model on a particular data point before the subsequent deletion request comes in. Furthermore, the retraining of the models also involves high computation costs. Some models, such as Sharded, Isolated, Sliced, and Aggregated Algorithms (SISA) VI-A [72], give perfect unlearning models, but it depends on retraining. In the case of non-adaptive learning, achieving 100% accuracy is still possible, as the model is not incremental, but in an adaptive model where learning is incremental, achieving 100% accuracy is very difficult and sometimes impractical as well. Nowadays, most ML models are adaptive. For example, deep neural networks learn from every data point and accordingly adjust its parameters (weights), due to which perfect unlearning is very difficult.

Approximate Unlearning

When it comes to unlearning algorithms, achieving 100% accuracy can be a challenging endeavor. A recent study by Neel et al. [60] sheds light on approximate unlearning algorithms, which typically attain an accuracy rate of approximately 80–90% in the unlearning process. This means that in about 10–20% of cases, the model may retain traces of previously deleted data. However, this trade-off between accuracy and computational cost offers a practical solution. Approximate unlearning is more cost-effective than exact unlearning, and it generally results in a model that has primarily forgotten undesirable data points. This method proves particularly useful when dealing with complex and adaptive ML algorithms, where it may be impossible to reconstruct the exact order and impact of data points on the model. Although techniques like influence functions have been developed to estimate the influence of individual data points, achieving perfect unlearning remains an elusive goal.

Data-Driven Unlearning

This group of MUL algorithms unlearns a given ML model by either (a) training on selective data, (b) enriching it with additional noisy data, or (c) studying the influence of specific data points on the ML model parameters and later removing that influence from the model on request. This data-driven MUL algorithms are described below.

Data Partitioning [73]

Bourtoule et al. [43] proposed the SISA (Sharded, Isolated, Sliced, and Aggregated) algorithm, which is used to unlearn a data item from a trained model. After removing the items to be deleted, the model is again retrained on the remaining dataset. The training algorithm is designed so that the complexity of re-training the model is less compared to the original training of the model. In this model, the data set is divided into various shards (let us say S shards), meaning that the data is divided into S parts, with a single data point present in only one of the shards [74]. The data is not repeated in any of the shards, nor is duplicated in any other shard, satisfying the “I” (isolated) condition. The shards are again divided into slices (each shard is divided into R slices). Like the shards, the data is not duplicated or repeated in any of the pieces. Training of the model is done shard by shard and then aggregated into a final model. Training of slices is done differently. Initially, the model is trained on the first slice, and then on the first and second slices, then on the first three slices, and in a similar way, all R slices are trained. The progress (parameters of the model at that particular stage of training) of

Table 2 Machine unlearning algorithms and techniques, their advantages and disadvantages

Machine unlearning method	Ref.	Advantages	Disadvantages
Data partitioning	[43]	Gives a perfect unlearning model with 100% accuracy	Has higher space and time complexities
Data augmentation	[54]	In cases of adversarial attacks, the data is not affected, as the machine refuses to learn anything from the data, keeping the user's data and organization's data safe	i) Can only be applied to the systems prone to cyberattacks ii) Cannot be used to process users' requests to unlearn their private data
Data influencing	[44, 55, 56]	The process is easy to apply, as one has to delete certain calculations which were made at the time of training of data	i) Has high space complexity as all the calculations must be stored ii) Model can be inaccurate as sometimes the influence of a particular data item depends upon the order in which it appeared in the dataset
Perfect or exact unlearning	[43, 57–59]	It gives fully accurate results, which are obtained when the data is retrained on the new data set	It is computationally costlier and sometimes practically impossible because the data to be removed might be just in hundreds (in number). Still, the actual data may be in billions
Approximate unlearning	[60, 61]	i) Practically feasible ii) Less time consuming iii) Computationally inexpensive	It cannot guarantee a user that their data will be completely deleted
Differential privacy	[36, 62]	It can maintain the tradeoff between privacy and the accuracy of results, giving models that are as accurate as possible along with maintaining privacy	i) This model is approximate ii) This model cannot provide good results when complex queries are put before the model
Statistical query learning	[43, 63]	This technique can be applied to multiple ML models such as linear regression and naive Bayes	It does not work well for complex models like deep neural networks, and the time complexity becomes exponential in case of complex queries
Decremental learning	[57, 64, 65]	The model remains unchanged when there is a small change in data	i) It is not an exact unlearning method ii) It can only process items unlearning requests
Knowledge adaptation	[66]	It can be applied to a wide range of MUL requests and scenarios	It can only process item requests
Unlearning for tree-based models	[67, 68]	This method is very efficient as it is made up of an ensemble of decision trees, and the splits and cut-off threshold are optimized using Gini and entropy measures	If the forgotten set is too large, then the divisions will become non-robust
Unlearning for DNN-based models	[41, 56, 60, 69, 70]	Data influencing method is used to address the complexity of unlearning in DNN models	Two different types of strategies are used for different DNN-based models
Unlearning for Softmax classifier	[71]	Computationally efficient	This method is only applicable for class removal

Table 3 Machine unlearning evaluation techniques, their description, and various usages

Evaluation metrics	Description	Usage	References
Accuracy	Accuracy on forget dataset, retained dataset, and test dataset	Evaluates the performance of the model based on its prediction capabilities	[55, 70, 104, 105]
Completeness	Comparing the results of the unlearned model and the retrained model	Evaluating the differences between the two models	[63]
Unlearn time	Amount of time taken to unlearn the model	Evaluating the efficiency of the unlearning mode	[63]
Relearn time	Amount of time required by the unlearned model to achieve the accuracy of the source model	Evaluating the efficiency of the unlearning mode	[63, 105]
Layer wise distance	Evaluating the weight differences between the original model and the retrained model	Evaluating the differences between the two models	[105]
Activation distance	Separation between the final activation of the scrubbed weights and the retrained model	Evaluating the differences between the output of the two models	[66, 70]
JS-divergence	Divergence between the predictions of the retrained model and the unlearned model	Evaluating the differences between the output of the two models	[66]
Membership inference attack	Ratio of number of detected items and number of forget items	Verifies the influence of the forget dataset on the unlearned model	[61, 70]
ZRF Score	Compares the unlearned model independent of the retrained model	1) ZRF score is 0 if the model intentionally gives the wrong output 2) ZRF score is 1 if the model is not giving random output on the forget dataset intentionally	[66]
Anamnesis Index (AIN)	The value ranges between 0 and infinity	Closer the value is to 1, the better the unlearning performed by the machine. If the value is closer to 0, then the hidden information is not removed, and a higher value suggests that some features or parameters have been deleted or created during the process	[55]
Epistemic uncertainty	Measures the confidence of giving optimal results on the new dataset	Measures only the overall information reduction and not the specific information changes were done during unlearning	[106]
Model inversion attack	Depends upon visualization of the user	Qualitative method and often used in image processing	[61]

each slice of training is stored. Whenever the user requests to delete a data sample, it is located in which the shard and slice of the data are stored. All other shards remain unaffected by the user's requests. Retraining is done from the slice from where the data is removed. Since each slice's progress is stored, training does not take much time. All those slices whose model is changed are updated, and finally, the models of each shard are aggregated to form the model. A disadvantage of this algorithm is its large space complexity. Still, it gives a perfect unlearning model, in which there is not even a trace of the user's data, which he requested to delete, valuing the user's privacy to the fullest. The SISA algorithm is presented in Fig. 4.

Data Augmentation

Generally, it adds more data to support the model training. In MUL, data augmentation introduces noise in the data to convince the model that a particular sample (which we want to forget) does not exist [54]. In this, the model is tricked into believing that nothing can be learned from a given data. It is mainly used when the model is prone to attacks like cyberattacks [47, 75].

Data Influencing

This technique, proposed by Warnecke et al. [44], is used to study how a model changes in the training data and use this information for unlearning some particular data. It is done by using influence functions. These influences (of data) are calculated and stored. Whenever a data sample has to be deleted from the records, then the unlearning of that sample boils down to just deleting the data stored in that particular influence function. This model can be inaccurate because, sometimes, the influence of a specific data item on the model depends upon the order in which it appears in the training data.

Model Independent Unlearning

Model-independent methods enable unlearning across a wide spectrum of machine-learning models. In machine learning, it is worth noting that many evaluation metrics, including precision, recall, and mean squared error (MSE), inherently operate independently of the model. Additionally, these methods are agnostic to the specific training dataset employed during the model's training phase. Some of them are mentioned below.

Differential Privacy

Differential privacy [62, 76] guarantees the outcome of a calculation to be insensitive to any particular record in the dataset. It works on the principle that if a minor substitution is made in the database, the query result would not be able to identify much information about a single individual. It was introduced to store the influence of a particular data sample on a machine-learning model. Some of the approaches proposed are also of an adaptive nature, meaning that the data that has to be removed depends on the current unlearned model. This model is approximate. It can process item and stream unlearning requests but cannot process feature, class, and task unlearning requests.

Statistical Query Learning

Statistical query learning [43, 77, 78], an ML algorithm, trains models by querying statistics on the training data instead of querying on the data itself. This technique assumes that most ML algorithms can be represented in the form of some efficiently computed transformations called statistical queries. These queries are requests to an oracle to estimate the statistical functions of comprehensive training data. This technique does not work well with complex models, such as deep neural networks, which is a disadvantage of this technique. The number of statistical queries in complex models would become exponentially large, making unlearning and relearning steps less efficient. This model also gives approximate unlearning. It can process item unlearning requests.

Decremental Learning

In the decremental learning [57, 79], the data is partitioned and quantized using the k-means learning algorithms [80]. It is done so that a slight change in the data does not affect the model. To prevent the accuracy from being significantly reduced, it removes unnecessary unlearning requests. It is not an exact method and can only process items unlearning requests [81].

Knowledge Adaptation

Knowledge adaptation [66] selectively removes the data samples that must be forgotten. In this approach, two teacher neural networks (one competent and the other incompetent) and one student neural network model are developed. The competent teacher model is trained on the reduced dataset (forget the sample deleted from the entire dataset), and the incompetent one is randomly initialized. The student is initialized in such a way that it resembles both the teachers through a loss function having KL-divergence evaluation values between teachers and students [82]. The competent teacher model deals with the retrained data (data after deleting a particular sample), and the incompetent teacher model deals with the forgotten dataset. It is an approximate method and can process item requests only (Fig. 5). The incorporated knowledge adaptation technique to achieve unlearning is presented in Fig. 6.

Model-Specific Unlearning

A few MUL algorithms are specially developed for unlearning in some particular types of ML models. However, these algorithms can also be used in other models if they fit there. In this section, we present some of these MUL models.

Unlearning for Tree-Based Models

A tree-based ML model [83] divides the feature space to form a tree so that every piece of information belongs to a particular split region. A better and more efficient tree could be made using classification techniques like the Gini index or entropy [84]. The MUL method for tree-based models (Hedgecut [67, 85]) is based on the query that if some k several data items are removed from the data set, then it would reverse the split (of feature space) in the tree [86]. The degree of reversal of split due to deletion gives the measurement of the robustness of the tree. As a result, the trees are designed so that most splits are as robust as possible. One disadvantage in these models is that if the forgotten set is too large, the splits will become non-robust. The working of Hedgecut is presented in Fig. 5.

Unlearning for DNN-Based Models

It is challenging to unlearn data from deep neural networks (DNNs) [87] because DNNs are neural network-based, automatically learning features from the data. Existing unlearning models can be applied for layers with convex activation functions [88, 89]. Still, for non-convex layers, a caching approach is used in which the model is trained on data,

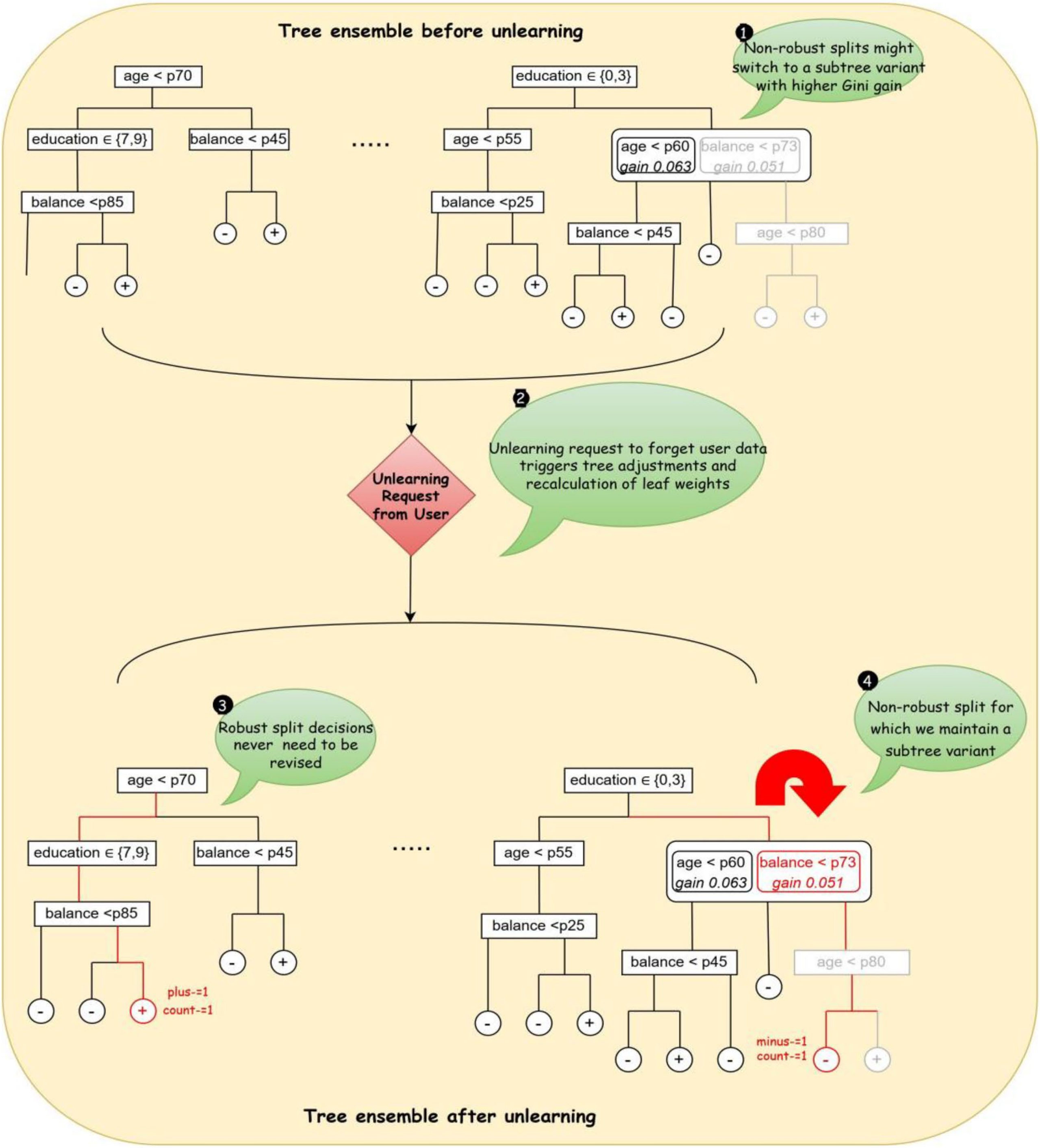
Tree ensemble before unlearning
age < p70
education ∈ {7,9}
balance < p45
balance <p85
education ∈ {0,3}
age < p55
age < p60 gain 0.063
balance < p73 gain 0.051
balance <p25
balance < p45
age < p80
1 Non-robust splits might switch to a subtree variant with higher Gini gain
2 Unlearning request to forget user data triggers tree adjustments and recalculation of leaf weights
Unlearning Request from User
3 Robust split decisions never need to be revised
4 Non-robust split for which we maintain a subtree variant
plus-=1
count-=1
minus-=1
count-=1
Tree ensemble after unlearning

Fig. 5 Hedgecut: machine unlearning for tree-based classifiers

which is known to be permanent, and then user data is fine-tuned to it, using some convex optimization [69, 90, 91].

Unlearning for Softmax Classifiers

The softmax function [92] converts a vector of k real numbers into a probability distribution of k possible outcomes. This method is a generalization of the multiple logistic regression function used in multinomial logistic regression. Unlearning algorithms proposed for the softmax classifiers [71] are based on a linear filtration operator, which proportionally shifts the classification of samples of the forget classes to other classes. This method can be used only for class removal requests.

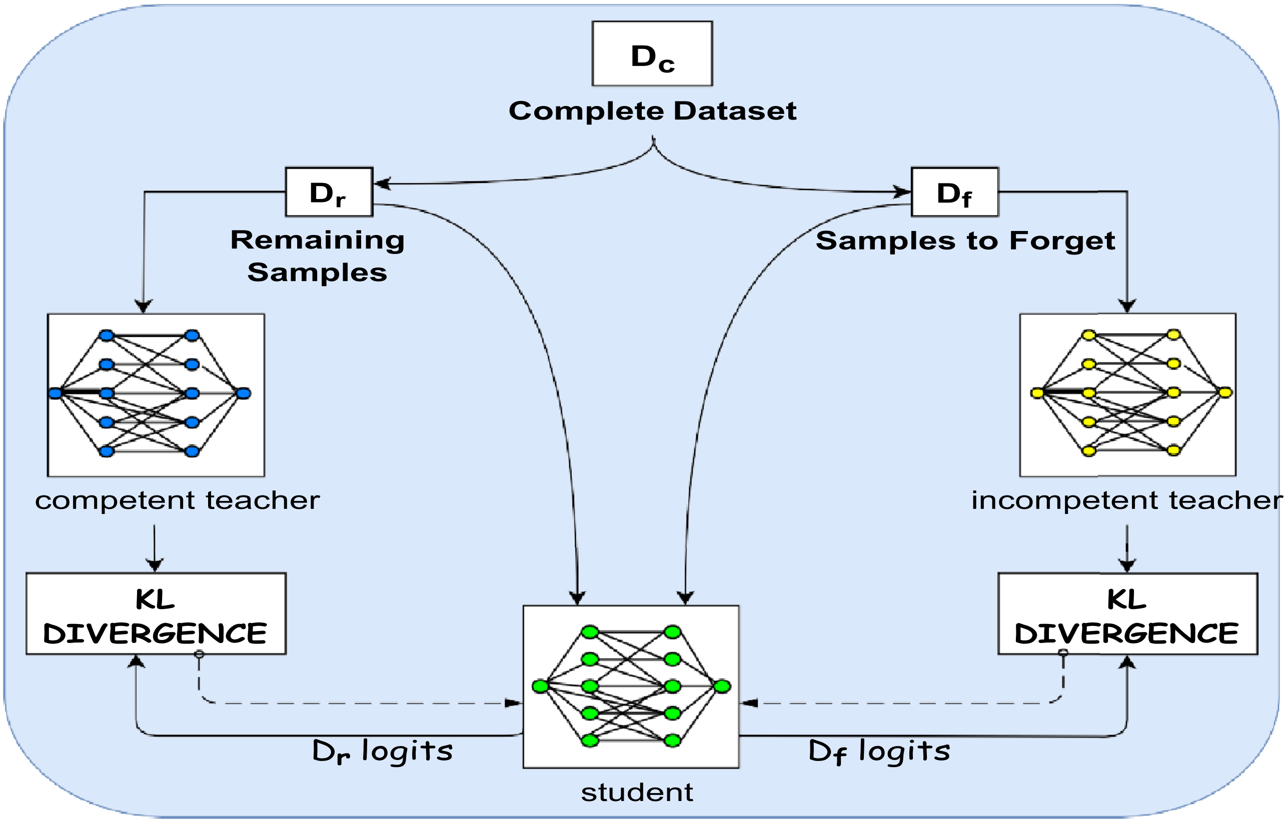

Fig. 6 Knowledge adaptation method. The student model is prepared to mimic both the teacher models by the loss functions with KL divergence evaluation values between teacher and student (Logits are probabilities)

Evaluation of Machine Unlearning Models

It is essential to evaluate to what extent the unlearning machine models could forget a given dataset. This section identifies and discusses various metrics the researchers use to assess unlearning machine models.

Accuracy

The accuracy metric is utilized to evaluate the output labels generated by the ML model to the actual labels [93]. The accuracy of an unlearning machine model is determined using the three distinct datasets listed below.

Forget Dataset The”Forget” dataset contains examples intended to be forgotten or unlearned. The objective of un-learning is to ensure that the unlearned model exhibits the same behavior as a model initially trained using the unlearned dataset. In other words, the unlearned model must replicate the performance and attributes of a model that has never been exposed to the Forget dataset samples.

Retain Dataset The objective of the unlearning procedure is to maintain the efficacy of the resultant MUL model on the retained dataset. It is crucial to ensure that incorporating samples from the retained dataset does not degrade the model’s performance during the unlearning process.

Test Dataset The test dataset comprises examples used to evaluate the effectiveness of the initial machine-learning model. On the test dataset, which was constructed particularly for evaluation purposes, it is expected that the model will perform nicely and generate positive results.

Completeness

To guarantee exhaustiveness, the unlearned and retrained models must generate identical predictions for every possible data sample. The intended-for-removal dataset must have zero influence. This would indicate that the effects of the purged datasets have not been fully eliminated and that the unlearning process of the model is incomplete.

Relearn Time

The relearning and unlearning periods of the model must be evaluated and assessed. The unlearning procedure must be used if the model cannot sustain performance even after many retraining steps. The relearning period of the model must be substantial to execute unlearning methods successfully.

Layer Wise Distance

The layer distance between them quantifies the discrepancy between two ML models' weights or parameters. The effects of each stage of the unlearning process may be seen by analyzing the layer-wise distance between the unlearned and original models. A lower layer distance indicates insufficient unlearning, whereas the Streisand effect and the potential for information leaking are indicated by a more significant layer distance [94].

Epistemic Uncertainty

Epistemic uncertainty, which is typically brought on by a lack of training data, is standard in ML models [95]. The under-representation of minority groups, such as the tribal people, in face recognition algorithms is an example of this problem. The amount of confidence that the model is suitable for new data sets is determined by the epistemic uncertainty metric, which quantifies our understanding of the best hypothesis in the parameter space. It should be noted that this measure does not require the existence of the retrained model and instead focuses on the overall reduction of information rather than the particular reduction linked to the ignored dataset.

Membership Inference

A programmer may detect if a particular data point was used in the training of a model using a membership inference attack. The privacy and security of sensitive data may be at risk since this attack makes the existence or absence of specific data items inside the training set of the model public [96–98]. Finding out if any data from the disregarded set is still present in the sample is the goal of membership inference. The unlearned model should be more resistant to inference attacks than the original model, especially when class-related data is excluded. The model's privacy and security will be increased due to this decrease in vulnerability [99].

AIN

An indicator used to evaluate the effectiveness of the unlearning approach is the Anamnesis Index (AIN) [100]. It establishes the amount of data from the disregarded set still in the unlearned model. Greater values indicate less successful unlearning; AIN values range from 0 to infinity. A number close to 1 means the unlearning approach completely eliminated the forgotten set's knowledge. But let us say the AIN number is much higher than 1. The Streisand effect, when efforts to delete information draw greater attention, occurs due to the unlearning algorithm's modifications to the model parameters being perceptible and possibly detectable.

Model Inversion Attack

Unauthorized individuals get access to the private data needed to build supervised neural network models to recover it in model inversion attacks [101, 102]. It is a subjective measure often used in image processing and depends on the viewer's visual perception.

ZRF Score

An unlearned model may be independently evaluated using the zero retain forgetting (ZRF) score. This measure compares the output distribution of the unlearned model to that of a model with randomly initialized data, often known as the ineffective teacher, in the context of knowledge adaption strategies [66].

Analyzing Machine Unlearning Models: Real-World Deployment

In this section, we analyze the deployment of MUL models in terms of different parameters like latency, on-chip v/s off- chip deployment.

Latency and Real-Time Processing Capabilities

Latency and real-time processing capabilities are essential when implementing a machine-unlearning model in practical situations. The length of time it takes a model to respond to queries is referred to as latency. The model must be capable of real-time processing and low latency. The duration needed to unlearn a unit data set is latency in the context of MUL algorithms. Models based on retraining often have significant latency since they have to be trained on the whole data set and the excluded samples. These retraining models are challenging to use in real-world applications where processing time is an issue because of their high spatial and temporal complexity. The retraining-based category, which may cause substantial delay, includes models of complete unlearning. While approximation unlearning algorithms have reduced latency, they sacrifice privacy. These rough models may not be able to ensure the total deletion of user

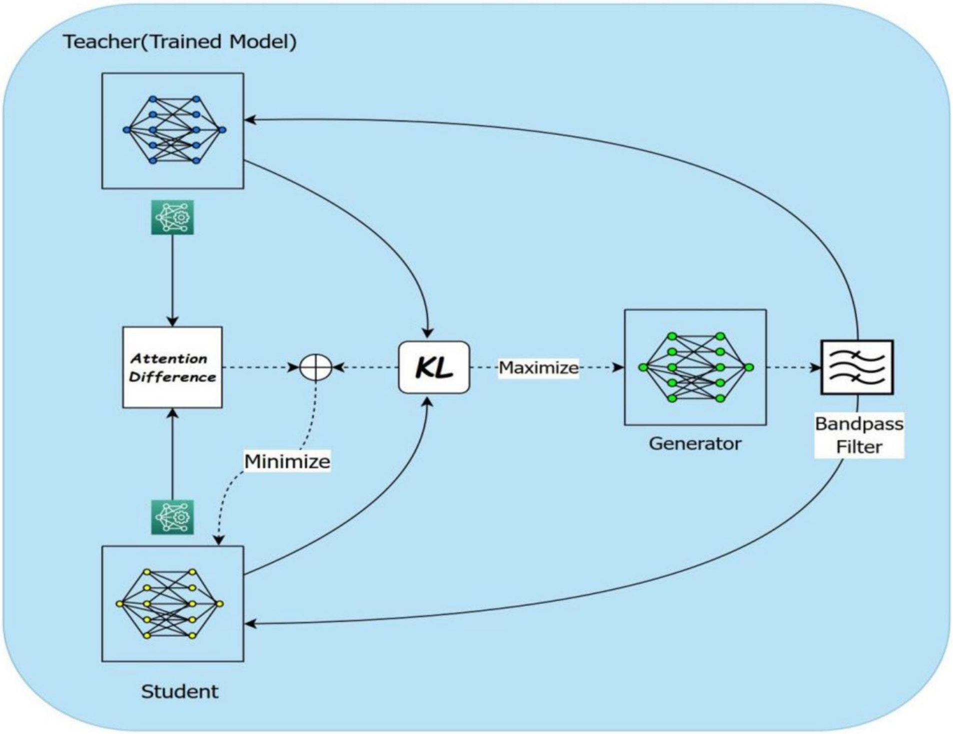

Fig. 7 Zero shot machine unlearning. It is an algorithm that relies on knowledge distillation to perform better in a situation without data. The bandpass filter in this context is referred to as the gate, and it prevents the transfer of forget-class data from the instructor to the student model. Data points are generated using a generator to optimize KL divergence between teacher and student models

data or the absence of any traces of user data. However, approximation models are more suited for application usage owing to their decreased latency and improved real-time processing capabilities. Recent studies on zero-shot MUL [103] have shown promising results in lowering latency for unlearning specific data samples (shown in Fig. 7).

Time and Space Complexities and Energy Consumption

Algorithms for partial retraining, in which the ML model is retrained on the retained dataset, are provided by frameworks like SISA. In comparison to ideal retraining models, this strategy is faster. However, it has been noted that SISA still has more space and temporal complexity, which could make it more challenging to use in actual situations [43].

It is crucial to recognize that algorithms with high space and temporal complexity use more resources, are less environmentally friendly, and leave a bigger carbon footprint. Exact unlearning models, retraining models, and even partial unlearning and retraining models like SISA fall under this category. On the other side, there are not many algorithms that demand less time and space. Even approximation unlearning-based algorithms, which can seem safer, do not necessarily abide by different nations' privacy regulations and might not satisfy all user needs.

To overcome these challenges, maintain compliance with privacy laws, and provide effective and efficient solutions, more study and assessment are required before unlearning frameworks can be used in real-world applications. ML researchers are working to find ecologically responsible and eco-friendly algorithms with small carbon footprints [107, 108]. Deep neural networks (DNNs), which are often utilized in sophisticated applications, are complex, which makes unlearning difficult [109]. Researchers are looking for ways to streamline the unlearning process and minimize the energy use and environmental effects of DNNs. A more ecologically friendly and environmentally sustainable approach to ML will be made possible by developing effective unlearning algorithms [110]. High spatial complexity DNN models are hard to unlearn since they are common in the industry. Implementing the unlearning model in DNNs is ineffective and unable to handle real-time processing. Unlearning algorithms are also ecologically unfriendly because of the complexity of DNNs, leaving a bigger carbon footprint.

Off-Chip and On-Chip Deployment

On-chip technologies are internet-independent, locally stored algorithms, and data for mobile devices. In contrast, server-client technologies for off-chip storage of data and algorithms are accessible through the Internet. Off-chip systems are more sophisticated than on-chip ones.

An example of an off-chip method is data influencing, which involves determining the impact of each unit data set on the final model. Due to its significant temporal and spatial complexity, this strategy is ineffective for big machine-learning models.

MUL methods based on approximation unlearning may be implemented on-chip due to decreased time and space difficulties. Data stays inside the system with on-chip installations, increasing security, and decreasing the possibility of data breaches or hijacking. Businesses with large datasets and complex models may be unable to consolidate everything on a single mobile device, requiring off-chip deployment.

Security hazards associated with off-chip deployment include data leaks [23], cyberattacks, and dangers including data tampering, membership inference attacks [111], and model extraction. It becomes harder and harder to execute unlearning models off-chip because of significant latency, processing time, and data security issues. Unlearning models get far more sophisticated in off-chip devices.

Incorporating Machine Unlearning Algorithms in Different Paradigms of Machine Learning

This section emphasizes the significance of implementing MUL algorithms into different machine-learning paradigms. We look at the effective integration of these algorithms into different ML paradigms. Building on-chip, low-latency, real-time, privacy-preserving, and domain-independent ML models is becoming increasingly important. Retraining privacy-preserving algorithms from scratch may be computationally demanding, mainly when specific data samples must be disregarded. MUL techniques provide a possible resolution in this situation.

Federated Learning

Federated learning systems [112] use local data storage in businesses or on individual PCs combined on a server to produce a shared learning model. The information does not go to the federated server; it stays on the computer of the business or person. For instance, Indian data rules mandate that businesses keep their data within the nation's boundaries. In these circumstances, the client and the global server construct a communication protocol that includes numerous cycles of weight exchange and stochastic gradient descent updates to local models. Due to the aggregation of models and the potential for data overlap, incorporating MUL into federated learning systems is challenging. It is conceivable that traditional approaches will not be directly practical. To address this, techniques like logging out and removing a client's previous contributions from the global server are used. However, as this might impact the model, additional techniques are used during global server training to minimize client contributions [113, 114].

Lifelong Learning

Numerous ML models are created to learn and adapt continuously, and they are often inspired by the human brain's potential for this. ML models are taught to do numerous tasks and find connections between dissimilar things, much as how people concurrently learn several activities and pick up new experiences over time. This enables them to draw upon existing knowledge rather than starting from scratch when dealing with new difficulties. Unlearning, however, sometimes becomes essential. Think of a robot that manages many tasks while providing medical treatment to a patient, for instance. To protect the patient's privacy during treatment, the patient or the hospital may ask that the automaton ignore the patient's information. This makes sure that the automaton will not unintentionally make mistakes in scenarios that are similar but somewhat different. The robot is designed to simultaneously learn and unlearn experiences in such circumstances, continually adjusting to new knowledge while rejecting outdated information [51, 115, 116].

Ethical Machine Learning

Concerns about privacy, ethics, and user security have increased as ML models are used more often in real-world settings. Legal action has been taken in cases where businesses have misused user data for their own advantage, such as when Los Angeles authorities sued IBM for stealing gathered data [117]. The presence of biased data, which may discriminate against specific communities, ethnic groups, races, or colors, as shown by the Tesla self-driving vehicle example previously described, is another ethical problem. Techniques for unlearning may be quite helpful in dealing with these problems. MUL algorithms may assist in creating ethical standards for ML models built by different organizations and by eliminating biased data from the model and replacing it with superior, unbiased data. Ensuring that the models are fair, transparent, and devoid of discriminatory biases encourages the responsible use of ML technology while protecting user interests [118].

Explainable Machine Learning

The idea of explainable ML enables one to explain what the model does, forecast how it will impact users and who will use it, and explain the model's resiliency in everyday obstacles. When a defective product harms the users, it becomes crucial. Hospitals are a typical example when patients' lives are often on the line, and even a little error might result in fatalities. Another example would be expensive missions, like those of SpaceX, ISRO, NASA, etc., where a little mistake might cost millions or even billions of dollars [119, 120].

Explainability could be of two types, as described below.

1. Post hoc: In this technique, the model is explained after the model has been trained, and some predictions have been made. It is a better technique as this allows us to explain even complicated models. This technique requires some particular libraries like SHAP [121] and LIME [122] libraries of Python to explain the model.
2. Inherent: The ability of specific simplified models to be easily described without needing external libraries or extra models is referred to as inherent explainability. Although these models may not have as much predictive potential as more sophisticated ones, they are naturally interpretable. MUL may indirectly support the concepts of explainability by emphasizing the relevance or importance of specific model inputs by removing particular data sets or sequences. MUL may enhance the model's interpretability and transparency by eliminating data that does not significantly contribute to the model's performance or understanding, making the model simpler to explain.

Green Machine Learning

ML-based models have been deployed to solve critical and complex problems with high speed and accuracy in various domains like education, agriculture, manufacturing, and finance. But recently, many questions have arisen about the external cost and environmental impact of ML models, intense learning models (neural network-based models), which have several layers of interconnected nodes, thus leading to a higher computational cost. In 2019, Strubell et al. [123] estimated that training a single natural language processing (NLP) model emitted five times more carbon dioxide than the average car emission over their entire lifetime. In 2020, Schwartz et al. [124] reported that the amount of computing used to train deep learning models had increased by 300,000 from 2012 to 2019. In 2021, Patterson [125] calculated the energy use and carbon footprint of recent large-scale deep learning models like Meena [126], GPT-3 [127], T5 [128], Switch transformer [129], and Gshard-600B [130]. Every time a user requests data deletion, complex models like GPT must be retrained, which takes time and is inefficient. Unlearning algorithms that can handle this challenge are needed. MUL algorithms can solve the problem by making it possible to delete particular data while upholding privacy and legal requirements. It is important to remember, however, that there are currently few machine unlearning algorithms that are effective in terms of time and space complexity and can satisfy a variety of user needs. Therefore, it is crucial to support research and development in this field to build unlearning algorithms that are more useful and efficient. By doing this, we can ensure that ML models can adapt to shifting privacy standards for data while still providing beneficial user experiences.

Privacy Preserving Machine Learning

The effectiveness of ML models is intrinsically tied to the quality of the data they are trained on. In essence, the saying "garbage in, garbage out" holds true for ML. If the data used to train an ML model is of poor quality, biased, or contains sensitive information, it can lead to unexpected consequences, including privacy breaches.

Recent research has brought to light a concerning issue: ML models, especially large language models, can inadvertently compromise user privacy. For instance, when these models are fine-tuned using private data, they can inadvertently leak confidential information. Imagine a scenario where a language model is trained on a dataset containing private conversations or sensitive documents. Even though the model is not explicitly trained to memorize this information, it may still remember and inadvertently reveal portions of it when generating text.

Additionally, large language models have demonstrated the ability to memorize their training data, including specific examples from that data. This can be problematic because these examples might include personally identifiable information (PII) or other sensitive content.

In response to these privacy concerns, differential privacy has gained prominence. Differential privacy is a technique that aims to protect individual user data within a dataset while allowing useful insights to be extracted. It does this by adding noise or randomness to the data in a controlled manner, making it much more challenging for an attacker to deduce sensitive information about any particular individual.

However, it is important to note that only a limited number of ML models currently provide robust privacy guarantees through differential privacy or similar methods. As a result, until more advanced algorithms that inherently prioritize user privacy are developed, a practical solution lies in using unlearning models.

Unlearning models enable the selective removal of specific aspects of a trained model that may pose privacy or security risks. For example, suppose a language model has inadvertently learned and memorized sensitive information during training. In that case, unlearning can erase or modify the model's knowledge of sensitive data.

One specific approach to implementing privacy measures like unlearning is to utilize differential privacy, as discussed in "Differential Privacy" of the document. This approach provides a systematic and well-established framework for introducing privacy guarantees into ML models.

In summary, the privacy challenges associated with ML models, particularly in the context of user data, are a critical concern. Differential privacy and unlearning are two strategies that can help mitigate these concerns by protecting user privacy while still allowing for the valuable insights that ML can provide. However, ongoing research and development are needed to create ML models that are inherently more privacy-preserving.

Applications of Machine Unlearning

MUL has many applications, from recommendation systems to machine translations. A few important applications are described in this section. Figure 8 presents the related applications.

Business

In business, finance, marketing, operations research, logistics, and strategy formulation, ML is utilized. When companies want to design a new product, ML algorithms now play a significant role in the research process. This discipline has become a hi-fi, high-tech topic due to the extensive use of ML, which has led to the development of extremely complex models whose validity and robustness have not yet been established. Here, MUL could clarify the models, thereby reducing their complexity and providing the research communities with a superior model [131, 132].

Cyberattacks

An aggressive assault on a computer network, its data, its information systems, its infrastructure, etc., is referred to as a cyberattack. ML techniques are used by the majority of servers and systems today to conserve data and boost speed [133]. In many situations, the nature of the cyberattacks includes not just data theft but also penetration of the client's artificial intelligence models. In such cases, a fault containing adversarial data is introduced into the system, from which the model eventually learns [134]. The computer acts irregularly as a result of the hostile data. MUL is used when all hostile data has to be erased, and the model needs to be rebuilt [135].

Recommendation Systems

Utilizing data on users' prior preferences and behaviors, recommendation systems provide recommendations for users. Companies may use MUL techniques to ensure that user data is permanently removed from their ML models if a user asks that their data be destroyed. This method safeguards user privacy and maintains data confidentiality [136].

Medical Diagnosis Algorithms

To diagnose patients, hospitals utilize ML algorithms. Nevertheless, mistakes made by humans or machines might result in patients receiving inaccurate diagnoses for illnesses they do not have. In such cases, it is vital to delete these disorders from the patient's medical records that were wrongly recorded. Using unlearning algorithms, it is feasible to remove these incorrect illness entries from patients' medical records [137–139].

Bias Removal

It has been noted that certain ML models are biased towards specific results. These inaccuracies might be the consequence of biased datasets or human mistakes. Unlearning approaches are used in these circumstances to reduce and eliminate these biases. For instance, it was shown that prejudice existed when it came to incidents involving persons of different skin tones, in particular, ML models used in autonomous cars. The model showed a bias for hitting people with dark skin when there were both people with white skin and people with dark complexion present. Retraining such massive models as those used in autonomous automobile systems is often impossible. Therefore, the best and most practical way to address and correct these kinds of biases is MUL [140, 141].

Maps

Digital maps, like Google Maps, are considered very significant applications and are crucial in today's rapidly expanding economy. The value of maps has increased with the ongoing building of new infrastructure and the ongoing expansion of the economy. To effectively lead users to their destinations depending on their travel preferences, maps must be updated often. MUL techniques are also essential in these circumstances. This is due to the possibility that particular roads may be wholly deleted or repurposed, requiring the computer to completely forget its past understanding of those routes and retrain them using brand new parameters [142].

Machine Translations

ML models are used for language translation jobs, such as in the popular software Google Translate, which gives real-time translations in various languages. But with time, specific phrases and linguistic slang every day in a community could become archaic. Unlearning algorithms are essential in these situations for allowing models to reverse previously acquired words and linguistic patterns [143]. MUL methods may be used to complete this unlearning process. These models may give customers more precise and up-to-date translations by unlearning out-of-date language components.

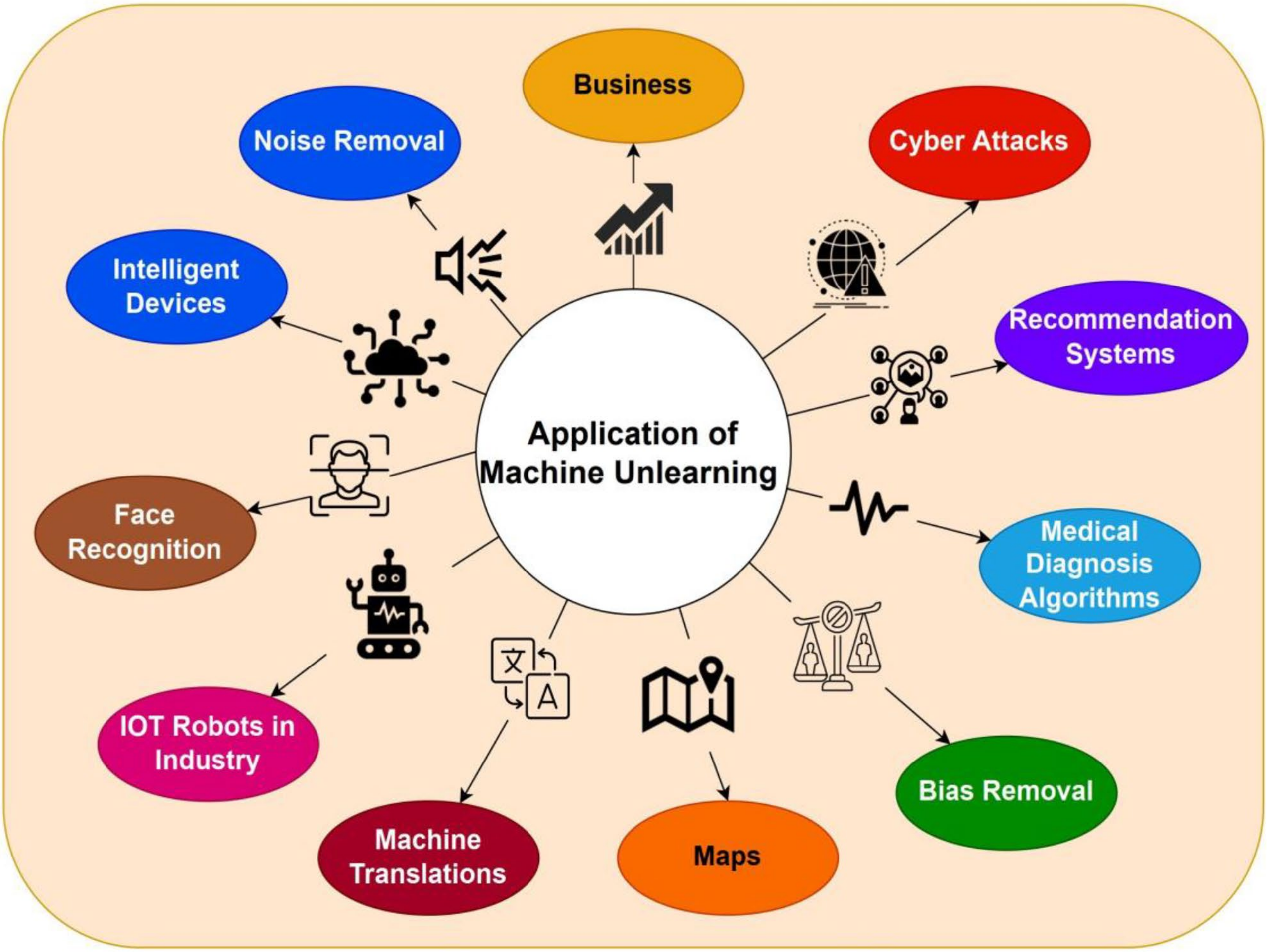

Fig. 8 Applications of machine unlearning

IOT Robots in Industry

It is impressive that IoT devices are trained to identify when industrial equipment needs repair with an accuracy rate of 90%. However, businesses often improve their equipment, making the presently used ML algorithms useless. In these situations, outdated information is discarded using unlearning methods, allowing the models to be retrained using more current data. The training process may be computationally expensive, particularly in sectors like manufacturing, automotive, aviation, and the military, where expensive technology is used for complicated tasks. Unlearning and retraining algorithms are necessary to provide the best performance and flexibility of ML models in these challenging industrial situations [144, 145].

Face Recognition

ML-based face recognition models are often employed in industries to control access to workspaces and prevent unlawful entrance. An employee's information has to be deleted from the list of approved users after they leave the organization. Retraining a model with fresh data may be computationally demanding. In such cases, data of a leaving employee or group of employees may be permanently erased using MUL techniques [146].

Intelligent Devices

ML models are used in intelligent products like Google's Nest thermostat [147], which uses IoT sensor data to train them. Based on the homeowner's preferences, these models are designed to change the climate of a house. The significance of MUL is expanding in light of the present situation of significant variations in seasons, temperatures, precipitation patterns, and other environmental elements. Unlearning helps the model to more quickly adapt to its environment because retraining is not an option owing to the absence of changed datasets that might be used for training in response to the aforementioned climatic changes.

Noise Removal

Background noise removal is the ability by which a noisy sound is removed from the background so that the actual audio signals can be perceived more clearly. Most noise removal techniques use subtractive algorithms, which identify the frequencies of background noise and then subtract it from the original audio to give a high-quality voice. Gogate et al. [148, 149], Hussain et al. [150], and Adeel et al. [151] have achieved noise cancellation in audio signals by using various ML algorithms such as deep neural networks, intelligibility-oriented (I-O) loss functions [150], convolutional neural networks (CNN), and long-short-term memory (LSTMs) and have achieved great results. In this context of noise removal, if any of the datasets on which neural networks are trained to contain misappropriate data, MUL techniques can be used selectively to eliminate the effect of the data samples from the ML model.

Challenges and Future Prospects

This section presents various challenges and future directions in MUL. Figure 9 presents an overview of this section.

Effect of a Data Sample

We often do poorly understand how certain data elements affect a model. We often use approximation approaches to unlearn individual data points, which may not result in an unlearned model that perfectly fits the new dataset. When neural networks are used to train data, this problem is even worse. Finding exactly which layer and at what particular instant a particular data item changed the activation function of the neural network is quite tricky. The unlearning process is more challenging due to the intricacy of neural networks.

Adaptive Training

Most ML models are trained progressively and adaptively, with the impact of earlier data points dictating the importance of later data points. As a result, removing a specific data point becomes challenging since doing so necessitates beginning the training process again when the model uses the data point for the first time.

Brute-Force Retraining

There does not exist a model that gives 100% unlearned model. To obtain 100% accuracy, we still depend on retraining, which is tedious.

Batching Data During Training

Data are usually organized in a certain way during the preprocessing step to enhance model performance. When it comes to unlearning in adaptive models, this may provide serious difficulties. Models sometimes employ tiny amounts of random data selection for training. Since it is difficult to identify the particular moment a given data item first appeared during exercise, unlearning it becomes more brutal in these circumstances [152].

Unified Design Requirements

As per the latest advancements in MUL, there is no absolute method to satisfy all design requirements, including accuracy, timeliness, completeness, etc. Most MUL algorithms are about deleting data items and making an approximate model. A few criteria to consider are methods like zero-shot [153], few-shot [154], and zero-glance [155] types of learning and removal requests like feature, class, stream, and task removal requests. Unlearning models could become industry-grade systems if all things could be unified, and a model or algorithm could be presented.

Unified Benchmarking

The lack of suitable benchmarking datasets and criteria for assessing MUL algorithms highlights the necessity to provide specialist resources in this field. Because there are currently few published resources, and they are usually scattered across different areas, it is essential to provide uniform databases and assessment frameworks.

Adversarial Attacks on Machine Unlearning Algorithms

Attacks on ML are pretty standard, and there are methods to deal with them. However, there does not exist a way by which adversarial attacks on unlearning models could be dealt with. Unlearning models are crucial as they are concerned about protecting users' privacy.

Data Auditing

The primary impetus behind the introduction of MUL stemmed from the need to ensure the complete deletion of specific datasets. Consequently, developing a precise algorithm became imperative to verify the success of the unlearning process. ML models are inherently designed to assimilate datasets and subtly encode them during training. As a result, there is no inherent mechanism for re-verifying whether the targeted

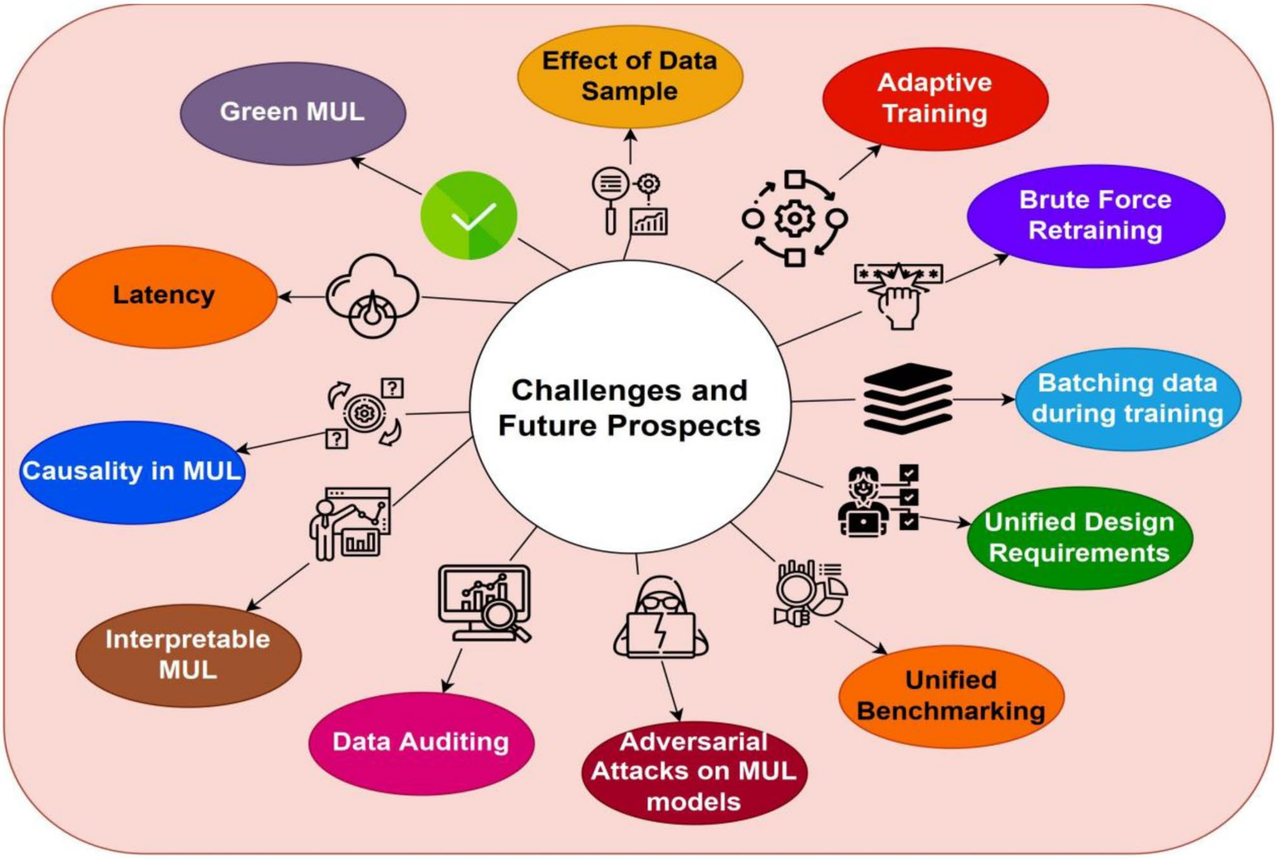

Fig. 9 Challenges and future prospects in machine unlearning

data has been effectively removed. This issue looms large, as there is a possibility that the MUL model could function correctly, yet the desired deletions might not have been executed. Furthermore, a method or algorithm capable of analyzing both the existing and unlearned models and subsequently discerning the disparities between them remains absent at present.

Interpretable Machine Unlearning

The nature of unlearning models of inverting the learned dataset and information may pose challenges for explaining ML methods, making them unsuitable for directly using MUL. More research in explainable unlearning will help develop trust in human and AI interactions and scenarios. Devising techniques that could explain the unlearning task is still an open question in the AI community.

Causality in Machine Unlearning

In some cases, even though the data points to be unlearned are considerably small in size, they may affect the model to a great extent. For example, a data sample in air pollution would be a minimal amount of data, but it generally affects the model significantly. In such cases, causality analysis would become an important tool to learn such data automatically and guarantee that these data points are not included in the final model.

Latency

Latency, in general, means a delay observed between the cause and the effect of specific system tasks, which could be followed physically. Latency in MUL would infer the time taken by the model to unlearn a unit data point. The more the latency, the less practical the algorithm is. Many unlearning models rely on retraining the model on the new data set. In retraining, the latency is very high because the forgotten set is usually insignificant compared to the original data set. The models with high latency could not be used in real-world scenarios. Hence, developing MUL algorithms with low latency is an excellent future direction in MUL.

Green Machine Unlearning

ML algorithms are usually highly space and time-complex, demanding more energy to implement. Since governments focus more on energy conservation and greener and greener ways of doing things, finding efficient ways to implement these complex algorithms has become much more critical. If a broad overview is taken, most unlearning algorithms are highly space and time-complex, thus requiring more energy. Some algorithms, like approximate unlearning, are less space and time-complex. However, they still have some serious issues, like they offer less protection to the user data, we cannot figure out whether the data is completely deleted from the model. There does not exist a green MUL model that could be implemented.

Conclusion

MUL research is gaining popularity for its ability to address issues in ML models, particularly regarding privacy, security, and usability. This technique enables models to selectively forget data, aligning with industry standards that require data deletion based on specific criteria such as items, features, classes, tasks, or streams. Data compartmentalization, achieved by reconfiguring models after data removal, is the most common method in MUL. Researchers employ various techniques to unlearn data. Model-specific unlearning methods are developed for Bayesian, Softmax, DNN, and tree-based classifiers. MUL finds applications in bias mitigation, recommendation systems, and more becoming essential whenever data point removal from training datasets is needed. Challenges and future research directions in MUL are also explored. Overall, this research underscores the growing significance of MUL in addressing limitations within ML models, highlighting its potential for widespread adoption in various domains.

Funding This work has been supported by Engineering and Physical Sciences Research Council (EPSRC) Grants Ref. EP/M026981/1, EP/T021063/1, EP/T024917/1 and ENU Development Trust Ref. LH Oct19'.

Data Availability Data sharing does not apply to this article as no datasets were generated or analyzed during the current study.

Declarations

Ethical Approval This article contains no studies with human participants or animals performed by authors.

Conflict of Interest The authors declare no competing interests.

References

1. Goldsteen A, Ezov G, Shmelkin R, Moffie M, Farkash A. Data minimization for gdpr compliance in machine learning models. AI and Ethics. 2021;1–15.
2. Mourby M, Cathaoir KO´, Collin CB. Transparency of machine-learning in healthcare: The gdpr & european health law. Comput Law Secur Rev. 2021;43:105611.
3. General data protection regulation (gdpr) – official legal text. https://gdpr-info.eu/. Accessed 23 Jun 2023.
4. Everything you need to know about the right to be forgotten - gdpr.eu. https://gdpr.eu/right-to-be-forgotten/. Accessed 23 Jun 2023.
5. Is the 'right to be forgotten' a fundamental right? https://timesofindia.indiatimes.com/readersblog/myblogpost/is-the-right-to-be-forgotten-a-fundamental-right-52529/. Accessed 23 Jun 2023.
6. Voigt P, Von A, dem Bussche, The EU general data protection regulation (gdpr), A Practical Guide, 1st Ed., Cham: Springer Inter- national Publishing. 2017;10(3152676):10–5555.
7. Strobel M, Aspects of transparency in machine learning, in Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. 2019;2449–2451.
8. Lu¨ L, Medo M, Yeung CH, Zhang Y-C, Zhang Z-K, Zhou T. Recommender systems. Phys Rep. 2012;519(1):1–49.
9. Resnick P, Varian HR. Recommender systems. Communications of the ACM. 1997;40(3):56–8.
10. Kaelbling LP, Littman ML, Moore AW. Reinforcement learning: a survey. J Art Intell Res. 1996;4:237–85.
11. Ullman RH. Redefining security. Int Secur. 1983;8(1):129–53.
12. Westin AF. Privacy and freedom. Washington and Lee Law Rev. 1968;25(1):166.
13. Jordan PW. An introduction to usability. Crc Press. 1998.
14. Facebook sued over Cambridge analytica data scandal - bbc news. https://www.bbc.com/news/technology-54722362. Accessed 23 Jun 2023.
15. Google faces mass legal action in uk over data snooping - bbc news. https://www.bbc.com/news/technology-42166089. Accessed 23 Jun 2023.
16. California consumer privacy act (CCPA) — state of California - Department of Justice - Office of the attorney general, https://oag.ca.gov/privacy/ccpa. Accessed 8 Jul 2023.
17. World investment report 2020 — unctad. https://unctad.org/publication/world-investment-report-2020. Accessed 23 Jun 2023.
18. Mutual legal assistance treaties — department of legal affairs, mol &j, goi. https://legalaffairs.gov.in/documents/mlat. Accessed 23 Jun 2023.
19. Data protection committee report.pdf. https://www.meity.gov.in/writereaddata/files/DataProtectionommitteeReport.pdf. Accessed 23 Jun 2023.
20. 4173ls(pre).p65. http://164.100.47.4/BillsTexts/LSBillTexts/Asintroduced/3732019LSEng.pdf. Accessed 23 Jun 2023.
21. Explained: Indian government makes user data collection mandatory for vpns — business insider India. https://www.businessinsider.in/tech/news/. Continual lifelong learning with neural networks: A review. Neural Networks. 2019:113; 54–71.
22. Mercuri S, Khraishi R, Okhrati R, Batra D, Hamill C, Ghasem- pour T, Nowlan A. An introduction to machine unlearning. arXiv preprint. http://arxiv.org/abs/2209.00939. 2022.
23. Ayyagari R. An exploratory analysis of data breaches from 2005–2011: trends and insights. J Inf Priv Secur. 2012;8(2):33–56.
24. Li Y, Liu Q. A comprehensive review study of cyber-attacks and cyber security; emerging trends and recent developments. Energy Rep. 2021;7:8176–86.

25. Sethuraman SC, Vijayakumar V, Walczak S. Cyber attacks on healthcare devices using unmanned aerial vehicles. J Med Syst. 2020;44(1):29.
26. Right to privacy as a fundamental right.pdf. https://loksabhadocs.nic.in/Refinput/NewReferenceNotes/English/Right%20to%20Privacy%20as%20a%20fundamental%20Right.pdf. Accessed 23 Jun 2023.
27. Mehrabi N, Morstatter F, Saxena N, Lerman K, Galstyan A. A survey on bias and fairness in machine learning. ACM Computing Surveys (CSUR). 2021;54(6):1–35.
28. Hellstro¨m T, Dignum V, Bensch S. Bias in machine learning–what is it good for? arXiv preprint. http://arxiv.org/abs/2004.00686. 2020.
29. Study finds a potential risk with self-driving cars: failure to detect dark-skinned pedestrians - vox. https://www.vox.com/future-perfect/2019/3/5/18251924/self-driving-car-racial-bias-study-autonomous-vehicle-dark-skin. Accessed 23 Jun 2023.
30. Grover H, Alladi T, Chamola V, Singh D, Choo KK. Edge computing and deep learning enabled secure multitier network for internet of vehicles. IEEE Internet Things J. 2021;8(19):14787–14796.
31. Zhou Z-H, Machine learning. Springer Nature. 2021.
32. Mitchell TM, et al. Machine learning. McGraw-hill New York. 2007;1.
33. El Naqa I, Murphy MJ. What is machine learning? Springer. 2015.
34. Bottou L. Stochastic gradient descent tricks. Neural Networks: Tricks of the Trade: Second Edition. 2012;421–436.
35. Kerr P. Adaptive learning. ELT J. 2016;70(1):88–93.
36. Gupta V, Jung C, Neel S, Roth A, Sharifi-Malvajerdi S, Waites C. Adaptive machine unlearning. Adv Neural Inf Process Sys. 2021;34:16319–16 330.
37. Ruder S. An overview of gradient descent optimization algorithms. arXiv preprint.http://arxiv.org/abs/1609.04747. 2016.
38. Melnikov Y. Influence functions and matrices. CRC Press. 1998;119.
39. Ketkar N, Ketkar N. Stochastic gradient descent. Deep learning with Python: a hands-on introduction. 2017;113–132.
40. Tahiliani A, Hassija V, Chamola V, Guizani M. Machine unlearning: its need and implementation strategies, in 2021 Thirteenth International Conference on Contemporary Computing (IC3–2021), ser. IC3 '21. New York, NY, USA: association for computing machinery. 2021;241–246. [Online]. Available: https://doi.org/10.1145/3474124.3474158.
41. Sekhari A, Acharya J, Kamath G, Suresh AT. Remember what you want to forget: algorithms for machine unlearning. Advances in Neural Information Processing Systems. 2021;34: 18075–18086.
42. Gill PE, Murray W, Wright MH. Practical optimization. SIAM. 2019.
43. Bourtoule L, Chandrasekaran V, Choquette-Choo CA, Jia H, Travers A, Zhang B, Lie D, Papernot N, Machine unlearning, in,. IEEE Symposium on Security and Privacy (SP). IEEE. 2021;2021:141–59.
44. Warnecke A, Pirch L, Wressnegger C, Rieck K. Machine unlearning of features and labels. arXiv preprint. http://arxiv.org/abs/2108.11577. 2021.
45. Welsch RE. Influence functions and regression diagnostics, in Modern data analysis. Elsevier. 1982;149–169.
46. Covert I, Lundberg S, Lee S-I. Feature removal is a unifying principle for model explanation methods. arXiv preprint. http://arxiv.org/abs/2011.03623. 2020.
47. Van Dyk DA, Meng X-L. The art of data augmentation. J Comput Graph Stat. 2001;10(1):1–50.
48. Parisi GI, Kemker R, Part JL, Kanan C, Wermter S. https://www.businessinsider.in/tech/news/it-ministry-orders-vpn-providers-to-store-user-data-for-fiveyears-tech-news/articleshow/91334830.cms. Accessed 23 Jun 2023.
49. Allison B, Guthrie D, Guthrie L. Another look at the data sparsity problem. InText, Speech and Dialogue: 9th International Conference, TSD 2006, Brno, Czech Republic, September 11-15, 2006. Proceedings 9. Springer. 2006;327–34.
50. Zhang Y, Yang Q. An overview of multi-task learning. Natl Sci Rev. 2018;5(1):30–43.
51. Laal M, Salamati P. Lifelong learning; why do we need it? Procedia Soc Behav Sci. 2012;31:399–403.
52. Liu B, Liu Q, Stone P. Continual learning and private unlearning. arXiv preprint. http://arxiv.org/abs/2203.12817. 2022.
53. Nguyen TT, Duong CT, Weidlich M, Yin H, Nguyen QVH. Retaining data from streams of social platforms with minimal regret, in Twenty-sixth International Joint Conference on Artificial Intelligence, no. CONF. 2017.
54. Huang H, Ma X, Erfani SM, Bailey J, Wang Y. Unlearnable examples: making personal data unexploitable. arXiv preprint. http://arxiv.org/abs/2101.04898. 2021.
55. Chundawat VS, Tarun AK, Mandal M, Kankanhalli M. Zero-shot machine unlearning. arXiv preprint. http://arxiv.org/abs/2201.05629. 2022.
56. Guo C, Goldstein T, Hannun A, Van Der Maaten L. Certified data removal from machine learning models. arXiv preprint. http://arxiv.org/abs/1911.03030. 2019.
57. Ginart A, Guan M, Valiant G, Zou JY. Making AI forget you: data deletion in machine learning. Adv Neural Inf Process Sys. 2019;32.
58. Brophy J, Lowd D. Machine unlearning for random forests, in International Conference on Machine Learning. PMLR. 2021;1092–1104.
59. Thudi A, Deza G, Chandrasekaran V, Papernot N, Unrolling sgd: understanding factors influencing machine unlearning, in,. IEEE 7th European Symposium on Security and Privacy (EuroS&P). IEEE. 2022;2022:303–19.
60. Neel S, Roth A, Sharifi-Malvajerdi S. Descent-to-delete: gradient-based methods for machine unlearning, in Algorithmic Learning Theory. PMLR. 2021;931–962.
61. Graves L, Nagisetty V, Ganesh V. Amnesiac machine learning, in Proceedings of the AAAI Conference on Artificial Intelligence. 2021;35(13):11516–11524.
62. Dwork C, Differential privacy: a survey of results, in International conference on theory and applications of models of computation. Springer. 2008;1–19.
63. Cao Y, Yang J, Towards making systems forget with machine unlearning, in,. IEEE Symposium on Security and Privacy. IEEE. 2015;2015:463–80.
64. Cauwenberghs G, Poggio T. Incremental and decremental support vector machine learning. Adv Neural Inf Process Sys. 2000;13.
65. Chen Y, Xiong J, Xu W, Zuo J. A novel online incremental and decremental learning algorithm based on variable support vector machine. Clust Comput. 2019;22(3):7435–45.
66. Chundawat VS, Tarun AK, Mandal M, Kankanhalli M. Can bad teaching induce forgetting? Unlearning in deep networks using an incompetent teacher. arXiv preprint. http://arxiv.org/abs/2205.08096. 2022.
67. Schelter S, Grafberger S, Dunning T, Hedgecut: maintaining randomised trees for low-latency machine unlearning, in Proceedings of the 2021 International Conference on Management of Data. 2021;1545–1557.
68. Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. Mach Learn. 2006;63(1):3–42.
69. Golatkar A, Achille A, Soatto S. Forgetting outside the box: scrubbing deep networks of information accessible from input-output observations, in European Conference on Computer Vision. Springer. 2020; 383–398.
70. 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.

71. Baumhauer T, Scho¨ttle P, Zeppelzauer M. Machine unlearning: linear filtration for logit-based classifiers. arXiv preprint. http://arxiv.org/abs/2002.02730. 2020.

72. Koch K, Soll M. No matter how you slice it: machine unlearning with sisa comes at the expense of minority classes, in 2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML). IEEE. 2023;622–637.

73. Mahmud MS, Huang JZ, Salloum S, Emara TZ. and K. Sadat- diynov, A survey of data partitioning and sampling methods to support big data analysis, Big Data Mining and Analytics. 2020;3(2):85–101.

74. Picard RR, Berk KN. Data splitting. The American Statisti- cian. 1990;44(2):140–7.

75. Feng SY, Gangal V, Wei J, Chandar S, Vosoughi S, Mitamura T, Hovy E. A survey of data augmentation approaches for nlp. arXiv preprint. http://arxiv.org/abs/2105.03075. 2021.

76. Ul Hassan M, Rehmani MH, Rehan M, Chen J. Differential privacy in cognitive radio networks: a comprehensive survey. Cognitive Computation. 2022;1–36.

77. Szo¨re´nyi B. Characterizing statistical query learning: simplified notions and proofs, in International Conference on Algorithmic Learning Theory. Springer. 2009;186–200.

78. Yang K. New lower bounds for statistical query learning. J Comput Syst Sci. 2005;70(4):485–509.

79. Zhou Y, Huang K, Cheng C, Wang X, Hussain A, Liu X. Fastadabelief: improving convergence rate for belief-based adaptive optimizers by exploiting strong convexity. IEEE Transactions on Neural Networks and Learning Systems. 2022.

80. Ralambondrainy H. A conceptual version of the k-means algorithm. Pattern Recogn Lett. 1995;16(11):1147–57.

81. Karasuyama M, Takeuchi I. Multiple incremental decremental learning of support vector machines. IEEE Trans Neural Networks. 2010;21(7):1048–59.

82. Joyce JM. Kullback-leibler divergence, in International encyclope- dia of statistical science. Springer. 2011;720–722.

83. Clark LA, Pregibon D. Tree-based models, in Statistical models in S. Routledge. 2017;377–419.

84. Myles AJ, Feudale RN, Liu Y, Woody NA, Brown SD. An introduction to decision tree modeling. Journal of Chemometrics: A Journal of the Chemometrics Society. 2004;18(6):275–85.

85. Spinelli I, Scardapane S, Hussain A, Uncini A. Biased edge dropout for enhancing fairness in graph representation learning. arXiv preprint. http://arxiv.org/abs/2104.14210. 2021.

86. Zhang Q, Zhong G, Dong J. A graph-based semi-supervised multi-label learning method based on label correlation consistency. Cogn Comput. 2021;13(6):1564–73.

87. Miikkulainen R, Liang J, Meyerson E, Rawal A, Fink D, Fran- con O, Raju B, Shahrzad H, Navruzyan A, Duffy N, et al. Evolving deep neural networks, in Artificial intelligence in the age of neural networks and brain computing. Elsevier. 2019;293–312.

88. Agostinelli F, Hoffman M, Sadowski P, Baldi P. Learning activation functions to improve deep neural networks. arXiv preprint. http://arxiv.org/abs/1412.6830. 2014.

89. Chhikara P, Tekchandani R, Kumar N, Chamola V, Guizani M. Dcnn-ga: a deep neural net architecture for navigation of uav in indoor environment. IEEE Internet Things J. 2020;8(6):4448–60.

90. Mahmud M, Kaiser MS, Hussain A, Vassanelli S. Applications of deep learning and reinforcement learning to biological data. IEEE transactions on neural networks and learning systems. 2018;29(6):2063–79.

91. Boyd SP, Vandenberghe L. Convex optimization. Cambridge university press. 2004.

92. Gao B, Pavel L. On the properties of the softmax function with application in game theory and reinforcement learning. arXiv preprint. http://arxiv.org/abs/1704.00805. 2017.

93. Freese F, et al. Testing accuracy. Forest Sci. 1960;6(2):139–45.

94. Hagenbach J, Koessler F. The Streisand effect: signaling and partial sophistication. J Econ Behav Organ. 2017;143:1–8.

95. Swiler LP, Paez TL, Mayes RL. Epistemic uncertainty quantification tutorial, in Proceedings of the 27th International Modal Analysis Conference. 2009.

96. Carlini N, Chien S, Nasr M, Song S, Terzis A, Trame`r F, Membership inference attacks from first principles, CoRR, vol. abs/2112.03570, 2021. [Online]. Available: https://arxiv.org/abs/2112.03570.

97. Shokri R, Stronati M, Shmatikov V. Membership inference attacks against machine learning models, CoRR, vol. abs/1610.05820, 2016. [Online]. Available: http://arxiv.org/abs/1610.05820.

98. Shuvo MSR, Alhadidi D. Membership inference attacks: analysis and mitigation, in 2020 IEEE 19th International Conference on Trust. Security and Privacy in Computing and Communications (TrustCom). 2020;1410–1419.

99. Liu X, Xie L, Wang Y, Zou J, Xiong J, Ying Z, Vasilakos AV. Privacy and security issues in deep learning: a survey. IEEE Access. 2020;9:4566–93.

100. Chundawat VS, Tarun AK, Mandal M, Kankanhalli M. Zeroshot machine unlearning. IEEE Transactions on Information Forensics and Security. 2023.

101. Wang K, Fu Y, Li K, Khisti A, Zemel RS, Makhzani A. Variational model inversion attacks, *CoRR*, vol. abs/2201.10787, 2022. [Online]. Available: https://arxiv.org/abs/2201.10787.

102. 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, ser. CCS '15. New York, NY, USA: Association for Computing Machinery. 2015;1322–1333. [Online]. Available: https://doi.org/10.1145/2810103.2813677.

103. Xian Y, Lampert CH, Schiele B, Akata Z. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. IEEE Trans Pattern Anal Mach Intell. 2018;41(9):2251–65.

104. 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.

105. Tarun AK, Chundawat VS, Mandal M, Kankanhalli M. Fast yet effective machine unlearning. arXiv preprint. http://arxiv.org/abs/2111.08947. 2021.

106. Becker A, Liebig T. Evaluating machine unlearning via epistemic uncertainty. arXiv preprint. http://arxiv.org/abs/2208.10836. 2022.

107. Wiedmann T, Minx J. A definition of 'carbon footprint.' Ecological economics research trends. 2008;1(2008):1–11.

108. Henderson P, Hu J, Romoff J, Brunskill E, Jurafsky D, Pineau J. Towards the systematic reporting of the energy and carbon footprints of machine learning. J Mach Learn Res. 2020;21(1):10039–10081.

109. L. F. W. Anthony, B. Kanding, and R. Selvan, Carbontracker: tracking and predicting the carbon footprint of training deep learning models. arXiv preprint. http://arXiv:2007.03051. 2020.

110. T. Alladi, B. Gera, A. Agrawal, V. Chamola, and F. R. Yu, Deepadv: a deep neural network framework for anomaly detection in vanets. IEEE Transactions on Vehicular Technology. 2021;70(11):12013–12023.

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

112. Zhang C, Xie Y, Bai H, Yu B, Li W, Gao Y. A survey on federated learning. Knowl-Based Syst. 2021;216: 106775.

113. Li L, Fan Y, Tse M, Lin K-Y. A review of applications in federated learning. Comput Ind Eng. 2020;149: 106854.

114. Li T, Sahu AK, Talwalkar A, Smith V. Federated learning: challenges, methods, and future directions. IEEE Signal Process Mag. 2020;37(3):50–60.
115. Aspin DN, Chapman JD. Lifelong learning: concepts and conceptions. Int J Lifelong Educ. 2000;19(1):2–19.
116. Thrun S. Lifelong learning algorithms Learning to learn. 1998;8:181–209.
117. L.a. is suing ibm for illegally gathering and selling user data through its weather channel app - Los Angeles Times. https://www.latimes.com/business/technology/la-fi-tn-city-attorney-weather-app-20190104-story.html. Accessed 27 Jun 2023.
118. Yapo A, Weiss J. Ethical implications of bias in machine learning, 2018.
119. Roscher R, Bohn B, Duarte MF, Garcke J. Explainable machine learning for scientific insights and discoveries. Ieee Access. 2020;8:42200–42216.
120. Belle V, Papantonis I. Principles and practice of explainable machine learning. Frontiers in big Data. 2021;39.
121. Lundberg SM, Lee S-I. A unified approach to interpreting model predictions, in Advances in Neural Information Processing Systems 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017;4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.
122. Ribeiro MT, Singh S, Guestrin C. Why should I trust you?: explaining the predictions of any classifier, in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August. 2016;1135–1144.
123. Strubell E, Ganesh A, McCallum A. Energy and policy considerations for deep learning in nlp. arXiv preprint. http://arxiv.org/abs/1906.02243. 2019.
124. Schwartz R, Dodge J, Smith NA, Etzioni O. Green AI. Commun ACM. 2020;63(12):54–63.
125. Patterson D, Gonzalez J, Le Q, Liang C, Munguia L-M, Rothchild D, So D, Texier M, Dean J. Carbon emissions and large neural network training. arXiv preprint. http://arxiv.org/abs/2104.10350, 2021.
126. Adiwardana D, Luong M-T, So DR, Hall J, Fiedel N, Thop- pilan R, Yang Z, Kulshreshtha A, Nemade G, Lu Y, et al. Towards a human-like open-domain chatbot. arXiv preprint. http://arxiv.org/abs/2001.09977. 2020.
127. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, et al. Language models are few-shot learners. Adv Neural Inf Process Syst. 2020;33:1877–901.
128. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res. 2020;21(140):1–67.
129. Fedus W, Zoph B, Shazeer N. Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. 2021.
130. Lepikhin D, Lee H, Xu Y, Chen D, Firat O, Huang Y, Krikun M, Shazeer N, Chen Z. Gshard: Scaling giant models with conditional computation and automatic sharding. arXiv preprint. http://arxiv.org/abs/2006.16668. 2020.
131. Apte C, The role of machine learning in business optimization, in Proceedings of the 27th International Conference on Machine Learning (ICML-10). Citeseer. 2010;1–2.
132. Singh S, Sulthana R, Shewale T, Chamola V, Benslimane A, Sikdar B. Machine-learning-assisted security and privacy provisioning for edge computing: a survey. IEEE Internet Things J. 2021;9(1):236–60.
133. Miao Y, Chen C, Pan L, Han Q-L, Zhang J, Xiang Y. Machine learning–based cyber attacks targeting on controlled information: a survey. ACM Computing Surveys (CSUR). 2021;54(7):1–36.
134. Wazid M, Das AK, Chamola V, Park Y. Uniting cyber security and machine learning: advantages, challenges and future research. ICT Express. 2022;8(3):313–21.
135. Chamola V, Goyal A, Sharma P, Hassija V, Binh HTT, Saxena V. Artificial intelligence-assisted blockchain-based framework for smart and secure emr management. Neural Computing and Appli- cations. 2022;1–11.
136. Isinkaye FO, Folajimi YO, Ojokoh BA. Recommendation systems: principles, methods and evaluation. Egypt Inform J. 2015;16(3):261–73.
137. Pavithra D. Jayanthi A. A study on machine learning algorithm in medical diagnosis. Int J Adv Res Comput Sci. 2018;9(4).
138. Rohmetra H, Raghunath N, Narang P, Chamola V, Guizani M, Lakkaniga NR. AI-enabled remote monitoring of vital signs for covid-19: methods, prospects and challenges. Computing. 2021;1–27.
139. Bansal G, Chamola V, Narang P, Kumar S, Raman S. Deep3dscan: deep residual network and morphological descriptor based framework forlung cancer classification and 3d segmentation. IET Image Proc. 2020;14(7):1240–7.
140. Delgado-Rodriguez M, Llorca Bias J. Journal of Epidemiology & Community Health. 2004;58(8):635–641.
141. Danks D, London AJ. Algorithmic bias in autonomous systems. Ijcai. 2017;17(2017):4691–7.
142. Malerba D, Esposito F, Lanza A, Lisi FA. Machine learning for information extraction from topographic maps. Geographic data mining and knowledge discovery. 2001;291–314.
143. Hutchins WJ, Machine translation: past, present, future. Ellis Horwood Chichester. 1986.
144. Grieco LA, Rizzo A, Colucci S, Sicari S, Piro G, Di Paola D, Boggia G. IoT-aided robotics applications: technological implications, target domains and open issues. Comput Commun. 2014;54:32–47.
145. Roy Chowdhury A, Iot and robotics: a synergy. PeerJ Preprints. 2017;5:e2760v1.
146. Zhao W, Chellappa R, Phillips PJ, Rosenfeld A. Face recognition: a literature survey. ACM computing surveys (CSUR). 2003;35(4):399–458.
147. Hernandez G, Arias O, Buentello D, Jin Y, Smart nest thermostat: a smart spy in your home. Black Hat USA. no. 2015, 2014.
148. Gogate M, Dashtipour K, Hussain A, Towards robust real-time audio-visual speech enhancement. arXiv preprint. http://arxiv.org/abs/2112.09060. 2021.
149. Gogate M, Dashtipour K, Adeel A, Hussain A. Cochleanet: a robust language-independent audio-visual model for real-time speech enhancement. Information Fusion. 2020;63:273–85.
150. Hussain T, M. Gogate, K. Dashtipour, and A. Hussain, Towards intelligibility-oriented audio-visual speech enhancement. arXiv preprint. http://arxiv.org/abs/2111.09642. 2021.
151. Adeel A, Gogate M, Hussain A. Contextual deep learning- based audio-visual switching for speech enhancement in real-world environments. Information Fusion. 2020;59:163–70.
152. Alladi T, Kohli V, Chamola V, Yu FR, Securing the internet of vehicles: a deep learning based classification framework. IEEE Networking Letters. 2021.
153. Wang W, Zheng VW, Yu H, Miao C. A survey of zero-shot learning: settings, methods, and applications. ACM Transactions on Intelligent Systems and Technology (TIST). 2019;10(2):1–37.
154. Wang Y, Yao Q, Kwok JT, Ni LM. Generalizing from a few examples: a survey on few-shot learning. ACM computing surveys (csur). 2020;53(3):1–34.

155. Rahman S, Khan S, Porikli F. A unified approach for conventional zero-shot, generalized zero-shot, and few-shot learning. IEEE Trans Image Process. 2018;27(11):5652–67.

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