An Empirical Evaluation of Machine Unlearning

Shilpa Roy
University of Maryland, College Park

Clifford Bakalian
University of Maryland, College Park

Abstract

Machine unlearning is a novel field in the research surrounding data privacy. Namely, compilation with the GDPR's "right to be forgotten" also implicates machine learning models that learned from an individual's data. This has resulted in the need for dedicated machine unlearning. The diversity of machine learning models and use cases has predictably resulted in a wide array of potential solutions to the problem, each with their own benefits and drawbacks. This paper directly compares two of the most competitive such solutions: Guo et al.'s ϵ-certified data removal [1], and Bourtoule et al.'s SISA training [2]. Our initial experimental results demonstrate that SISA is marginally better in terms of both accuracy and efficiency in comparison to certified data removal.

1 Introduction

Recent GDPR regulations and the "right to be forgotten" have mandated that data collectors find efficient ways to delete data on request. For most data, this amounts to scraping the DBMS clean of traces of the raw data. The natural extension for compliance in the case of machine learning models, then, is to delete the data that the model was trained on.

Unfortunately, this is a necessary, but not sufficient condition for compliance with the GDPR. During training, a single data point can permanently influence the predictive power of the model, because the decisions the model makes is directly influenced by the data. Machine learning models are intrinsically "lossy, compressed version[s] of data" [3]. Indeed, efficient attacks, such as membership inference and model inversion, are able to leverage auxiliary data against the model to uncover information even if the training data is deleted.

The broader implication of these attacks is the need for dedicated **machine unlearning**, by which we attempt to remove the training data from the model itself. The naive method for doing so is to remove the data point on request for deletion and retrain the model on the remaining training data. As this is an overly costly operation, the fundamental problem faced by researchers in this space is to identify efficient and effective methods for achieving machine unlearning.

Inspired by the theoretical guarantees of differential privacy, a natural problem in this space, then, is to remove data from a machine learning model or computationally bound the influence of that data, in such a way that an adversary cannot guess what was deleted. A crude analogy presented by Hume et al is for a human to forget a single shade of blue, but still understand what blue is, but not know which exact shade was for-

gotten [4].
A naive, yet powerful, attempt in this direction is to retrain the model from scratch. That is, on request to delete a data point, discard the data point and the existing model, and create a new model based on the $n-1$ data points that were not deleted. For deep networks, which see millions of data points, doing this even once is a costly operation, let alone being required to do so at regular intervals, in the worst case. Therefore, the literature in the space of machine unlearning universally requires the following:

1. Remove a data point from a model

2. Do so efficiently

The novelty of the problem space means there is still much disagreement on what constitutes a 'good' solution to machine unlearning.
Schelter [3] proposes a strong theoretical formalism:

$$t_{forget}(t_{learn}(D, \theta), d_m, \theta) = t_{learn}(D \backslash d_m, \theta)$$

In layman's terms, any process of unlearning (t_{forget}) ought to be exactly equivalent to a model that never saw the data to begin with. This definition is not universally accepted; Baumhauer et al. [5] demonstrate that stringent guarantees are near impossible for deep neural networks. They argue instead for a probabilistic definition: namely, that a model after unlearning only need belong to the same distribution of models in the hypothesis space as a model that never saw the data. Still others provide no definitions at all; an idea is proposed and empirical evidence on sample architectures is provided as evidence of its efficacy.

Often, these evaluations are conducted only in comparison to naive retraining, which is universally agreed upon as an impractical solution. Since improvement over naive retraining is easily demonstrated, it is difficult to know the true feasibility of many unlearning algorithms currently being employed. Our contribution to this space is to conduct an empirical evaluation of some of the most competitive unlearning algorithms. We now offer an exploration of the methods that have been developed thus far to contextualize our experimental design.

2 Related Works

Several clear classes have emerged in machine unlearning in terms of approach and evaluation. Decremental learning [3, 6, 7] offers exact deletion guarantees for specific types of models. General model updates [8, 1, 9] leverage differential privacy for more epochs of training to solve the problem. Others [5, 10] offer approximate data deletion that is probabilistic in nature. Obfuscating the model by altering the labels of the training data [11] or splitting up the work by binning [12, 2] are also popular solutions.

2.1 Decremental Learning

The first class of solutions, called *decremental learning*, is a form of machine unlearning that is exact: they follow the strong definition proposed by Schelter. Schelter provides algorithms for decrementally learning three types of ML models, each generally involving incurring a space overhead to keep intermediate forms of the model. On request to unlearn a specific data point, we successively update the intermediates, thereby effectively retraining from scratch. Ginart's [7] algorithm for k-means clustering is similar; it stores metadata within each data point during training for performance gains when unlearning. Finally, DART [6], presented by Brophy and Lowd, similarly provides strong deletion guarantees for decision trees. It stores sensitive data at the leaves of the tree, and caches data statistics at each node. By choosing the split variable at a given node randomly, the authors show that unlearning can be accomplished by retraining only certain subtrees, cutting the runtime significantly.

The central caveat here is the lack of *generalizability* these solutions offer. DART only works specifically for decision trees - extending the idea for general graph-based data was shown to be insecure [13]. Schelter's algorithms work only on non-iterative algorithms in order to maintain the intermediates. This means any algorithm that requires gradient descent (practically all realistic models) cannot use decremental learning - they are simply not deterministic enough. In summation, despite the strong theoretical guarantees, the central flaw of this type of learning is a lack of **model agnosticity**, i.e. a catch-all method that works for all neural networks, regardless of the type of data or architecture.

2.2 Model Updates

Another class of solutions follows the same general philosophy of decremental learning in attempting to modify the existing model in some way to achieve machine unlearning. Early on, Du et al [14] suggested, completely unrelated to privacy concerns, but for the sake of removing unsavory data, the idea of using controlled gradient ascent in order to reverse the gradient descent by which most models are trained. More significantly, they were the first to suggest an idea that would become the precursor to training use a one-step Newton.

The Newton method, proposed by Guo et al. [1] (in the context of machine unlearning), involves modifying the loss function to remove the offending data point and then continuing to train, with the hope that a new model minimized on a separate loss function will mask the data that needs to be forgotten. This philosophy of unlearning is echoed in Golatkar et al. [9], where the weights of the models are directly modified in order to "scrub" them clean of the influence of a data point. Both of these papers are also significant for their motivations. Heavily inspired by the guarantees of differential privacy, Guo et al. offers a novel definition for the problem space dubbed ϵ-certified data removal:

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

In this framework, unlearning is said to be effective if we can bound the difference between a model whose data is removed (the numerator) and a model that never saw the data to begin with. As with differential privacy and the need to add noise to provide strong guarantees, Guo et al. offer a solution that adds noise to mask the deletion of the data, sacrificing the accuracy of the final model.

To combat this, DeltaGrad [8], another mechanism that borrows from the ideas of this class of solutions, allows for rapid retraining for data cached during the training phase via quasi-Newton methods. In addition to empirically demonstrating substantial speedups when training on generalized machine learning models, this paper is insightful for contradicting the influence of differential privacy in the space of machine learning. It argues that differential privacy can only bound the influence of a certain point, whereas machine unlearning requires ϵ to be 0 in order to remove any possibility of data recovery. Its methods offer a form of unlearning without the need to add probabilistic noise.

2.3 Approximate Deletion

While having a network intentionally forget an entire data point (or subset of the data) is usually used for the goal of removal, having a network forget certain attributes or parts of the data can additionally be used to combat privacy concerns or to preserve anonymity. These next approaches to deletion in machine learning keep this in mind as they go for a more approximate deletion strategy rather than completely deleting the data in question.

Baumhauer et al. [5], stress that a probabilistic, approximate method of data deletion is the only practical way to implement machine unlearning for deep networks whose connections are harder to interpret. To this end, they reframe the problem in terms of the distributions over the models. In fact, they go a step further by defining weak unlearning in terms of the ability to distinguish between the distribution of the outputs themselves. To do so, they propose training a meta machine learning model that attempts to distinguish between a model that is naively retrained and one that is untrained; unlearning is achieved when we are able to fool such a classifier with high probability. To this end, they propose linear filtration, which amounts to shifting outcomes away from the sensitive data, as an efficient form of unlearning for classifiers and deep neural networks.

Recently Izzo et al. [10] proposed an approximate deletion model which was linear in the feature dimension and is also independent of the training data. This could only be done if the restrictions for removing data from the network was relaxed, similar to the definition found in [5]. Like Baumhauer, Izzo et al. reframe the problem specification in terms of resistance to attacks by proposing a new test called the feature injection test (FIT), which checks how much information a model retains about some sensitive attribute. Under this framework, Izzo et al. define a successful unlearning as

$$\theta^{approx}[d+1]/\theta^{full}[d+1] \approx 0$$

That is, assuming the sensitive feature we wish to delete is, without loss of generality, a last, inserted dimension in a d-dimensional space, and is encoded with 1 (for presence), we would like our approximate form of data deletion to remove all traces of the attribute, and approach 0.

To this end, in contrast to other solutions such as the use of influence functions and one-step Newton updates, they present the projective residual update (PRU), which computes parameter update vectors onto a lower dimensional subspace in order to get runtime proportional to the dimensionality and is optimal in terms of deletion accuracy.

2.4 Obfuscation

All of the previous approaches deal with deletion of data from a model and then trying to get a network adjusted to this loss of data. One approach that deals with unlearning in a more literal sense attempts to use new data to cancel out old data, an idea novel enough to warrant its own class despite only one major paper by Graves et al. [11] being written on it.

To provide more detail, Graves et al. provide two methods of unlearning. The first, simply dubbed unlearning, involves replacing data with incorrect copies and retraining for a couple of iterations in order to ”confuse” the model from what it has learned from the sensitive data we wish to delete. Despite the lack of theoretical guarantees of security, the authors do provide compelling empirical evidence for privacy by showing that the resulting models are resistant to some of the attacks the researchers devised. They paper also provides amnesiac learning, which involves a space overhead to keep track of which batches of data contain the points to be deleted and then retraining only for those batches with a backtracking approach. This particular strategy of splitting up work is also employed by two of the most important papers in the field, explained further in the following section.

2.5 Splitting

Splitting focuses on reducing the computational overhead of retraining the network through means of splitting up the work. One of, if not the absolute, earliest machine unlearning approach, by Cao et al., was to convert the learning algorithms to a set of summations, where each summation is some transformation on sets of the training data. Many machine learning algorithms can be formalized in this way thanks to guarantees from statistical query model algorithms [15]. Cao et al. use this structure for non-negligible speed ups. The problem with operating within this framework is that it is not model agnostic; statistical query based algorithms are not general and may diverge in unbounded ways for deep neural networks. Though it is neither model agnostic nor theoretically powerful (like the exact solutions offered by decremental learning), it is an important precursor for solutions in this space.

Splitting is also the approach adopted by one of the most famous and well-cited papers in this space: Bourtoule et al.'s seminal SISA paper [2]. In SISA, the data is split into distinct **shards** and trained in **isolation** to create many separate models. During training, these distinct models contain **slices**, which are fragments of state-containing information that allow us to backtrack to a previous point in the training on request of deletion (which aligns naturally with methods proposed in Class 2). Finally, to ensure all data gets a say in the final result, the model is used by **aggregating** the posterior of all shard models (usually by a majority vote, or average). The notoriety of this paper deals not with a strong theoretical proof of security like other papers, but the introduction of a set of six guidelines (denoted G1-G6) for machine unlearning that challenge most existing solutions: the SISA paper mandates that any unlearning solution needs to be easily implemented (G1), must not (overly) sacrifice the accuracy of the model (G2), must be computationally efficient (G3) must provide provable guarantees of data removal (G4), must be model agnostic (G5), and must induce limited overhead to models that are already resource intensive (G6). It is relatively easy to see the impact this paper has had on the space as a whole by directly invalidating entire classes of solutions. For instance:

- decremental learning is not model agnostic
- update methods either 1) incur too much overhead or 2) are inaccurate due to noise added by differential privacy
- approximate methods and obfuscation are too probabilistic to prove the removal of data adequately
- amnesiac learning incurs too much overhead

This is not to say the SISA paper is without criticism. Splitting up the data reduces the expressive power of the resulting model in a way that the authors of the paper argue is negligible.

In reality, this may not be so trivial for companies whose bottom line relies on the precision that a model trained on all the data would provide. Furthermore, the SISA paper is a guideline for how to train a GDPR-compliant model, not how to be GDPR-compliant for a model that *already exists*, and has existed for long enough that starting from scratch is unfeasible.

2.6 Context for Our Evaluation

Unlearning is a very interesting concept theoretically, but for our purposes, we offer a definition of "competitive" in terms of its feasibility. Since unlearning grew as a field primarily to address GDPR compliance for proprietary ML models, practical adoption by companies acts as our metric.

Naturally, such a definition eliminates certain classes of solutions. Exact methods of data deletion are not competitive by our definition because they are not model agnostic, and therefore unlikely to be universally adopted across the board. The same is true for obfuscation. For high dimensional data, obfuscation pollutes the original model's ability to learn characteristics about the feature space, which is unlikely to be popular for proprietary models that have been tuned to near perfection.

For this experiment, we provide an evaluation of Guo et al.'s certified data removal [1] and Bourtoule et al.'s SISA training framework [2]. Despite criticisms detailed above, both methods are model agnostic and likely to be adopted. Certified data removal's close tie with the strong guarantees of differential privacy make it a popular choice, and the SISA training framework is so pivotal to the unlearning space to the point of being considered a quintessential unlearning algorithm for which to design attacks against [16].

3 Methodology

This section discusses the setup and configurations used to run our evaluations for both Guo et. al.'s certified data removal [1] and Bourtolue et. al.'s SISA training framework [2].

3.1 Dataset

Our goal with this experiment is to evaluate industry-level unlearning techniques. In industry, one of the most common machine learning tasks is visual image processing. To this end, we use the Street View House Numbers (SVHN) dataset [17]. This is similar in flavor to the seminal MNIST dataset [18, 19], but it not only contains 600,000 labelled images (in contrast with MNIST's 70,000), but also solves a more computationally difficult real-world problem of recognizing digits in natural images (once again, in contrast to MNIST's recognizing hand-written digits).

The high volume of the SVHN dataset and the central problem it solves makes it an ideal candidate for the type of data that industry models are designed to classify.

3.2 Evaluation Metric

Furthermore, we also define an evaluation metric by which to judge the unlearning methods. Recalling that the central goal of machine unlearning is to beat naive retraining, time is an obvious metric to use. Another metric that we considered was whether the data was "sufficiently" deleted. Common ways to do this include training a separate neural network that attempts to distinguish between elements that are in the training set, or similar to Chen et al. [16], design attacks against models that attempt to violate privacy. Ultimately, we decided not to employ this metric. Since most unlearning methods will likely have some form of being classified as "good enough" via GDPR regulations, ensuring complete deletion is unlikely to be a goal of most industry models. Instead, companies are likely to be more concerned with ensuring that their models do not suffer in terms of accuracy as a result of the unlearning process, since this directly affects their bottom line. As a result, we intend to directly measure the accuracy and efficiency of the unlearned model for our evaluation.

3.3 Certified Data Removal

We started with Guo et. al.'s approach [1], which focused on a model update approach using a Newton Step. The goal of this approach was to provide a quick retraining model, which would also be indistinguishable from a model which never saw the data to begin with.

This approach uses one convolutional layer with 2 hidden layers to train the data and utilizes a trick by Goodfellow [20] to obtain efficient gradient computation.

We started off by training using the recommendations made by the authors of the paper by using $\epsilon = 0.1$ and $\delta = 0.00001$. After training, we used Guo et. al.'s implementation of data removal to remove various amounts of training points via the Newton removal mechanism [1]. We tested removing 1, 100 and and 1000 training points and tested both the accuracy of the initial model, the updated model, and the time it took to update the model. Each of these points was chosen at random. These results are marked in section 4 and compared against the SISA model in section 3.4 and the naive unlearning approach in section 3.5.

3.4 SISA

The second model we used in our experiment is the SISA model discussed in Bourtoule et. al.'s paper on a sharding approach [2]. This model is very straightforward: split up the modeling into shards, further split up each shard into slices, and train the shards separately. The novelty here being that when data is deleted, only certain shards need to be retrained. Further, the slices act as checkpoints in the retraining process, making retraining for a single shard doubly more efficient.

Using recommendations by the paper, we decided to use 5 shards for our experiment. Like our approach for testing ϵ-certified data removal, this network architecture also uses a single convolutional layer followed by multiple hidden layers. Specifically, we employed the wide ResNet-1-1 (shown in Figure 1) architecture described in [21] as this was the implementation used in Bourtoule et. al.'s paper [2]. From here, we followed the same methodology as described above by training the dataset, and randomly querying 1, 100, and 1000 data points to be removed, all while keeping track of the time taken to unlearn and the accuracy of the model afterwards.

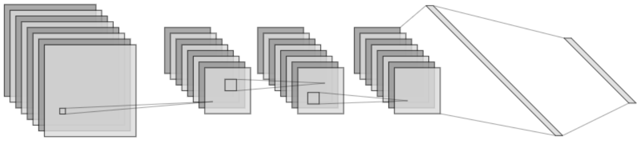

Figure 1: A sample architecture (not to scale), featuring a convolution layer followed by three hidden layers and output

3.5 Naive Unlearning

Since both approaches take fundamentally different approaches to the training and unlearning process, we also decided to test naive retraining using the same architectures and methods.

For Guo et al.'s ϵ-certified data removal in section 3.3, the process was straightforward. To simulate naive retraining on the same network, we simply queried 1, 100, and 1000 data points to be removed, extracted them from the data, and repeated the initial retraining process.

For the SISA model in section 3.4, we simply trained using only one shard in order to simulate training on the entire model. From there, we simply retrained and recorded the accuracy and time.

4 Results

A summary of our experimental results is provided in Figure 2. As expected, both approaches consistently outperformed their naive counterparts in terms of efficiency. Retraining from scratch for such a high volume dataset is simply too expensive. Despite this, the results demonstrate the as the number of datapoints to delete increases, ϵ-certified removal seems to take longer to unlearn. In contrast, the time which SISA naively retrained was about the same per shard regardless of how much data was removed. However, the number of shards retrained would vary depending on how much data was removed, thus, the naive approach was also significantly slower at an overall time which the machine was training.

The accuracy of the models also followed a downward trend the more points which had to be removed, even for naive retraining. Since the initial training methods for the two approaches was different (by consequence of the sharding approach employed by SISA), it is difficult to compare the degradation in accuracy between the two post-unlearning. By magnitude alone, certified removal seems to take a greater hit in accuracy, but since the initial accuracy of the models after training is different, it's difficult to know the precise impact unlearning had on the accuracy. Regardless, there does not seem to be too much accuracy degradation for either model.

5 Limitations and Future Work

One of the largest limitations of this project was resources. We ran our models on a desktop computer with an NVIDIA 2060 with CUDA 11.2 framework through a docker container. Our host system was Arch Linux with a docker container running Ubuntu 20.04. A memory constraint of six gigabytes made running the models very computationally expensive. Should we decide to revist this project in the future, a dedicated instance on Amazon Web Services or Google Collab would probably be used.

We were also limited by time, which heavily limited the number of unlearning algorithms we were able to evaluate, the number of datasets to base this evaluation on, and time we could have dedicated to optimizing our results. With more time, we could potentially use other datasets like MNIST or CIFAR, and implement more approaches like obfuscation or approximate deletion.

Based on the evaluation metric we used and our experimental design, it appears that SISA training is slightly superior to certified removal in terms of accuracy and efficiency.

	Certified Removal	Certified Removal-Naive	SISA	SISA-Naive
Retraining Time	4 sec.	53 min.	12 min per shard	67 min.
Original Accuracy	86%	88%	97%	96%
Unlearned Accuracy	86%	88%	97%	96%

(a) results for removing 1 datapoint

	Certified Removal	Certified Removal-Naive	SISA	SISA-Naive
Retraining Time	10 min.	57 min.	10 min per shard	58 min.
Original Accuracy	86%	88%	97%	96%
Unlearned Accuracy	84%	87%	96%	96%

(b) results for removing 100 datapoints

	Certified Removal	Certified Removal-Naive	SISA	SISA-Naive
Retraining Time	47 Min.	55 min.	11 min per shard	62 min.
Original Accuracy	86%	88%	97%	96%
Unlearned Accuracy	83%	85%	95%	95%

(c) results for removing 1000 datapoints

Figure 2: results for time and accuracy

However, we hesitate to recommend one over the other absolutely - it is possible that with a different number of shards or differing values of ϵ and δ, the results may be markedly different.

For the future, an interesting project we consider is leveraging Optuna, a hyperparameter optimization framework [22], which recursively searches for hyperparameters to maximize accuracy. By searching for the ideal values of ϵ, δ, and the number of shards (for SISA), it would be possible to give both unlearning method their "best possible shot", which would make it easier to reach a definitive conclusion about the efficacy of one over the other. It would also be possible to optimize for network architecture. The architecture for certified removal involved 2 hidden layers, while the architecture for SISA involved 3 hidden layers. Further optimizing for this architecture may improve the quality of the experiment in the future.

6 Closing Remarks

The ubiquity of machine learning in the modern era makes the topic of machine unlearning very important and intrinsically related to the goals of maintaining privacy and data confidentiality.

Though the space certainly does not suffer from a lack of solutions, it is still remarkably young. Not only do researchers themselves disagree about problem specifications and design, but even if an optimal solution were found, machine learning in industry naturally lends itself to trying to shave off retraining time, and providing even stronger guarantees.

Our contributions to this space in offering preliminary calculations of two separate unlearning methods with different goals we also consider to be a step towards larger comparison studies using some of the frameworks we described in our Future Work section.

References

[1] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. Certified data removal from machine learning models, 2020.

[2] Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning, 2020.

[3] Sebastian Schelter. “amnesia” – towards machine learning models that can forget user data very fast, Aug 2019.

[4] David Hume. *An enquiry concerning human understanding: A critical edition*, volume 3. Oxford University Press, 2000.

[5] Thomas Baumhauer, Pascal Schöttle, and Matthias Zeppelzauer. Machine unlearning: Linear filtration for logit-based classifiers, 2020.

[6] Jonathan Brophy and Daniel Lowd. Dart: Data addition and removal trees, 2020.

[7] Antonio Ginart, Melody Y Guan, Gregory Valiant, and James Zou. Making ai forget you: Data deletion in machine learning. *arXiv preprint arXiv:1907.05012*, 2019.

[8] Yinjun Wu, Edgar Dobriban, and Susan B. Davidson. Deltagrad: Rapid retraining of machine learning models, 2020.

[9] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9304–9312, 2020.

[10] Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models, 2021.

[11] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning, 2020.

[12] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pages 463–480. IEEE, 2015.

[13] Michael Ellers, Michael Cochez, Tobias Schumacher, Markus Strohmaier, and Florian Lemmerich. Privacy attacks on network embeddings. *arXiv preprint arXiv:1912.10979*, 2019.

[14] Min Du, Zhi Chen, Chang Liu, Rajvardhan Oak, and Dawn Song. Lifelong anomaly detection through unlearning. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS ’19, page 1283–1297, New York, NY, USA, 2019. Association for Computing Machinery.

[15] Cheng Chu, Sang Kyun Kim, Yian Lin, YuanYuan Yu, Gary Bradski, Andrew Y Ng, and Kunle Olukotun. Map-reduce for machine learning on multicore. *Advances in neural information processing systems*, 19:281, 2007.

[16] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. When machine unlearning jeopardizes privacy, 2020.

[17] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[18] Xudie Ren, Haonan Guo, Shenghong Li, Shilin Wang, and Jianhua Li. A novel image classification method with cnn-xgboost model. In Christian Kraetzer, Yun-Qing Shi, Jana Dittmann, and Hyoung Joong Kim, editors, *Digital Forensics and Watermarking*, pages 378–390, Cham, 2017. Springer International Publishing.

[19] Xudie Ren, Haonan Guo, Shenghong Li, Shilin Wang, and Jianhua Li. A novel image classification method with cnn-xgboost model. In Christian Kraetzer, Yun-Qing Shi, Jana Dittmann, and Hyoung Joong Kim, editors, *Digital Forensics and Watermarking*, pages 378–390, Cham, 2017. Springer International Publishing.

[20] Ian Goodfellow. Efficient per-example gradient computations. *arXiv preprint arXiv:1510.01799*, 2015.

[21] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[22] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '19, page 2623–2631, New York, NY, USA, 2019. Association for Computing Machinery.