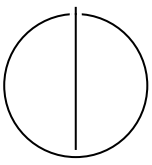
SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

# Adaptation Techniques for using NAS Methods in the FL Setting

Max Coetzee



SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

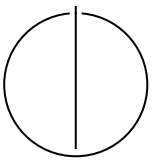
TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

# Adaptation Techniques for using NAS Methods in the FL Setting

## Titel

| | |
|---|---|
| Author: | Max Coetzee |
| Examiner: | Supervisor |
| Supervisor: | Advisor |
| Submission Date: | Submission date |



I confirm that this bachelor's thesis is my own work and I have documented all sources and material used.

Munich, Submission date Max Coetzee

# Acknowledgments

# Abstract

[TODO: overhaul]

Applying techniques from Neural Architecture Search (NAS) to Federated Learning (FL) has been fruitful (remove) in recent years. The combination was identified as a promising research (remove) direction by [11]. It has yielded methods for finding architectures that deal with the challenges imposed by the FL setting.

Research into NAS has grown rapidly [19] since it was popularized by [22]; consequently, literature on its application to FL has grown. The last survey on NAS applied to FL compared approaches of four papers [21]. Since then, we have identified approximately 50 new papers. This motivates a new systematic survey of the landscape to identify progress and gaps in the literature.

In this thesis, we propose a map of the literature landscape based on the FL challenges they address. We achieve this by systematically evaluating the literature and identifying which challenge it solves.

We refer to the FL challenges described in [22], i.e., non-IID data, limited communication, client heterogeneity, privacy of client data, and break them down into smaller subchallenges — each subchallenge being associated with a pattern in the literature. We include personalized FL [17] as an additional subchallenge that was not originally posited, but has since drawn the community's attention.

We then analyze how the subchallenges are addressed and focus on the contribution of the used NAS method towards overcoming the subchallenge. For each subchallenge, we keep track of the NAS types used (following [19], [2]) and assess whether the underexplored methods are candidates for future research.

Neural Architecture Search

# Contents

# 1 Introduction

Both *Neural Architecture Search* (NAS) and *Federated Learning* (FL) have made significant progress independently in the past decade. To benefit from the advantages of NAS methods in FL, researchers have started combining them by using NAS in FL [8] [20] [15] [5] [13] [18].

NAS automates the process of engineering neural network architectures for Deep Learning application domains [6]. This saves manual labour compared to the traditional approach driven by a team of experts.

FL is a machine learning method whereby *clients* collaboratively train a model without sharing their local data. FL enables privacy-preserving machine learning on the increasing volume of privacy-sensitive, distributed data and avoids centrally collecting client data.

Using NAS in FL reduces manual effort for practitioners and can be used to mitigate the issues with predefined architectures in FL. Since client data is invisible to practitioners and client data distributions as well as client hardware vary in FL, predefined architectures are often suboptimal. Some NAS methods can be used to tailor architectures to clients and their data and improve FL training effectivenes.

Despite the potential, NAS research has focused on the traditional centralised setting as opposed to the FL setting. This makes many NAS methods infeasible for direct application in FL, because centralised NAS can make several assumptions about the search process that do not hold in the FL setting. Each assumption discrepancy creates a *challenge* for using NAS in FL and researchers have developed *adaptation techniques* for overcoming them. Applying adaptation techniques to NAS methods results in *Federated Neural Architecture Search* [8] (FedNAS) methods.

For example, one challenge arises from the fact that centralised NAS can assume worker nodes are computationally powerful, whereas clients in most FL settings are not. Since most centralised NAS methods place large computational burdens on worker nodes, practitioners using these NAS methods in FL without modification would experience detrimental search completion times. Researchers have developed adaptation techniques for reducing the computational burden on individual clients in FedNAS methods [18] [5] [9] [13]. This typically involves reducing the overall computational work and splitting it up into smaller units, which manifests as a challenge, because the implementation of the resulting FedNAS method tends to be complex.

The subset of challenges faced by FedNAS methods depend on the specific FL setting, which can differ in many aspects [11]. Each FL setting violates centralised NAS assumptions to a different extent, making some challenges more relevant than others for that setting. For example, two major classes of FL settings appear in the literature: the *cross-device* class, wherein clients are edge devices, and the *cross-silo* class wherein clients are entire organisations. For FL settings in the cross-silo class, clients can be expected to be equipped with GPUs, making the challenge described above less relevant.

FedNAS Researchers need to prioritise which subset of challenges to address for their targeted FL setting: the relevance of a challenge depends on how strongly the corresponding centralised NAS assumption is violated, and overcoming one challenge typically comes at the expense of neglecting others. However, the literature is fragmented and often lacks clarity regarding the targeted FL setting and challenges addressed. As a result extending and re-using existing techniques remains difficult. Prior literature surveys [21] [12] [7] do not focus on adaptation techniques and only analyse a small fraction of the growing FedNAS literature. To aid in the development new FedNAS methods, we set out to answer our research question:

**What challenges arise from particular FL settings for FedNAS methods in the literature and which adaptation techniques are used to address them?**

To answer our research question, we conduct a systematic literature review using grounded theory and lean on the methodology employed by [10].

We first identify papers that present FedNAS methods. Then, we perform open coding on each FedNAS method to extract unrefined adaptation techniques. Next, we perform axial coding by iteratively refining and merging unrefined adaptation techniques to obtain a coherent set of mutually exclusive and collectively exhaustive adaptation techniques.

We then analyse how each adaptation technique works towards, against, or does not affect each FedNAS challenge, and aggregate these coded effects into two overview tables: one for FedNAS methods and one for adaptation techniques. These tables help practitioners decide on FedNAS methods suited to their use case or adaptation techniques that practitioners can use to create new FedNAS methods suited to their use case.

Our review organises the n extracted adaptation techniques into a consolidated body of knowledge that provides FedNAS practitioners with an overview of the FedNAS landscape through the lens of adaptation techniques. Compared to existing surveys, our review is exhaustive of the FedNAS landscape published up to 2025. Additionally, our review of each adaptation technique and our overview tables help practitioners

find and choose FedNAS methods relevant to their use case or construct new FedNAS methods by re-using appropriate adaptation techniques.

In Chapter 2, we cover the background required for this thesis and related work. In Chapter 3, we describe the method with which we conduct our literature review in detail. In Chapter 4, we describe our process for including FedNAS literature and provide an overview of the reviewed FedNAS literature. In Chapter 5, we present our taxonomy of adaptation techniques and our review of how adaption techniques overcome FedNAS challenges. In Chapter 6, we conduct a discussion about our work. Chapter 7 contains our conclusion.

# 2 Background and Related Work

NAS is contained entirely in the 5th step of the FL pipeline as described in [11].

## 2.1 Neural Architecture Search

## 2.2 Federated Learning

## 2.3 Federated Neural Architecture Search

# 3 Method

We use various techniques from [3] and lean on parts of the methodology of [10] to perform a systematic literature review.

1. **Literature Selection:** We follow the guidelines and flow diagrams provided by PRISMA 2020 [16] for inclusion and exclusion of papers and perform forward and backwards citation searching. We identify 58 papers that present FedNAS methods. Each paper contains one or more FedNAS methods.

2. **Unrefined Adaptation Technique Extraction:** Once the set of included papers is fixed, we perform open coding on each FedNAS method to extract unrefined adaptation techniques. Any modification to a NAS method that is explicitly motivated by the federated setting is treated coded as one unrefined adaptation technique.

3. **Adaptation Techniques Coneceptualization:** We iteratively refine and merge unrefined adapatation techniques in an axial coding step to obtain a coherent set of adaptation techniques that are mutually exclusive and collectively exhaustive. Unrefined adaptation techniques with conceptually highly-similar mechanisms are merged into a single representative adaptation technique.

4. **Categorise Adaptation Techniques:** After merging, in a second axial coding step, we cluster adaptation techniques based on a) the FedNAS challenges they address and b) the conceptual similarity of their mechanisms. As a result, we obtain a taxonomy of adaptation techniques.

5. **Discuss FedNAS Challenges for Adaptation Techniques:** We discuss how each adaptation technique works towards, against, or does not affect overcoming each of FedNAS challenges.

6. **Produce Table Overviews:** Finally, we create two tables that practicioners can use to make decisions about FedNAS methods for their use case.

   The first table contains a coded vector of effects over the FedNAS challenges for each *FedNAS method*. The effects per FedNAS method are the result of the effects of its adaptation techniques in aggregate. Practicioners can use this table

to decide which FedNAS method suits their use case or if their use case would benefit from a new FedNAS method.

The second table table contains a coded vector of effects over the FedNAS challenges for each *adaptation technique* based on the prior discussion. If practicioners need to create new FedNAS methods, they can use this table to choose existing adaptation techniques relevant to the set of FedNAS challenges they need to address for their use case.

# 4 Reviewed Literature

add some overview and statistics about the literature

# 5 Adaptation Techniques

## 5.1

| Assumption in Centralised Setting | Reality in FL Setting | Resulting FedNAS Challenge Description | FedNAS Challenge Name |
|---|---|---|---|
| Worker nodes' hardware is homogeneous. | Clients' hardware varies significantly. More pronounced in the cross-device setting than the cross-silo setting. | FedNAS methods need to be heterogeneity-aware; otherwise, the search will be delayed by low-end devices or leave high-end devices waiting in idle most of the time. | Hardware Heterogeneity |
| Worker nodes communicate over high-bandwidth, low-latency links. | Clients typically communicate with the central server over high-latency and low-bandwidth connections. | FedNAS must be communication-efficient; otherwise, exchanging model weights and architecture parameters becomes a bottleneck during the search process. | Limited Networking Capabilities |
| Worker nodes are high-end machines with powerful CPUs, GPUs, and large amounts of RAM. | Clients are edge devices with few computational resources. | The computational burden placed on clients by a NAS method needs to be adjusted such that the architecture search takes place within an acceptable time frame. This can involve splitting up computational work, usually done on single worker nodes, amongst multiple clients, making the implementation complex. | Limited Computational Resources |
| Training data is gathered at a central location. | Training data is distributed unevenly across clients | Since some clients have more data than others, FedNAS methods must ensure that these clients are not overrepresented in the searched architecture. | Unbalanced Client Data |

| | | | |
|---|---|---|---|
| The training data is drawn from the same distribution. | Each client draws training data from their own distribution. | FedNAS methods need to be robust with respect to non-i.i.d. data, which makes evaluating and ranking candidate architectures noisier than in the centralised setting. | Non-I.I.D. Training Data |
| Worker nodes are equally available. | Some clients are more frequently available than others. | FedNAS methods must prevent the search from being dominated by the most frequently available clients; otherwise, the resulting architecture will be biased towards them. | Variable Client Availability |
| All worker nodes participate in each iteration. | Only a subset of clients participates in each iteration of the search. | FedNAS methods must ensure that a realistic sample of the client population is represented in the architecture search and address high-variance performance estimates, as candidate architectures are evaluated on changing subsets of clients. | Client Participation |
| Worker nodes are inside the same trust domain. | All participating parties (i.e. the central server and clients) can consider another potentially malicious. | FedNAS methods must address attacks from participating parties, such as architecture parameter poisoning during the search process. | Security |
| Worker nodes consistently participate in the search process. | Clients can drop out of a communication round at any time, and completion times of epochs vary. | FedNAS methods need to handle client dropouts and stragglers, since interrupted or delayed evaluations of candidate architectures can slow down or destabilise the search process. | Client Reliability |

| | | | |
|---|---|---|---|
| Model accuracy is more important than model resource consumption in selecting architectures. | Model accuracy needs to be traded for model resource consumption in selecting architectures. | Models trained in FL tend to be deployed on the same resource-constrained clients they are trained on. Therefore FedNAS methods need to optimise the architecture of the trained model w.r.t. resource constraints. | Model Resource Constraints |

Table 5.1: Discrepancies between NAS in the centralised setting and the FL setting and the resulting FedNAS challenges. *Worker nodes* perform the architecture search in the centralised setting, whereas a *server* and *clients* perform the search in the FL setting. The challenges are based to some extent on previously identified challenges in the FL setting in general [14] [11] [4] [1].

# 6 Discussion

# 7 Conclusion

# Abbreviations

# List of Figures

# List of Tables

# Bibliography


[1] M. Arbaoui, M.-A. Brahmia, A. Rahmoun, and M. Zghal. "Federated learning survey: A multi-level taxonomy of aggregation techniques, experimental insights, and future frontiers." In: *ACM Trans. Intell. Syst. Technol.* 15.6 (Nov. 2024). ISSN: 2157-6904. DOI: 10.1145/3678182.

[2] S. S. P. Avval, N. D. Eskue, R. M. Groves, and V. Yaghoubi. "Systematic review on neural architecture search." In: *Artificial Intelligence Review* 58.3 (Jan. 6, 2025), p. 73. ISSN: 1573-7462. DOI: 10.1007/s10462-024-11058-w.

[3] J. Corbin and A. Strauss. *Basics of qualitative research*. Core textbook v. 14. SAGE Publications, 2015. ISBN: 9781412997461.

[4] K. Daly, H. Eichner, P. Kairouz, H. B. McMahan, D. Ramage, and Z. Xu. "Federated learning in practice: Reflections and projections." In: *2024 IEEE 6th international conference on trust, privacy and security in intelligent systems, and applications (TPS-ISA)*. 2024, pp. 148–156. DOI: 10.1109/TPS-ISA62245.2024.00026.

[5] L. Dudziak, S. Laskaridis, and J. Fernandez-Marques. *FedorAS: Federated architecture search under system heterogeneity*. 2022. arXiv: 2206.11239 [cs.LG].

[6] T. Elsken, J. H. Metzen, and F. Hutter. "Neural architecture search: A survey." In: *Journal of Machine Learning Research* 20.55 (2019), pp. 1–21.

[7] M. Hartmann, G. Danoy, and P. Bouvry. *Multi-objective methods in Federated Learning: A survey and taxonomy*. 2025. arXiv: 2502.03108 [cs.LG].

[8] C. He, M. Annavaram, and S. Avestimehr. *Towards Non-I.I.D. and invisible data with FedNAS: Federated deep learning via neural architecture search*. 2021. arXiv: 2004.08546 [cs.LG].

[9] C. Huo, J. Jia, T. Deng, M. Dong, Z. Yu, and D. Yuan. "NASFLY: On-device split federated learning with neural architecture search." In: *2024 IEEE international symposium on parallel and distributed processing with applications (ISPA)*. 2024, pp. 2184–2190. DOI: 10.1109/ISPA63168.2024.00298.

[10] D. Jin, N. Kannengießer, S. Rank, and A. Sunyaev. "Collaborative distributed machine learning." In: 57.4 (Dec. 2024). ISSN: 0360-0300. DOI: 10.1145/3704807.

[11] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, H. Eichner, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao. *Advances and Open Problems in Federated Learning*. 2021. arXiv: 1912.04977 [cs.LG].

[12] S. Khan, A. Rizwan, A. N. Khan, M. Ali, R. Ahmed, and D. H. Kim. "A multi-perspective revisit to the optimization methods of Neural Architecture Search and Hyper-parameter optimization for non-federated and federated learning environments." In: *Computers and Electrical Engineering* 110 (2023), p. 108867. ISSN: 0045-7906. DOI: https://doi.org/10.1016/j.compeleceng.2023.108867.

[13] J. Liu, J. Yan, H. Xu, Z. Wang, J. Huang, and Y. Xu. "Finch: Enhancing federated learning with hierarchical neural architecture search." In: *IEEE Transactions on Mobile Computing* 23.5 (2024), pp. 6012–6026. DOI: 10.1109/TMC.2023.3315451.

[14] 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*. Ed. by S. Aarti and Z. Jerry. Vol. 54. Proceedings of machine learning research. PMLR, Apr. 2017, pp. 1273–1282.

[15] E. Mushtaq, C. He, J. Ding, and S. Avestimehr. *SPIDER: Searching personalized neural architecture for federated learning*. 2021. arXiv: 2112.13939 [cs.LG].

[16] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan, R. Chou, J. Glanville, J. M. Grimshaw, A. Hróbjartsson, M. M. Lalu, T. Li, E. W. Loder, E. Mayo-Wilson, S. McDonald, L. A. McGuinness, L. A. Stewart, J. Thomas, A. C. Tricco, V. A. Welch, P. Whiting, and D. Moher. "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews." In: *BMJ* 372 (2021). DOI: 10.1136/bmj.n71. eprint: https://www.bmj.com/content/372/bmj.n71.full.pdf.

[17] A. Z. Tan, H. Yu, L. Cui, and Q. Yang. "Towards personalized federated learning." In: *IEEE Transactions on Neural Networks and Learning Systems* 34.12 (2023), pp. 9587–9603. DOI: 10.1109/TNNLS.2022.3160699.

[18] X. Wei, G. Chen, C. Yang, H. Zhao, C. Wang, and H. Yue. "EFNAS: Efficient federated neural architecture search across AIoT devices." In: *2024 international joint conference on neural networks (IJCNN)*. 2024, pp. 1–8. DOI: 10.1109/IJCNN60899.2024.10650653.

[19] C. White, M. Safari, R. Sukthanker, B. Ru, T. Elsken, A. Zela, D. Dey, and F. Hutter. *Neural architecture search: Insights from 1000 papers*. 2023. arXiv: 2301.08727 [cs.LG].

[20] H. Zhu and Y. Jin. *Real-time federated evolutionary neural architecture search*. 2020. arXiv: 2003.02793 [cs.LG].

[21] H. Zhu, H. Zhang, and Y. Jin. "From federated learning to federated neural architecture search: a survey." In: *Complex and Intelligent Systems* 7.2 (Apr. 2021), pp. 639–657. ISSN: 2198-6053. DOI: 10.1007/s40747-020-00247-z.

[22] B. Zoph and Q. V. Le. *Neural architecture search with reinforcement learning*. 2017. arXiv: 1611.01578 [cs.LG].