THE UNIVERSITY OF HONG KONG

DEPARTMENT OF
COMPUTER SCIENCE

A Repository of Jupyter Notebooks on Unlearning in Federated Learning

Final Year Project
Interim Report

Sheng "Victor" HUANG

Supervised by **Prof. S.M. YIU**

HONG KONG
Sun 22nd Jan 2023

Abstract

Machine Learning (ML) systems may contain a great amount of private data, just like other types of computer systems, and the abundance of data has paved the road for rapid development of ML technologies in recent years. Yet, erasing data from ML systems is orders of magnitude harder than from an ordinary database system. Recent uptake of interest by technology consumers, providers and governments has created such a demand. Federated Learning (FL) is also a technique to address some of the privacy and security issues in ML. Since they target different aspects of privacy and security, FL systems still have problems that could be solved by machine unlearning. As there lacks such work, this project aims to provide materials unifying the field of machine unlearning in federated learning, using a repository of Jupyter Notebooks. Half of the repository has been done, including with the background research and about half of the related papers, meeting the educational and research goals but more work needs to be done next semester.

Keywords

machine unlearning, federated learning, right to be forgotten, privacy, security, data deletion, memory loss, amnesia

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

This project focuses on the cross-section between the areas of machine unlearning and Federated Learning (FL), namely how to apply machine unlearning in FL. Both machine unlearning and FL are techniques to enhance privacy and security in Machine Learning (ML) that are subject to intense research and rapid development.

1.1 Machine Unlearning

ML sometimes involves using data of personal and sensitve nature, such as medical [1], biometric [2] and geolocation information [3]. In some cases, we even need to constantly bring in new data to update ML models using various methods such as incremental learning, online learning, and data stream learning [4]. On the other hand, there exists the need to remove certain data, and the influence thereof, from ML models, as well.

There are many reasons why we want to remove data or the data's influence from ML models. For example, in adversarial settings, data used for training may be contaminated, or poisoned, by malicious data [5, 6], causing ML models to malfunction. Or, there may be too much data from unlikely scenarios that contributes to inefficient storage and sometimes even leads to wrong predictions [6]. Sometimes biased data could lead to discriminatory and unfair models, exacerbating the inequality in race, sex and religion [7]. In such scenarios, one may also need to remove some data and its influence to repair the model. In some cases, it is also about privacy, which is also increasingly discussed due to recent regulations.

With the introduction of regulations in multiple jurisdictions, such as the European Union's General Data Protection Regulation (GDPR) [8] and the California Consumer Privacy Act of 2018 (CCPA) [9], **Right to be Forgotten (RtbF)** or **right to erasure** has been established, in some parts of the world, where an entity may be required to erase data concerning certain people. In most cases, it is simply removing certain data from back-end databases. However, since ML models may memorize data [10], it is possible that a company may be requested to remove some individual's data from their ML models [6]. The process of making models "forget" that it has learnt from certain data is called **machine unlearning**, or simply, **unlearning**. This term was only put forward by [11] in 2015. Nevertheless, due to the sheer scale of ML applications in present days, it is worth looking into this topic.

1.2 Federated Learning

FL is a privacy-aware collaborative learning method first proposed by [12] in 2017, where participants jointly train a model without sharing data. The main idea is to have distributed datasets held by participants, each of whom generates a sub-update from training on its data, and then, either centrally [13] or decentrally [13, 14], build an ML model based on participants' sub-updates. FL helps accelerate model training speed and avoids direct privacy leakage [15]. However, due to the large amount of changes brought to the ML system, it is worth investigating the implications of those changes and what they mean to applying some of the ML privacy-preserving techniques to FL.

2 Outline

The remainder of this report will first define what problem this project is trying to solve. Then, it will discuss the objectives of this project. After that, it will introduce what types of technologies and resources have been and will be used to achieve the objectives. Later, the report will show what has been done from September to mid-January. Limitations of work already done will then be discussed, and some recommendations will be proposed. Next, there will be a conclusion on the current status of the project. Finally, a detailed future schedule will be included at the end. More details of the work will be included in appendices after the reference list.

3 Problem Definition

Since FL is not immune from some of the privacy vulnerabilities that other ML techniques may have [16] and it cannot replace machine unlearning as data's influence is recorded in sub-updates that are sent out during the FL process, it is important for unlearning to be introduced to FL. Yet, despite some efforts towards unifying the machine unlearning field [6, 17–21] and some research published in the sub-field of unlearning applications to FL [15, 22–27], there still lacks a comprehensive and in-depth study introducing and summarizing developments in the sub-field.

4 Project Objectives

This project takes aim at several educational and research objectives towards the promotion and betterment of the topic on unlearning in FL, a relatively new area that has seen some research developments in recent years. This project tries to provide materials unifying this sub-field and examine the recent research developments.

4.1 Educational Objective

The main educational objective of this project is to provide materials for learning the topic of unlearning in FL. The materials, including documentations and experiment instructions, code and scripts, will gather together the ML, security and privacy knowledge and outtakes from recent research publications in an interactive and organized way to allow readers, at different levels, to learn about this topic from basic knowledge to the front line of advanced research studies, so that people such as ML developers, researchers, as well as technology lawyers, managers and operators, can benefit from this project.

4.2 Research Objectives

The main research objective is to examine the unlearning in FL methods proposed in recent publications and compare their approaches to machine unlearning in FL. This project will draw conclusions from these experiments and present the findings in a clear and detailed fashion. After that, this project will try to find directions in which further research could be done that may improve the performance of unlearning in FL.

5 Methodology

5.1 Literature Review

The project searches for related research publications in Google Scholar, the most widely used academic search engine, using the keywords "federated" and "unlearning". Some of the relevant publications already found in search results were from arXiv, ACM Digital Library, IEEE Xplore, SpringerLink and official proceedings of conferences such as AAAI, NeurIPS and PMLR. This project uses HKUL E-resources to access these resources, if paywalled. This project also closely monitors news from the first IEEE Conference on Secure and Trustworthy Machine Learning (IEEE SaTML 2023), which will take place in February 2023, as it is closely related to the topic of this project.

5.2 Experiments

This project conducts experiments on HKUCS GPU Farm and with GPU sessions on Google Colab, as ML systems usually train faster on GPU due to large quantities of tensor calculation. This project uses testing and development environments constructed using technologies such as Miniconda, CUDA Toolkit and packages used in related publications. This project will conduct comparative experiments using different techniques proposed by different publications using datasets as summarized by [6], such as MNIST, CIFAR, SVHN and Adult. This project will modify code provided with reviewed publications to remove bugs and test the code's performance with different parameter settings.

5.3 Notebooks

This project uses Jupyter Notebooks (Figure 1) to document the concepts and design small interactive experiments. Instructions of larger experiments, or experiments requiring multiple terminal sessions to run, which may not be suitable to run directly from notebooks, will also be noted. Users will be directed to conduct their own experiments in a different setting. The notebooks will be systematically organized and put into a repository hosted on a cloud storage and sharing system. The project page will show the structure of the repository and have links to each notebook. Users will have the option to download or open the notebooks in Google Colab for convenience, whenever appropriate. Additional information on notebook setup will also be shown on the web page. Jupyter Notebook and Google Colab are chosen because they provide interactive, informative and easy-to-use functionalities for code execution and documentation.

5.4 Web Page

The project web page[1] (screenshot in Figure 2) is hosted on GitHub Pages with GitHub Actions, a world class CI/CD tool, and Hugo, which supports the Markdown language and is one of the most popular open-source static site generators. This set of technologies are chosen for their prevalence in the industry, ease of use and powerful functionalities.

[1] https://vicw0ng-hk.github.io/feul/

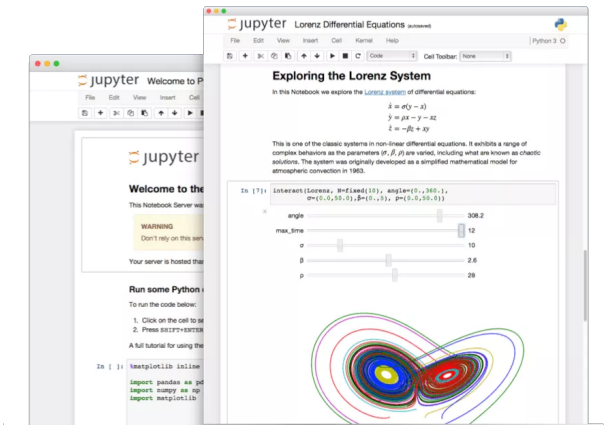

Figure 1: Jupyter Notebooks.[2]

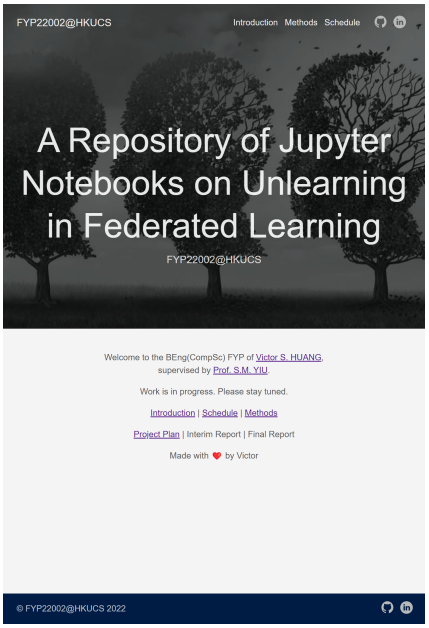

Figure 2: Project Web Page.

5.5 Reporting

This project uses LaTeX, a high-quality typesetting system and *de facto* standard for the communication and publication of scientific documents, to generate the reports required by the FYP course. More specifically, Overleaf will be used for its ease of use and support for a large amount of packages.

6 Progress and Results

6.1 Literature Review

6.1.1 Machine Unlearning

Goal of Machine Unlearning. A naive implementation of machine unlearning is to have the model retrained from scratch using remaining data, which is all the data but the ones we want to erase. This is obviously computationally expensive. However, one can learn from this scenario what unlearning outcomes we want to achieve. That is, we want models trained from scratch with dataset S after unlearning a set of data X to be "equivalent" to, or in the same distribution with [28], the model trained from scratch using dataset $S \setminus X$. We can also learn from this definition that there are subtle differences between machine unlearning and data deletion, with the former coming from a model perspective and the latter on a data basis [6]. To put it in layman's terms, we want to induce "amnesia", or loss of specific memories, in ML models, and ideally to do it efficiently and with precision, so as to uphold privacy, security, usability and fidelity in ML systems.

[2]Retrieved from https://jupyter.org/assets/homepage/jupyterpreview.webp.

Comparison with Differential Privacy. There is a different but related approach for addressing privacy issues in ML called **Differential Privacy (DP)** [29]. However, it must be stressed that DP is not the same as machine unlearning and their respective privacy guarantees are also different. In a machine unlearning case, the exact unlearning process will make sure that any influence the unlearnt data has had on the model is cleared completely [6], while ε-differential privacy for any non-zero ε only helps bound the influence any data point has, but such a bound remains non-zero; a 0-differential privacy, while it can achieve 0 influence, will defeat the purpose by making the learning algorithm unable to learn anything [6, 28].

Challenges of Machine Unlearning. Much like it is challenging to induce memory loss with precision and efficiency, it must be noted that machine unlearning is also very difficult [6]. There are a lot of challenges that need to be tackled. First, there is the **stochasticity of training**. It is unclear how we could trace and map a single data point's effect in the training, especially in complex models, such as Deep Neural Network (DNN), given a great amount of randomness during training [28]. Second, we have the **incrementality of training**. This means the effect data X_0 being trained at time t_0 has on the model will keep influencing later training at t_i where $t_0 < t_i$ with X_i. On the other hand, model training with X_0 at t_0 is also influenced by training prior to t_0. Determining what influence we should clear from the model if we want the model to unlearn X_0 is challenging [6]. In addition, recent studies found that an unlearnt model usually underperforms compared to models trained from scratch with remaining data, with performance growing worse as we make the model unlearn more data [30, 31]. This **catastrophic unlearning** [31] is difficult to prevent.

Overview of Machine Unlearning. A 2022 survey by Nguyen *et al.* [6] summarizes recent developments in the field, in which they divide machine unlearning strategies into three approaches: model-agnostic, model-intrinsic and data-driven methods. They compared different approaches in various unlearning scenarios, design requirements and unlearning requests. They also discussed unlearning applications, among which is unlearning federated learning, a topic on which this project will focus. They also discussed future aspects of machine learning, one of which is that federated unlearning is emerging through recent research [15, 22, 24, 26]. [6] provides a high-level overview of machine unlearning to help research the topic, which is helpful when considering improvements on proposed methods, as it may contain some ideas on unlearning in other forms of ML that we could modify and apply in FL.

6.1.2 Unlearning in FL

Additional challenges. Compared to unlearning in centralized ML, unlearning in FL is even harder, because it uses aggregation instead of raw gradients to calculate global weights, which can become tedious to handle when multiple clients participate [25]. Data partition and statistical heterogeneity in FL can also add to the complexity, namely the difference between vertical FL and horizontal FL, and non-IID data [32]. In addition, there could be overlapping data among participating clients, although most proposed methods assume the data to be removed exists solely in one client [6, 15, 22, 26].

Methods. Liu *et al.* [22] proposed the first unlearning method for FL, using calibration training to separate the target client's contribution to the central model. However, this method does not scale well, meaning its performance on complicated models, such as DNNs, is unsatisfactory. Wu *et al.* [26] came up with an unlearning method using knowledge distillation, without the need for participating client's data to be used in the unlearning process, which works well with more complex models, such as DNNs. Liu *et al.* [23] put forward a rapid retraining method to meet the requirements of unlearning, using L-BFGS algorithm to calculate a Hessian approximation with historical parameter updates. This method, however, does not scale well, either.

Verification. In a typical ML setting, unlearning verification can be done using a variety of methods, such as membership inference attacks [6, 22] and backdoor attacks [6, 15, 27]. However, this can be more complicated in FL. Introducing adversarial attacks for verification may undermine the security features of FL, and participation of clients may have subtle effect on the output space [25]. Hence, [25] proposes a verification mechanism that utilizes the same communication channels during training and can verify unlearning in a few rounds of communication.

6.2 Repository of Jupyter Notebooks

A repository of Jupyter Notebooks[3] has been created (Figure 3). There are 20 folders in total, including two folders storing the images used in the repository and additional references that may be helpful.

The first three folders contains Jupyter Notebooks on background research for the basic knowledge of unlearning and FL, including documentation of concepts and discussions, as well as code that can be directly run from Notebooks or instructions to run more complex scripts in terminal emulators.

The rest of the folders contain Jupyter Notebooks for the papers reviewed or to be reviewed, including the main concepts introduced and some sample code for readers to run or instructions for readers to try. These papers include [15, 22–24, 26, 33] reviewed and [25, 27, 34–39] to be reviewed.

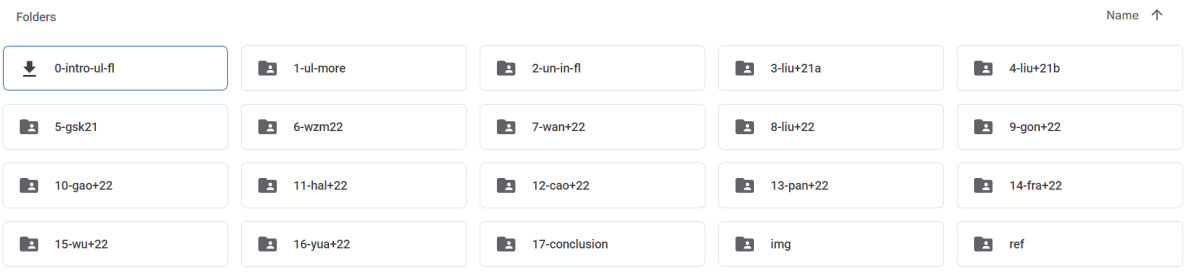

Figure 3: Screenshot of the repository hosted on Google Drive.

[3] https://drive.google.com/drive/folders/1swP7focASoFLV2eiyvXps47qyL8hoSUY

Figure 4: Contents of folder **0-intro-ul-fl**.

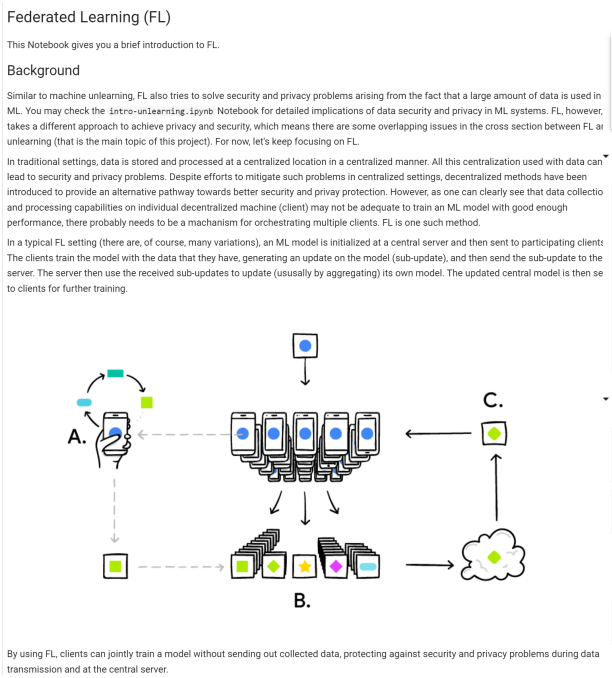

Figure 5: **intro-fl.ipynb**.

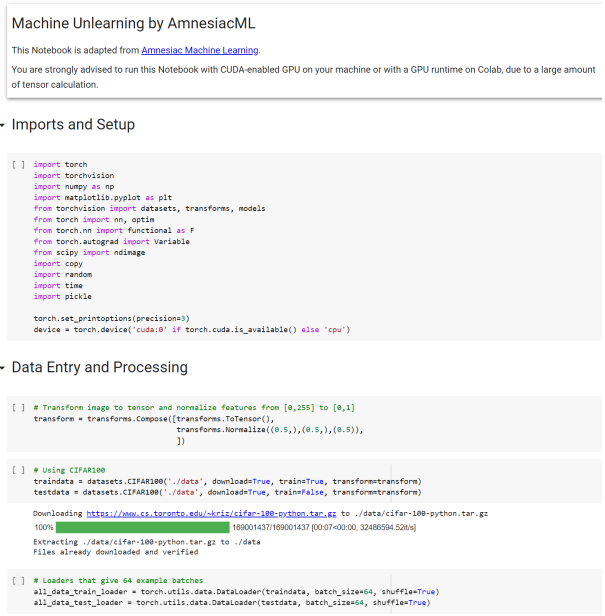

Figure 6: **code-amnesiac-ml.ipynb**.

7 Limitations and Recommendations

First, few papers reviewed have provided implementation code, making it difficult to conduct quantitative research among different methods proposed. Second, these papers don't necessarily use the same evaluation metrics, datasets, standards and environments to evaluate their work, thus creating another variable factor for comparison. Last but not least, some papers only target very specific unlearning requests, ML models, FL data partition, learning algorithm and other aspects of FL, making them less useful when considering the general field, as generalization of their work is more difficult.

The recommendations are to try to build code for certain papers according to the code of other papers which have a similar design of algorithm, try to use more diverse experiment designs when evaluating a single piece of code and use more general designs when doing comparisons, and to open separate discussions for less generalized work.

8 Conclusion

This project focuses on providing materials unifying the field of unlearning in FL. To that end, literature review was conducted on recent publications discussing the topic and a repository of Jupyter Notebooks has been created, half the content of which has been completed. Work done so far has been satisfactory at achieving the stated goal. More work is expected to complete the rest of the project next semester.

9 Future Schedule

Table 1 shows the tentative schedule based on department requirements and specific situations with this project. In construction phase from January to April, more advancements will be made based on previous stages. With more reading and experiments, more of the topic will be explored and tested. Final presentation and report will be prepared. In late April, preparations will be made for project exhibition.

Tentative Future Schedule		
Time	**Milestones**	**Notes**
- 02 Apr 2023	Comprehensively test and compare methods in readings	
	Summarize knowledge and write in Notebooks	
	Explore future research and development directions	
	Develop interactive learning materials in Notebooks	
- 16 Apr 2023	Prepare for Final Presentation	Final Presentation 17-21 Apr 2023
- 18 Apr 2023	Write Final Report and Wrap up final implementation	Deliverable 3 due 18 Apr 2023
- 02 May 2023	Prepare for Project Exhibition	

Table 1: Tentative Project Schedule updated on Sun 22nd Jan 2023

10 Acknowledgement

This project owes much gratitude to the supervisor, Prof. S.M. YIU, Associate Head of Department of Computer Science, HKU (HKUCS), as well as the second examiner, Dr. Tao YU, Assistant Professor at HKUCS.

This project was partially inspired by an HKUCS FYP in 2020-21, *Building a code and data repository for teaching algorithmic trading*[4] by Woo, Wu and Lee, which won champion of 2020-21 FYP competition at HKUCS.

[4] https://awoo424.github.io/algotrading_fyp/

References

[1] J. A. Castellanos-Garzón, E. Costa, J. L. Jaimes S. and J. M. Corchado, 'An evolutionary framework for machine learning applied to medical data,' *Knowledge-Based Systems*, vol. 185, p. 104 982, 2019, ISSN: 0950-7051. DOI: 10.1016/j.knosys.2019.104982. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705119304046.

[2] B. Arslan, E. Yorulmaz, B. Akca and S. Sagiroglu, 'Security perspective of biometric recognition and machine learning techniques,' in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2016, pp. 492–497. DOI: 10.1109/ICMLA.2016.0087.

[3] P. Zola, P. Cortez and M. Carpita, 'Twitter user geolocation using web country noun searches,' *Decision Support Systems*, vol. 120, pp. 50–59, 2019, ISSN: 0167-9236. DOI: https://doi.org/10.1016/j.dss.2019.03.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167923619300442.

[4] H. M. Gomes, J. Read, A. Bifet, J. P. Barddal and J. Gama, 'Machine learning for streaming data: State of the art, challenges, and opportunities,' *SIGKDD Explor. Newsl.*, vol. 21, no. 2, pp. 6–22, Nov. 2019, ISSN: 1931-0145. DOI: 10.1145/3373464.3373470. [Online]. Available: https://doi-org.eproxy.lib.hku.hk/10.1145/3373464.3373470.

[5] P. McDaniel, N. Papernot and Z. B. Celik, 'Machine learning in adversarial settings,' *IEEE Security & Privacy*, vol. 14, no. 3, pp. 68–72, 2016. DOI: 10.1109/MSP.2016.51.

[6] T. T. Nguyen, T. T. Huynh, P. L. Nguyen, A. W.-C. Liew, H. Yin and Q. V. H. Nguyen, *A survey of machine unlearning*, 2022. DOI: 10.48550/ARXIV.2209.02299. [Online]. Available: https://arxiv.org/abs/2209.02299.

[7] S. Verma, M. Ernst and R. Just, *Removing biased data to improve fairness and accuracy*, 2021. DOI: 10.48550/ARXIV.2102.03054. [Online]. Available: https://arxiv.org/abs/2102.03054.

[8] European Parliament and Council of the European Union, *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)*, 2016. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[9] California State Legislature, *An act to add title 1.81.5 (commencing with section 1798.100) to part 4 of division 3 of the civil code, relating to privacy.* 2018. [Online]. Available: https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375.

[10] S. Chatterjee, 'Learning and memorization,' in *Proceedings of the 35th International Conference on Machine Learning*, J. Dy and A. Krause, Eds., ser. Proceedings of Machine Learning Research, vol. 80, PMLR, Jul. 2018, pp. 755–763. [Online]. Available: https://proceedings.mlr.press/v80/chatterjee18a.html.

[11] Y. Cao and J. Yang, 'Towards making systems forget with machine unlearning,' in *2015 IEEE Symposium on Security and Privacy*, 2015, pp. 463–480. DOI: 10.1109/SP.2015.35.

[12] B. McMahan, E. Moore, D. Ramage, S. Hampson and B. A. y. Arcas, 'Communication-Efficient Learning of Deep Networks from Decentralized Data,' in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, A. Singh and J. Zhu, Eds., ser. Proceedings of Machine Learning Research, vol. 54, PMLR, Apr. 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html.

[13] P. Kairouz *et al.*, 'Advances and open problems in federated learning,' *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021, ISSN: 1935-8237. DOI: 10.1561/2200000083. [Online]. Available: http://dx.doi.org/10.1561/2200000083.

[14] S. R. Pokhrel and J. Choi, 'Federated learning with blockchain for autonomous vehicles: Analysis and design challenges,' *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4734–4746, 2020. DOI: 10.1109/TCOMM.2020.2990686.

[15] Y. Liu, Z. Ma, Y. Yang, X. Liu, J. Ma and K. Ren, 'Revfrf: Enabling cross-domain random forest training with revocable federated learning,' *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2021. DOI: 10.1109/TDSC.2021.3104842.

[16] B. Hitaj, G. Ateniese and F. Perez-Cruz, 'Deep models under the gan: Information leakage from collaborative deep learning,' in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17, Dallas, Texas, USA: Association for Computing Machinery, 2017, pp. 603–618, ISBN: 9781450349468. DOI: 10.1145/3133956.3134012. [Online]. Available: https://doi.org/10.1145/3133956.3134012.

[17] S. Mercuri *et al.*, *An introduction to machine unlearning*, 2022. DOI: 10.48550/ARXIV.2209.00939. [Online]. Available: https://arxiv.org/abs/2209.00939.

[18] S. Schelter, '"amnesia" - machine learning models that can forget user data very fast,' in *10th Conference on Innovative Data Systems Research, CIDR 2020, Amsterdam, The Netherlands, January 12-15, 2020, Online Proceedings*, www.cidrdb.org, 2020. [Online]. Available: http://cidrdb.org/cidr2020/papers/p32-schelter-cidr20.pdf.

[19] S. Shintre, K. A. Roundy and J. Dhaliwal, 'Making machine learning forget,' in *Privacy Technologies and Policy*, M. Naldi, G. F. Italiano, K. Rannenberg, M. Medina and A. Bourka, Eds., Cham: Springer International Publishing, 2019, pp. 72–83, ISBN: 978-3-030-21752-5.

[20] M. Veale, R. Binns and L. Edwards, 'Algorithms that remember: Model inversion attacks and data protection law,' *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 376, no. 2133, p. 20 180 083, 2018. DOI: 10.1098/rsta.2018.0083. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rsta.2018.0083. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2018.0083.

[21] E. F. Villaronga, P. Kieseberg and T. Li, 'Humans forget, machines remember: Artificial intelligence and the right to be forgotten,' *Computer Law & Security Review*, vol. 34, no. 2, pp. 304–313, 2018, ISSN: 0267-3649. DOI: https://doi.org/10.1016/j.clsr.2017.08.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0267364917302091.

[22] G. Liu, X. Ma, Y. Yang, C. Wang and J. Liu, 'Federaser: Enabling efficient client-level data removal from federated learning models,' in *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, 2021, pp. 1–10. DOI: 10.1109/IWQOS52092.2021.9521274.

[23] Y. Liu, L. Xu, X. Yuan, C. Wang and B. Li. 'The right to be forgotten in federated learning: An efficient realization with rapid retraining.' arXiv: 2203.07320. (2022).

[24] J. Wang, S. Guo, X. Xie and H. Qi, 'Federated unlearning via class-discriminative pruning,' in *Proceedings of the ACM Web Conference 2022*, ser. WWW '22, Virtual Event, Lyon, France: Association for Computing Machinery, 2022, pp. 622–632, ISBN: 9781450390965. DOI: 10.1145/3485447.3512222. [Online]. Available: https://doi.org/10.1145/3485447.3512222.

[25] X. Gao *et al.* 'Verifi: Towards verifiable federated unlearning.' arXiv: 2205.12709. (2022).

[26] C. Wu, S. Zhu and P. Mitra. 'Federated unlearning with knowledge distillation.' arXiv: 2201.09441. (2022).

[27] A. Halimi, S. Kadhe, A. Rawat and N. Baracaldo, *Federated unlearning: How to efficiently erase a client in fl?* 2022. DOI: 10.48550/ARXIV.2207.05521. [Online]. Available: https://arxiv.org/abs/2207.05521.

[28] L. Bourtoule *et al.*, 'Machine unlearning,' in *2021 IEEE Symposium on Security and Privacy (SP)*, 2021, pp. 141–159. DOI: 10.1109/SP40001.2021.00019.

[29] C. Dwork, 'Differential privacy: A survey of results,' in *Theory and Applications of Models of Computation*, M. Agrawal, D. Du, Z. Duan and A. Li, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1–19.

[30] Q. P. Nguyen, R. Oikawa, D. M. Divakaran, M. C. Chan and B. K. H. Low, 'Markov chain monte carlo-based machine unlearning: Unlearning what needs to be forgotten,' ser. ASIA CCS '22, Nagasaki, Japan: Association for Computing Machinery, 2022, pp. 351–363, ISBN: 9781450391405. DOI: 10.1145/3488932.3517406. [Online]. Available: https://doi.org/10.1145/3488932.3517406.

[31] Q. P. Nguyen, B. K. H. Low and P. Jaillet, 'Variational bayesian unlearning,' in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 16 025–16 036. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/b8a6550662b363eb34145965d64d0cfb-Paper.pdf.

[32] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li and Y. Gao, 'A survey on federated learning,' *Knowledge-Based Systems*, vol. 216, p. 106 775, 2021, ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2021.106775. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705121000381.

[33] J. Gong, O. Simeone and J. Kang, 'Bayesian variational federated learning and unlearning in decentralized networks,' in *2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2021, pp. 216–220. DOI: 10.1109/SPAWC51858.2021.9593225.

[34] J. Gong, O. Simeone, R. Kassab and J. Kang, *Forget-svgd: Particle-based bayesian federated unlearning*, 2021. DOI: 10.48550/ARXIV.2111.12056. [Online]. Available: https://arxiv.org/abs/2111.12056.

[35] X. Cao, J. Jia, Z. Zhang and N. Z. Gong, *Fedrecover: Recovering from poisoning attacks in federated learning using historical information*, 2022. DOI: 10.48550/ARXIV.2210.10936. [Online]. Available: https://arxiv.org/abs/2210.10936.

[36] C. Pan, J. Sima, S. Prakash, V. Rana and O. Milenkovic, *Machine unlearning of federated clusters*, 2022. DOI: 10.48550/ARXIV.2210.16424. [Online]. Available: https://arxiv.org/abs/2210.16424.

[37] Y. Fraboni, R. Vidal, L. Kameni and M. Lorenzi, *Sequential informed federated unlearning: Efficient and provable client unlearning in federated optimization*, 2022. DOI: 10.48550/ARXIV.2211.11656. [Online]. Available: https://arxiv.org/abs/2211.11656.

[38] L. Wu, S. Guo, J. Wang, Z. Hong, J. Zhang and Y. Ding, 'Federated unlearning: Guarantee the right of clients to forget,' *IEEE Network*, vol. 36, no. 5, pp. 129–135, 2022. DOI: 10.1109/MNET.001.2200198.

[39] W. Yuan, H. Yin, F. Wu, S. Zhang, T. He and H. Wang, *Federated unlearning for on-device recommendation*, 2022. DOI: 10.48550/ARXIV.2210.10958. [Online]. Available: https://arxiv.org/abs/2210.10958.

Appendix A

Useful Resources

- Privacy and Security in ML Seminars - Privacy & Security in Machine Learning (PriSec-ML) Interest Group
- Virtual Seminar Series - Challenges and Opporunities for Security & Privacy in Machine Learning
- IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)
- the cleverhans blog - a blog by Ian Goodfellow and Nicolas Papernot about security and privacy in machine learning
- ECE1784H/CSC2559H: Trustworthy Machine Learning Fall 2022 - University of Toronto
- Awesome Machine Unlearning - GitHub Repo

Related HKUCS FYPs

- FYP22019: Quantitative Performance and Security Evaluation of Federated Learning on open-sourced platforms (Industry-based Project) by Fong 2022-23
- FYP20009: Building a code and data repository for teaching algorithmic trading by Woo, Wu and Lee 2020-21

Appendix B

FL [12]

The `FederatedAveraging` Algorithm

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
initialize w_0
for each round $t = 1, 2, \dots$ **do**
  $m \leftarrow \max(C \cdot K, 1)$
  $S_t \leftarrow$ (random set of m clients)
  for each client $k \in S_t$ **in parallel do**
    $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
  $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate(k, w): // *Run on client* k
$\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
for each local epoch i from 1 to E **do**
  for batch $b \in \mathcal{B}$ **do**
    $w \leftarrow w - \eta \nabla \ell(w; b)$
return w to server

Federated Optimization

Federated optimization has several key properties that differentiate it from a typical distributed optimization problem:

- **Non-IID** The training data on a given client is typically based on the usage of the mobile device by a particular user, and hence any particular user's local dataset will not be representative of the population distribution.
- **Unbalanced** Similarly, some users will make much heavier use of the service or app than others, leading to varying amounts of local training data.
- **Massively distributed** We expect the number of clients participating in an optimization to be much larger than the average number of examples per client.
- **Limited communication** Mobile devices are frequently offline or on slow or expensive connections.

Appendix C

Unlearning Framework [6]

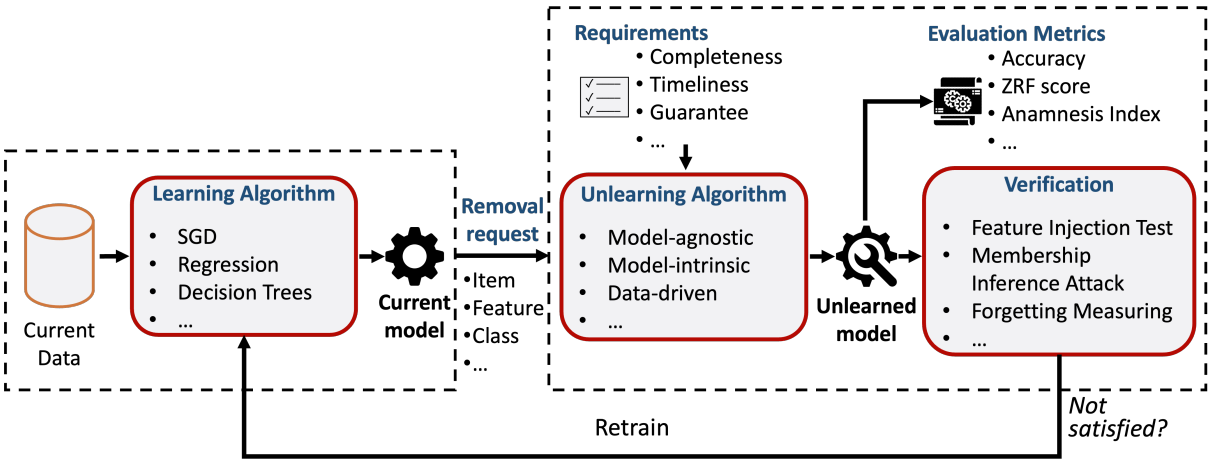

Figure 7: A Machine Unlearning Framework

FedEraser [22]

Algorithm 2 FedEraser

Require: Initial global model $\mathcal{M}^1$; retained client updates U
Require: Target client index k_u
Require: Number of global calibration round T
Require: Number of local calibration training epoch E_{cali}
Central server executes:
  for each round $R_{t_j}, j \in \{1, 2, \cdots, T\}$ **do**
    for each client $C_{k_c}, k_c \in \{1, 2, \cdots, K\} \setminus k_u$ **in parallel do**
      $\widehat{U}_{k_c}^{t_j} \leftarrow \text{CaliTrain}(C_{k_c}, \widetilde{\mathcal{M}}_{k_c}^{t_j}, E_{cali})$
      $\widetilde{U}_{k_c}^{t_j} \leftarrow |U_{k_c}^{t_j}| \frac{\widehat{U}_{k_c}^{t_j}}{||\widehat{U}_{k_c}^{t_j}||}$ {Update Calibrating}
    end
    $\widetilde{\mathcal{U}}^{t_j} \leftarrow \frac{1}{(K-1)\sum w_{k_c}} \sum_{k_c} w_{k_c} \widetilde{U}_{k_c}^{t_j}$ {Update Aggregating}
    $\widetilde{\mathcal{M}}^{t_{j+1}} \leftarrow \widetilde{\mathcal{M}}^{t_j} + \widetilde{\mathcal{U}}^{t_j}$ {Model Updating}
  end

CaliTrain$(C_{k_c}, \widetilde{\mathcal{M}}_{k_c}^{t_j}, E_{cali})$: // Run on client C_{k_c}
  for each local training round j from 1 to E_{cali} **do**
    $\widetilde{\mathcal{M}}_{k_c}^{t_j}|_{j+1} \leftarrow Train(\widetilde{\mathcal{M}}_{k_c}^{t_j}|_j, D_{k_c})$
  end
  $\widehat{U}_{k_c}^{t_j} \leftarrow$ Calculating Update$(\widetilde{\mathcal{M}}_{k_c}^{t_j}|_{E_{cali}}, \widetilde{\mathcal{M}}_{k_c}^{t_j}|_1)$
  return $\widehat{U}_{k_c}^{t_j}$ to the central server

Federated Unlearning with Knowledge Distillation [26]

$$M_F = M_1 + \sum_{t=1}^{F-1} \Delta M_t \tag{1}$$

$$\Delta M_t = \frac{1}{N}\sum_{i=1}^{N} \Delta M_t^i = \frac{1}{N}\sum_{i=1}^{N-1} \Delta M_t^i + \frac{1}{N}\Delta M_t^N$$

$$\Delta M_t' = \frac{1}{N-1}\sum_{i=1}^{N-1} \Delta M_t^i = \frac{N}{N-1}\Delta M_t - \frac{1}{N-1}\Delta M_t^N$$

Assume client N still participated in the training process but set his updates $\Delta M_t^N = 0$ for all rounds.

$$\Delta M_t' = \frac{1}{N}\sum_{i=1}^{N-1} \Delta M_t^i = \Delta M_t - \frac{1}{N}\Delta M_t^N$$

A combination of the above formula with Equation 1 gives us the unlearning result of the final global model M_F'.

$$\begin{aligned} M_F' &= M_1 + \sum_{t=1}^{F-1} \Delta M_t' + \sum_{t=1}^{F-1} \epsilon_t \\ &= M_1 + \sum_{t=1}^{F-1} \Delta M_t - \frac{1}{N}\sum_{t=1}^{F-1} \Delta M_t^N + \sum_{t=1}^{F-1} \epsilon_t \\ &= M_F - \frac{1}{N}\sum_{t=1}^{F-1} \Delta M_t^N + \sum_{t=1}^{F-1} \epsilon_t \end{aligned}$$

Algorithm 3 Federated Unlearning with Knowledge Distillation

Input: Global model M_F, Total number of clients N
Input: Historical updates ΔM_t^A of target client A at round t
Input: Outsourced unlabelled dataset X
Parameter: Distillation epoch k, Temperature T
Output: The unlearning model M_F'

$M_F' \leftarrow M_F - \frac{1}{N}\sum_{t=1}^{F-1} \Delta M_t^A$
for *epoch* $= 1, 2, \ldots, k$ **do**
    $y_{teacher} \leftarrow M_F(X), T$
    $y_{student} \leftarrow M_F'(X), T$
    Calculate $loss_{distillation}$ of $y_{teacher}$ and $y_{student}$
    Back-propagate model M_F'
return unlearning model M_F'