René Karsten Schwermer

Federated Computing Systems In The Context of Energy Informatics

Technische
Universität
München

Technische Universität München

TUM School of Computation, Information and Technology

Federated Computing Systems In The Context of Energy Informatics

René Karsten Schwermer

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technische Universität München zur Erlangung eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Viktor Leis

Prüfer der Dissertation:

1. Prof. Dr. Hans-Arno Jacobsen
2. Prof. Dr. Hans-Joachim Bungartz
2. Prof. Dr. Frank Eliassen

Die Dissertation wurde am 20.12.2023 bei der Technische Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 23.05.2024 angenommen.

Abstract

Federated computing (FC) is an emerging privacy-preserving computing model. It consists of federated analytics and federated learning. Besides privacy enhancement, FC also helps reduce network traffic compared to a centralized approach because the raw data stays locally on each remote device. To mimic realistic conditions, we built a testbed of physically distributed devices to investigate the resource requirements and scalability challenges of FC systems. More devices participating in the process increase the systems' total energy consumption. This increase is especially the case when combining FC with methods such as differential privacy to boost privacy protection further. This work highlights the challenges of applying FC concepts and techniques to energy informatics. We conduct FC experiments on a high-frequency dataset capturing electrical signals in an office environment, prediction of coolant temperature in battery electric vehicles to protect semiconductor components from damage, and quantify trade-offs between energy consumption and privacy in experiments conducted on our edge device testbed consisting of 60 devices. Additionally, we offer a thorough analysis of the state-of-the-art of FC systems.

This thesis aims to improve FC use cases in the context of energy informatics, quantifying an FC system's energy consumption and providing a taxonomy for FC systems to identify trends and research gaps. Our findings highlight the potential of privacy-preserving machine learning (ML) for building management systems in an office environment. However, working with unevenly distributed labels on remote devices increases complexity, and three out of four ML architectures yield worse results when compared to a centralized approach. We obtain similar results for a dataset coming from battery electric vehicles. Additionally, we quantified the energy consumption of different FC systems. Their energy demand increases linearly with the number of clients and quickly out scales the energy consumption of a centralized ML pipeline when neglecting network traffic. All our findings emphasize trade-offs between privacy and resource consumption. An increase in privacy by leveraging FC and other privacy-enhancing, such as differential privacy, comes at the cost of higher system complexity, resource allocations, and energy consumption.

Zusammenfassung

Federated Computing (FC) nutzt Daten auf verteilten Systemen ohne direkt auf diese zuzugreifen. Es besteht aus Federated Analytics und Federated Learning. Neben der Verbesserung des Datenschutzes hilft es auch dabei den Netzwerkverkehr im Vergleich zu einem zentralisierten Ansatz zu reduzieren. Die meisten FC-Experimente laufen auf einem Gerät mit simuliertem Server und Clients. Wir haben eine Testumgebung mit physisch verteilten Geräten aufgebaut, um die Ressourcenanforderungen und Skalierbarkeit von FC-Systemen zu untersuchen. Je mehr Geräte an dem Prozess beteiligt sind, desto höher ist der Gesamtenergieverbrauch des Systems. Dies ist insbesondere dann der Fall, wenn FC mit Methoden wie Differential Privacy kombiniert wird, um den Schutz der Privatsphäre weiter zu erhöhen. In dieser Arbeit zeigen wir die Herausforderungen von FC im Kontext der Energieinformatik. Dazu gehören FC-Experimente mit einem hochfrequenten Datensatz, der elektrische Signale in einer Büroumgebung erfasst, die Vorhersage der Kühlmitteltemperatur in batteriebetriebenen Elektrofahrzeugen, um Halbleiterteile vor Beschädigung zu schützen, und die Quantifizierung von Kompromissen zwischen Energieverbrauch und Datenschutz in Experimenten, die in unserem Edge-Device-Testbed, bestehend aus 60 Geräten, durchgeführt wurden. Darüber hinaus erstellen wir in einem Survey eine Taxonomie für FC-Systeme, um Trends und Forschungslücken zu identifizieren.

Unsere Ergebnisse zeigen das Potenzial von datenschutzfreundlichem maschinellem Lernen (ML) für Gebäudemanagementsysteme in einer Büroumgebung. Allerdings erhöht die Arbeit mit ungleichmäßig verteilten Labeln die Komplexität, und drei von vier ML-Architekturen liefern im Vergleich zu einem zentralisierten Ansatz schlechtere Ergebnisse. Ähnliche Ergebnisse erhalten wir für einen Datensatz über batteriebetriebene Elektroautos. Darüber hinaus haben wir den Energieverbrauch von verschiedenen FC-Systemen quantifiziert. Ihr Energiebedarf steigt linear mit der Anzahl der Clients und übertrifft schnell den Energieverbrauch einer zentralisierten ML-Pipeline, wenn man den Netzwerkverkehr vernachlässigt. Alle unsere Ergebnisse zeigen, dass es einen Kompromiss zwischen Datenschutz und Ressourcenverbrauch gibt. Eine Erhöhung der Privatsphäre durch den Einsatz von FC und anderen Methoden zur Verbesserung der Privatsphäre, wie z. B. Differential Privacy, geht mit einer höheren Systemkomplexität, Ressourcenverbrauch und Energieverbrauch einher.

Acknowledgments

I would like to express my deepest gratitude and appreciation to the individuals who have played an instrumental role in the completion of this doctoral thesis. Their unwavering support, guidance, and encouragement have been invaluable to me throughout this journey.

I extend my heartfelt thanks to my esteemed professors, Prof. Jacobsen and Prof. Mayer, whose expertise, mentor ship, and insightful feedback have shaped the course of my research. Your dedication to academic excellence and your willingness to invest time in my intellectual growth have been truly inspiring.

To my circle of friends and co-workers, thank you for being a source of camaraderie, support, and motivation. Your stimulating discussions, shared insights, and willingness to lend a helping hand have made this journey not only academically enriching but also personally fulfilling.

I am profoundly grateful to my partner, Vera Denzer, for her unwavering belief in me and her endless patience throughout this endeavor. Your love, encouragement, and understanding during the long hours of research and writing have been my constant inspiration. Your presence in my life has brought balance and joy, making this accomplishment all the more meaningful.

Lastly, I extend my appreciation to all those whose contributions might not be explicitly mentioned but have nonetheless played a role in shaping my academic and personal growth.

This thesis stands as a testament to the collective effort of these remarkable individuals. Without your support, wisdom, and belief in my potential, this achievement would not have been possible.

Contents

1

Introduction

Businesses and public institutions are motivated to enhance their productivity, with potential boosts manifesting in improved and expedited service offerings or the more cost-effective manufacturing of goods. Innovation stands out as a key means to achieve this goal. Every technology undergoes an innovation cycle; initially, the trade-offs between research and development aimed at enhancing these technologies for generated benefits are significant, but they gradually reach a plateau. Therefore, to heighten societal productivity, it becomes crucial to fine-tune the performance of existing technologies and introduce or develop new ones. A data-driven approach empowers stakeholders across all fields to accomplish this. Harnessing data to refine design and decision-making processes enhances the efficiency of existing methods or facilitates the creation of new business models.

Generating, gathering, and processing data is necessary to fully utilize the potential of data-driven business models. Many such business models depend on distributed Internet-of-Things (IoT) devices, which monitor machines or offer connectivity (e.g., mobile phones). Recently, the number of non-IoT devices has stayed more or less constant at around 10.3 billion devices [1]. On the other hand, the proliferation of IoT devices is projected to grow from 0.8 billion in 2010 to 30.9 billion in 2025 [1]. These remote and distributed devices generate data and send it to a central server. This process stresses central processing entities and the network due to a permanent increase in data volume and complexity in pre-processing and evaluation.

Additionally, privacy concerns of policy-makers, customers, and business partners introduce another level of complexity. Complying with legal constraints in such a fast-paced and distributed environment is challenging. Therefore, this work focuses on leveraging distributed data sources in a privacy-preserving fashion and decreasing its network and energy footprint to reduce the increasing greenhouse gas (GHG) emissions of Information and Communication Technologies (ICT). The increase in energy demand for data centers rose by about 3 % from 2014 to 2020. Data centers consume about 200 TWh of electricity or constitute about 0.8 % of global electricity demand. [2, 3]

This work addresses this fundamental shift towards increasingly data-driven business models. We develop and benchmark privacy-preserving algorithms in public utilities and mobility applications. In particular, this work focuses on deploying systems on physically distributed devices to capture the real-world behavior of our algorithms concerning their energy footprint and hardware utilization.

1.1 Motivation

Artificial intelligence and data-driven business models are reliant on data for their functioning. Prominent data sources encompass machines (e.g., robots, engines, medical equipment), IoT devices, and wearable or mobile phones. As of 2023, the proliferation of IoT devices has reached approximately 20 billion, while global mobile phone numbers are anticipated to rise from 14.02 billion in 2020 to 18.22 billion in 2025 [4]. These devices actively monitor machinery, capture user interactions, and track service usage, continuously collecting data on temperature, pressure, humidity, vibrations, current, and various other physical properties. The escalating availability of sensors, log files, images, and text contributes to an increase in infrastructure overhead for handling and maintaining larger datasets. The datasets for training language and computer vision models exhibit a linear increase over time, expanding from approximately 100 data points in 1988 to a staggering 10 trillion in 2022 [5]. The transfer of all available data to a central server or data center places significant stress on network resources. Figure 1.1.1 provides

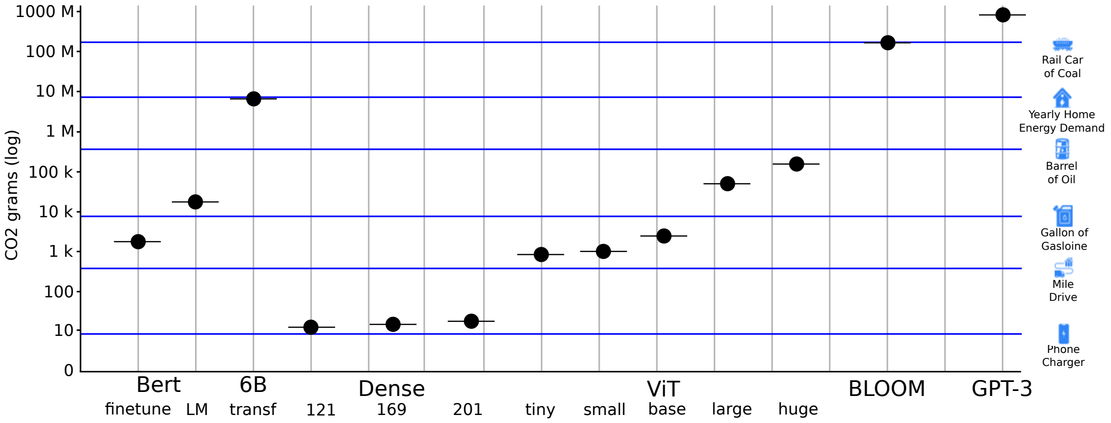


Figure 1.1.1: Average emissions for 13 natural language processing models on a logarithmic scale. [6, 7, 8]

a comprehensive overview of estimated GHG emissions associated with training large language models on diverse datasets, contextualizing them with everyday activities. This visualization underscores the burgeoning energy demand attributed to machine learning (ML).

In addition to considerations of data generation, transportation, and evaluation, it is imperative to comprehensively quantify the energy footprint encompassing the entire data life cycle. The energy consumption induced by network activities has risen significantly from 220 TWh in 2015 to 340 TWh in 2021 [9]. While video streaming constitutes a predominant share of internet traffic, mitigating the ecological footprint involves strategies such as reducing the transmission of images, text, or time series data over the network. ICT span compute and peripheral equipment, encompassing local area networks, telecommunication devices, networks, and data centers. Its contribution to global GHG emissions and the world's electrical energy consumption stood at 2 % (2009) [10] and 4.7 % (2012) [11], respectively. The handling and processing of data are becoming increasingly intricate due to the sheer volume of data and the prevailing trend toward distributed data sources. The energy expended in moving data from its point of origin to a database or processing unit is nontrivial. Simultaneously, computational capabilities within IoT devices are on a consistent upward trajectory. For instance, the first Raspberry Pi model 1b boasted a CPU frequency of 700 MHz, while the latest model 400 features a CPU frequency of 1.8 GHz and 8 times more memory [12]. The iPhone series has witnessed a notable surge in CPU clock speed, progressing from 612 MHz (iPhone 1, 2007) to 3.46 GHz (iPhone 14 Pro, 2022). Consequently, from both an economic and latency perspective, it becomes increasingly viable to leverage local resources rather than transmitting data to more centralized and remote data processing facilities, such as data centers.

Various entities can undertake the tasks of data generation and evaluation. A business, for instance, may choose to sell its data to a broker or enlist the expertise of an external data scientist to derive insights. A prevalent strategy involves the execution of a non-disclosure agreement among all participating entities or the anonymization of data before its dissemination to external parties. Nonetheless, the efficacy of such measures is context-dependent. Certain scenarios necessitate a level of trust between stakeholders, and there exists a potential risk of de-anonymization, especially when combining an anonymous dataset with publicly available ones. For instance, Netflix movie ratings, even when anonymized, could be partially de-anonymized by cross-referencing rankings and timestamps with publicly accessible information in the Internet Movie Database [13]. Other instances involve the exploitation of anonymized internet usage patterns [14] or location data [15, 16] to infer sensitive information about individuals.

Notable examples of legislation safeguarding sensitive information on a national level include the General Data Protection Regulation (GDPR) in the European Union [17] and the Personal Data Protection (Amendment) Act in Singapore [18]. In the United States, various state-specific data protection laws, such as the Californian Consumer Privacy Act [19], Colorado Privacy Act [20], Connecticut Data Privacy Act [21], and Virginia Consumer Data Protection Act [22], exemplify regional efforts to regulate the handling of personal data. While the degree of protection and the definition of sensitivity may vary across these frameworks, they collectively share the goal of curtailing the uncontrolled aggregation of data within a few entities.

Privacy concerns and regulatory constraints make it more challenging to deploy centralized pipelines due to legal risks, compliance efforts [23, 24], and a higher consumer sensitivity. Governments worldwide have implemented regulatory constraints of varying intricacy to address these challenges. Privacy-preserving techniques, such as Federated Computing (FC), can help to tackle privacy concerns and simultaneously decrease network traffic by shifting computational workloads to the devices that generate the data in the first place. With FC, data scientists and other stakeholders try to unravel the contradiction

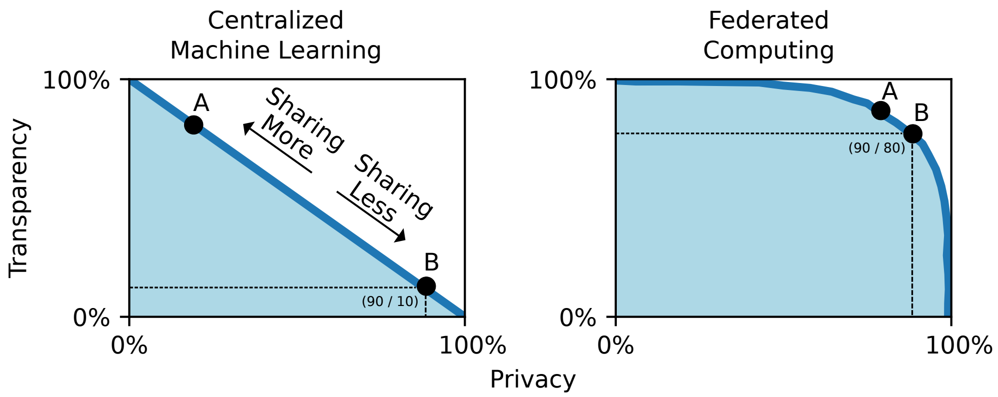


Figure 1.1.2: Two Pareto trade-offs between data transparency and privacy. Everything on the curve represents a desired outcome. The level of privacy for point B is in both cases the same (90 %), but the level of transparency increased in the right trade-off due to privacy-enhancing techniques from 10 % to 80 %. [25]

between using distributed, privacy-sensitive data, adhering to regulatory frameworks, achieving consumer needs, and reducing the ICT energy footprint. Figure 1.1.2 illustrates the abstract idea of FC. It shows the trade-offs between privacy and transparency. Those two metrics currently contradict each other. Increasing privacy subsequently reduces transparency or the amount of generated insights. With privacy-preserving techniques such as FC, it is possible to keep privacy levels high without interfering with transparency.

When distributing a copy of a dataset, the risk of losing control over it gives rise to a critical concern referred to as the copy problem, situated within the domain of input privacy. There is a risk of an uncontrolled creation of data copies when giving somebody a copy of your data. Potential solutions include FC, encryption, legal frameworks, and secure-multi-party computation. Employing data encryption ensures that access is restricted to individuals possessing the corresponding key, thereby preventing the uncontrolled dissemination of information. Legal frameworks play a crucial role in defining data usage practices; however, enforcing these constraints poses significant challenges.

To further improve the privacy level of FC systems, it is possible to combine them with other privacy-enhancing techniques from neighboring domains. The bundling problem arises from potential information leakages that unintentionally reveal more information than necessary due to the interconnectedness of data sources. For instance, checking the age on an ID card also makes it apparent where that person lives. Therefore, it is possible to get additional information, which is unnecessary to solve a given task, or to do backward inference to the input data based on the output. Differential Privacy (DP) stands out as a widely adopted technique to address these concerns. This approach artificially adds noise to a dataset without changing its statistical properties. This algorithm falls into the output privacy domain and mitigates the bundling problems.

FC is not domain-specific and it is applicable in a wide range of use cases. This research, however, concentrates on applications within the energy domain, encompassing areas such as building, public utilities, and mobility. A broader term to encapsulate these applications is energy informatics. It is a multidisciplinary field that leverages methods from computer science, data analysis, energy technology, and energy economics to address challenges in the design of energy systems. The overarching objectives of energy informatics are centered on sustainability, affordability, and security in the operation and optimization of energy systems.

Energy informatics encompasses a wide array of applications, spanning energy generation, transportation, storage, efficiency, and system optimization. Notably, the scope of energy is not confined to electricity; it extends to various forms of energy, including thermal or chemical energy. As both energy and IT systems undergo a transition towards increased decentralization (with energy systems incorporating decentralized renewable resources and IT systems evolving towards IoT scenarios), they share analogous characteristics in terms of resource management and allocation.

1.2 Problem Statement

Working with data on distributed systems introduces challenges related to managing devices, communication, and aggregating results. In such a system, each device may observe different aspects, even when measuring the same metrics. Consider the example of measuring vehicle velocity to identify driving patterns: one vehicle predominantly operates in a city, while another primarily travels on highways. This discrepancy results in biased datasets per client. Within FC, each device processes its data, and the number of devices in an FC system varies widely, ranging from a few clients to multiple thousands. Eventually, a server aggregates updates from all clients. However, each client's data can exhibit a significant bias toward a specific label or event, creating a non-independent and non-identically distributed (non-IID) scenario.

Merging client-specific models into a central model increases complexity due to potential biases in the models. Combining biased models often leads to a less effective model compared to a centralized approach. To address a decrease in model accuracy, two strategies are commonly employed: selecting clients with similar label distributions or adjusting the aggregation strategy on the server side. The subsequent sections delve into the intricacies of accuracy issues in models trained on energy data and how the distributed training of models impacts the overall energy consumption of the system.

1.2.1 Accuracy of Federated Computing

Currently, the majority of ML use cases deploy a centralized architecture. All clients send their data to a central server, where data scientists pre-process and evaluate them. Consequently, this architecture serves as the baseline for all published experiments. We are among the first to investigate the impact of FA and FL on model accuracy for use cases from the energy and mobility domain on physically distributed hardware. The latter captures battery electric vehicles.

In general, model accuracy-related issues emerge from either an uneven distribution of labels on the client side or the server aggregating the clients' updates poorly. Other potential causes for a decrease in model accuracy are FC systems enhancements. Those are privacy-preserving techniques, such as DP or compression. However, in this work, we limit the experiments to investigate the effect of non-IID data on model accuracy mainly with pure FC systems, which consist of the minimum number of components, such as client selection, aggregation, and communication. The experiments capturing an FC system's energy consumption also incorporate extensions such as DP, which is computationally heavy.

FC use cases frequently encounter non-IID challenges. In our experiments, we draw on datasets from the building and mobility sector, both characterized by highly biased clients. The first dataset captures electrical signatures from various appliances and devices within an office environment. Offices consist of diverse devices, with personal computers, lights, and USB chargers being prevalent. Meeting rooms, on the other hand, may have additional equipment such as projectors or a designated space for printing. Training an FL model in this office environment at the room level introduces highly biased models, and merging them into a single model poses significant challenges. The dataset is inherently non-IID.

The second dataset records multiple temperature metrics (e.g., ambient, inverter, and oil), battery metrics (e.g., state-of-charge, current, and voltage of the high voltage system), and movement-related metrics (torque, velocity) of battery electric vehicles. Each customer operates a vehicle differently, resulting in a non-IID label distribution due to varied driving styles and environments. Some vehicles may exclusively operate in regions with constant ambient temperatures, while others traverse a broader range of temperature conditions.

1.2.2 Energy Demand for Federated Computing

Privacy-preserving ML on distributed systems poses challenges concerning network traffic and energy consumption. Such systems can have a heterogeneous pool of devices, which might not be optimal for ML-specific tasks. Centralized servers in data centers have an optimized cooling strategy and customized hardware for heavy workloads. Additionally, working with multiple devices instead of one increases the total management overhead. Each device runs an operating systems, maintains its memory, and uses CPU resources for other background tasks. Therefore, an FC system might consume more energy for specific tasks than a central device. However, depending on the data set size, FC requires less network traffic, reducing its associated energy consumption. It is challenging to quantify network-related energy consumption due to its diverse architectures. The number of hops a package takes to reach its final destination or the type of network transmission (e.g., cable or mobile network with 2G, 3G, 4G, or 5G) affect data transfer's energy consumption. For example, 2G, 3G, and 4G mobile networks require about 15 kWh/GB, 1 kWh/GB, and 0.7 kWh/GB, respectively, whereas a fixed cable connection requires about 0.08 kWh/GB [2].

In our work, we neglect this energy footprint. Nowadays, new technologies should quantify their energy footprint to avoid any long-term adverse effects. A higher energy demand is neglectable when deploying and testing in a lab environment. When a new approach finds its way into applications and its usage scales up, adjusting it to be more energy efficient might be challenging. Therefore, it is paramount to know the energy footprint of FC systems as early as possible to improve the FC framework's underlying architecture.

Knowing the energy footprint of services throughout the value chain of a product becomes crucial to comply with frameworks such as the Environmental, Societal, and Governmental framework [26]. It captures the impact of businesses on those three areas. It helps customers and investors assess the level of responsibilities a company is taking, and increases transparency. GHG emission is one part of this framework, consisting of scope 1 to 3 emissions. Knowing how much a deployed FC system consumes helps to comply with such a framework. The following list describes Scope 1 to 3 emissions and provides an example. FC systems contribute to Scope 1 (server) and Scope 3 (clients) emissions.

- Scope 1 emissions are GHG emissions released directly from a business (e.g., on-premise servers consuming electricity).
- Scope 2 emissions are indirect GHG emissions released from the energy purchased by an organization (e.g., power plant generating electricity for the server).
- Scope 3 emissions are indirect GHG emissions, accounting for upstream and downstream emissions of a product or service, and emissions across a business's value chain (e.g., resources required to build the server).

1.3 Approach

FC requires a server and client architecture. It is possible to emulate such an environment on one device. This approach is feasible when optimizing model accuracy. However, there needs to be more attention on potential bottlenecks concerning network or throughput as well. Deploying an entire FC system on one machine lets the server and clients communicate with each other via internal memory, which is faster than going over the network. Such a mismatch between the execution speed on one device and the network traffic between multiple devices is especially the case when using wireless connections. To simulate more real-world scenarios, we built an IoT testbed. Given the growing prominence of IoT devices, we identify suitable IoT devices, which are experiencing an increasing market share, to establish a robust testbed. We constructed an initial testbed comprising 48 Raspberry Pis and an expansion using 12 Jetson Nanos and 10 Orins. A power-over-Ethernet (PoE) switch powers all devices in the testbed. This architecture provides continuous power readings. We additionally emulate different networks settings on those devices with *netem*. This tool allows to artificially change network throughput and package losses.

The following sections describe in more detail how we tackle the problem of emulated FC systems, how to cope with non-IID scenarios in the energy and mobility domain, and how we measure the electricity consumption of FC systems.

1.3.1 Accuracy of Federated Computing

To improve the accuracy of FC models, we first implement a centralized reference that achieves a desired performance metric. Second, we deploy multiple FC use cases with different label distributions to identify the impact of aggregation algorithms on the models' performance. Those use cases are twofold. The first one assumes evenly distributed labels over all participating clients. This results in similar models per client if the training runs for enough rounds. The second FC use case runs with the actual distribution of the labels without artificially altering it.

The initial step involves deploying a Federated Learning (FL) use case within the Non-Intrusive Load Monitoring (NILM) domain, aimed at acquainting oneself with its inherent characteristics and distinctive features. An essential part of this endeavor is the identification of challenges inherent to FL, including instances where experiments primarily run on a single device, which mimics an FL system. While this approach aims to enhance ML performance, it introduces complexities in comprehending network and energy consumption patterns, particularly when the system scales up with a more significant number of clients. These challenges extend to client selection strategies, ML optimization, and aggregation strategies.

1.3.2 Energy Demand for Federated Computing

In general, there are two approaches to measuring the electricity consumption of a piece of hardware. Those are either hardware- or software-based. Figure 1.3.1 provides an overview of their respective subcategories.

System monitoring strategies leverage external hardware such as power meters (measuring the power supply unit) to measure an entire device's energy consumption. This approach needs to scale better due to the manual labor required to install those power meters. Load disaggregation also leverages an external power meter. However, it tries to identify unique patterns in the energy consumption profile that belong to a given task or component (e.g., CPU and memory). For instance, instead of measuring all appliances in a household individually, it is also possible to measure the power drawn at the main and then disaggregate those measurements to an appliance level. Such an approach works well for households or other use cases with distinct patterns. There are approaches to applying this strategy to obtain energy readings of specific software artifacts. However, they yield inaccurate results, so this strategy is not widely adopted. Another approach to measuring the energy consumption of entire systems or specific software artifacts is software-driven. The software runs in the background and tries to estimate energy consumption instead of installing physical power meters. This strategy's disadvantage is lower accuracies compared to its physical counterpart. The error can be up to 40 % on a

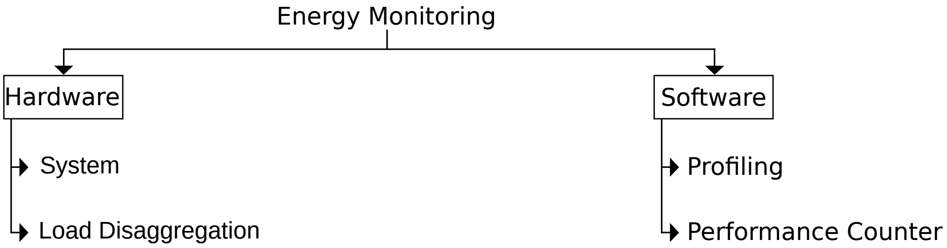


Figure 1.3.1: Strategies to measure the energy consumption of software.

device level [27], whereas software-based power meters achieve an accuracy of up to 95 % on a CPU or GPU level [28]. Software-based power meters indirectly measure energy consumption via profiling, performance counters or by using on device hardware, such as Running Average Power Limit (RAPL) for x86 architectures [29] or NVIDIA Management Library physical (NVML) for GPUs [30]. Different profiling approaches exist to estimate a device's energy consumption [29]:

- Software-based prediction models leveraging for example CPU or GPU cycles in combination or without readings from RAPL or NVML sensors,
- Device datasheet: Assuming full hardware utilization and getting the respective maximum power drawn from the data sheet

Raspberry Pis have an ARM architecture, which currently does not support sensors such as RAPL, and load disaggregation is not feasible. Therefore, we use external power meters measuring the entire device's energy consumption. Even though it is impossible to identify the energy consumption of specific components (e.g., CPU, memory, network) or processes with that approach. To get closer to the FC-induced energy consumption, we first measure the energy consumption of an IoT device in idle mode and then subtract it from the measurements during the experiments.

A crucial aspect of the research involves quantifying the combined impact of energy and network consumption and assessing the scalability of FL both with and without DP mechanisms. The insights garnered from this comprehensive study will be harnessed to refine the FL use case, particularly its applicability in an automotive scenario. Additionally, we aim to consolidate the findings of a literature research into a taxonomy describing the fundamentals of FC systems. This summary will encompass the core building blocks of such systems while addressing potential extensions and enhancements.

1.4 Contributions

The main contributions of our work about FC in the context of energy informatics are:

i. We investigate the effect of training data distributions and variations of model training (early stopping, learning rate scheduler and aggregation strategy) on model metrics. The dataset contains power readings (voltage and current), which come from individual offices. Our results show that it is possible to train ML models on non-IID data while achieving a model performance that can compete with a centrally trained model.

ii. We investigate the impact of two FC approaches (FA and FL) on model transferability and hardware metrics for multiple vehicles with different engine types. The computational power in vehicles is increasing, but with it the number of services competing for them. We identified trade-offs between privacy, energy consumption, and hardware metrics (CPU utilization, memory, storage, and network) by training models of different complexities (linear regression and ML) on an IoT testbed to emulate a vehicles' available computational resources.

iii. We quantify the impact of different emulated networks (5 Mbit/s, 8 Mbit/s and 4G) and privacy-enhancing techniques like DP on the energy consumption of an FL system. We address deviations between a centralized ML approach and FL to get a better understanding of potential trade-offs between energy consumption and privacy. All experiments run from two clients with up to 47 clients, which shows scalability issues with respect to an increasing energy consumption of the entire systems due to an increase in client overhead and total training time, when using the FL framework *Flower*.

iv. We evaluate the network on physically distributed edge devices and virtual machines (VM) in the context of FC to highlight differences. Building and maintaining an IoT testbed is time consuming. Therefore, we investigate if the network traffic of an FL system deployed on virtual machines with respect to network traffic (package size and number) is comparable to the measurements coming from an IoT testbed. Using the Ethernet interface to measure network traffic between VMs gives a good estimate of the expected network bandwidth consumption, but underpredicts the number of messages by 94 %. Using the local loopback interface to measure network traffic on one machine hosting multiple simulated clients does not give a good estimate for the actual network traffic.

Parts of the content and contributions of this work have been published in:

- R. Schwermer, J. Buchberger, R. Mayer, and H.-A. Jacobsen. “Federated Office Plug-Load Identification for Building Management Systems.” In: e-Energy ’22. Virtual Event: Association for Computing Machinery, 2022, pp. 114–126. ISBN: 9781450393973. DOI: 10.1145/3538637.3538845
- R. Schwermer, R. Mayer, and H.-A. Jacobsen. “Energy vs Privacy: Estimating the Ecological Impact of Federated Learning.” In: *Proceedings of the 14th ACM International Conference on Future Energy Systems.* e-Energy ’23. Orlando, FL, USA: Association for Computing Machinery, 2023, pp. 347–352. DOI: 10.1145/3575813.3597344
- R. Schwermer, E.-A. Bicer, P. Schirmer, R. Mayer, and H.-A. Jacobsen. “Federated Computing in Electric Vehicles to Predict Coolant Temperature.” In: *Proceedings of the 24th International Middleware Conference Industrial Track.* Middleware Industrial Track ’23. Bologna, Italy: Association for Computing Machinery, 2023, pp. 8–14. ISBN: 9798400704277. DOI: 10.1145/3626562.3626829. URL: https://doi.org/10.1145/3626562.3626829

1.5 Organization

The rest of the document is organized as follows. Chapter 2 presents our methodology for building and deploying FC systems. It describes how FC works and highlights the importance of knowing how much energy a distributed system consumes compared to a centralized approach. We summarize the key achievements of each publication and highlight the author’s contributions in Chapter 3. Chapter 4 discusses the results by comparing our findings with the literature. Chapter 5 presents the conclusion and an outlook for future work. Finally, Appendix A, B, and C show our published papers.

2

Methodology

This chapter provides an overview of the relevant background information and presents our methodology for developing FC systems in the context of energy informatics. Section 2.1 describes all components of FC systems and how they work together. Additionally, it provides information about the need to compare the energy footprint of FC systems to currently adopted centralized approaches. Section 2.2 presents our methodology for building and monitoring FC systems. It describes our testing infrastructure used for our experiments.

All published work deals either with describing FC frameworks and quantifying their energy footprint or how to apply them to the energy and mobility domain. The following list gives a first overview of all publications and works:

- Paper I (see 3.1): FL on a dataset capturing electrical signals (current and voltage) of devices in an office environment,
- Paper II (see 3.2): Quantifying the energy footprint of FL systems with its different aggregation strategies and privacy-enhancing techniques, such as DP,
- Paper III (see 3.3): FL and FA deployed on simulated battery electrical vehicles to predict its average coolant temperature while also quantifying its hardware and energy utilization.

2.1 Background

This thesis focuses on FC in the context of energy and mobility. Mechanical, electrical, and process engineers continuously improve the performance and cost-efficiency of components. However, some technologies (e.g., wind turbines and electric motors) reach a state where such incremental improvements plateau. Therefore, more and more engineers are trying to optimize the operations of such components. Those optimization efforts require data to build value-added models, which can be private or company-sensitive. FC builds the basis for such business models. Section 2.1.1 explains FC in more detail, and Section 2.1.2 provides information about the broader context of the connections between ICT, energy, and policies.

2.1.1 Federated Computing Systems

FC belongs to the domain of privacy-preserving computations. Its goal is to extract information from distributed data sources without compromising the raw data. In the process, no client sends private data to a central location. Instead, the server sends computing tasks to each participating client, which executes them and then only sends the respective update to the requesting server. An FC round consists of the following steps:

1. Server selects participating clients,
2. Server sends computation task to all clients,
3. Clients execute the computation task and send their individual update to the server,
4. Server aggregates all clients' results and distributes the aggregated result back to each client.

FC includes FL and FA. Bonawitz et al. introduced the concept of FL in 2016 with the next word prediction of Google's Android keyboard Gboard [34]. The difference between FL and FA is the type of executed computing tasks and the number of aggregation rounds. FL focuses on ML, mainly consisting of multiple aggregation rounds. The goal is to iteratively reduce a loss function and subsequently increase model performance. On the other hand, FA leverages statistical instructions, such as averages or sums, which are only executed once on each client and focus on concluding data [35]. Some example use cases for FA include model evaluation or debugging [36, 37]. FL consists of a combination of multiple FA steps [38]. Another way to divide FC is by system focus. It can focus on reasoning or learning, translating to FA and FL. Another term for the former is deductive systems, and for the latter, inductive systems [39]. Deductive systems are also called "Good old-fashioned AI" and typically rely on rule-based or logical agents [40, 41]. Conversely, inductive systems try to learn based on the input data and are less prone to changes in the observed environment. In this work, we do not cover federated databases. They also have a client-server architecture, which connects distributed databases to one another. The end user only sees one database, even though it consists of multiple ones.

The number of publications in the area of FL increased since 2016 steadily. Farooq et al. [42] and Lo et al. [43] present in their quantitative analysis existing FL research papers without considering FA. They highlight the recent increase in publications starting in 2017. The yearly documents increased from 25 in 2017 to 280 in 2020. Additionally, they cluster papers into different categories to distill focus areas. Most papers investigate the impact of training settings on ML model performance. The main reason to adopt FL is data privacy (62 % of papers), followed by communication efficiency (23 %). In general, their research emphasizes the increasing interest in FL. Multiple tutorial-like surveys exist as a reaction, which describe individual components of an FL pipeline together with some application examples.

Depending on the FC system architecture, either a central server aggregates all results (central) or all or some clients act as a server as well (hierarchical or peer-to-peer). Figure 2.1.1 gives an overview of three architectures. The first architecture in the figure shows a centralized FC approach. A server aggregates all results from the clients. The hierarchical architecture has an intermediate layer between clients and servers to increase redundancy. The devices in this intermediate layer act as servers and clients simultaneously. A fully decentralized FC system peerages in a peer-to-peer fashion without a central server. Another term for this architecture is galaxy FC [45, 46]. All those architectures work for FA and FL. All FC systems consist of three basic building

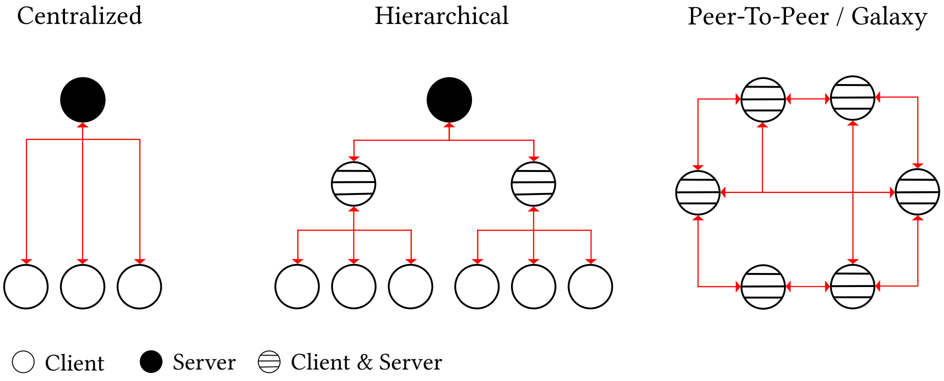


Figure 2.1.1: Three FC architectures (centralized, hierarchical and peer-to-peer) with different aggregation server locations. Illustration inspired by [44]. A node can be a client, a server or both, which is indicated by the circle filling.

modules:

- Client selection strategies decide if all or a subset of available clients will participate in a training round. The selected set of clients can change after each round.
- An aggregation algorithm merges all client updates into a central model. One of the first and simplest algorithms is FedAvg. It averages all client updates.
- A pre-defined communication and serialization protocol is necessary to enable communication between servers and clients. A widely used combination in FC frameworks is gRPC (protocol) with Protobuf (serialization) [34, 47, 48, 49, 50].

A challenge for FC systems is an uneven distribution of features and labels on the clients. The server has no data access on the client side. It only knows some meta information, such as image resolution or, for time series, the respective units of each column. In an ideal FC system, all clients' datasets have similar statistical attributes and yield identical results to the executed computation tasks. However, some clients' datasets might be biased towards specific labels. Therefore, the same execution task can deliver different results per client. Aggregating results based on non-IID data is challenging due to its impact on the final result on the server side.

In some cases, aggregating individual client updates can result in worse results than a traditional centralized ML approach. Choosing a subset of available clients or a suitable aggregation strategy can improve the generated insights. Another issue in this context is clients dropping out during an FC process. Such instances can also result in a non-IID scenario even though the initial client selection ensures an even distribution of labels, or the entire run gets delayed because the server is waiting for all clients to finish their execution tasks.

2.1.2 Federated Computing in Energy Informatics

FC is application-independent. However, its architecture captures well the shift from centralized energy generation to a more decentralized renewable energy system. Energy informatics looks at the ecological impact of ICT and how to improve energy systems as a whole, going from generation to distribution and consumption. Using new algorithms is paramount to avoid an increase in GHG emissions. Therefore, this work looks at the efficiency of one FL framework in combination with DP and how to apply FC to different applications.

The term energy productivity or energy intensity gives us a first indicator in which direction an economy is going. It describes the relationship between a country's gross domestic product and its energy consumption. The energy productivity of Germany is increasing since 1990 (see Figure 2.1.2). Since 2005 the final energy consumption is decreasing indicating a decoupling of economic growth and energy consumption [51, 52]. Financial, political or healthcare crises interrupt this upwards trend. A similar trend applies to other European Union member states, partly due to a shift from industry-related business to service-driven economies [52]. It is crucial to quantify the energy consumption of software and subsequently try to reduce it by keeping its performance constant or increasing it. There is the risk of over-compensating efficiency gains with additional usage. An explanation for those scenarios is the Jevons paradox or rebound effect. It occurs when technological progress or government policy increase the efficiency of a resource. Still, the falling cost of using those resources increases its demand, increasing, rather than reducing, resource use [53]. Jevons investigated the impact of more efficient coal usage, leading to broader adoption of this resource and higher energy consumption. There are multiple examples of such rebound effects. For example, combustion engines in cars have become more and more efficient, but a car's weight has increased, resulting

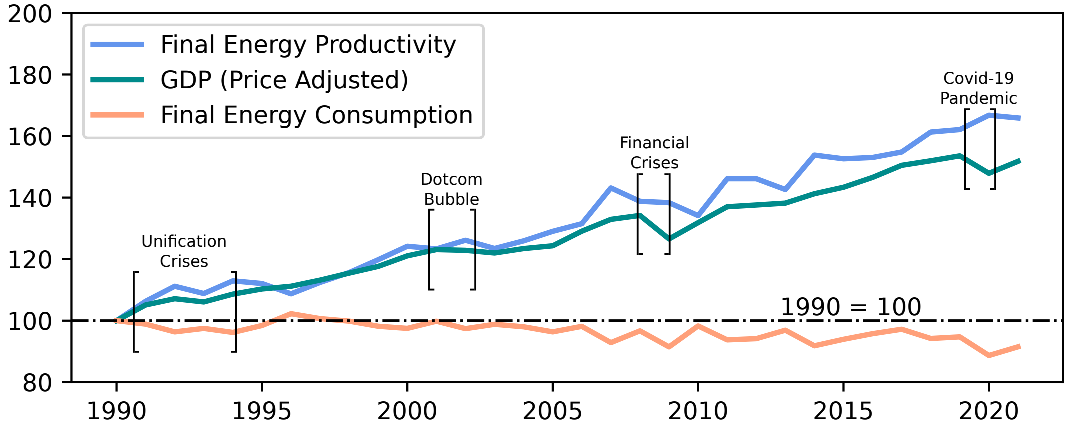


Figure 2.1.2: Energy productivity of Germany from 1990 till 2020 including time spans of financial crises responsible for a recession [56]. Data coming from the Federal Environment Agency of Germany. [51].

in constant or higher energy consumption per kilometer. A similar pattern applies to software development. Hardware resources become computationally more powerful and accessible. Therefore, it is feasible to compensate for inefficient software by using more hardware resources. Studies show that this effect impacts energy-efficiency policies, but to which extent depends on the specific use case [54, 55].

2.2 Implementation of Experiments

Our experiments run on our IoT testbed and VMs. All devices and VMs run on Ubuntu 20.04 LTS. The deployment of software and required files happens via Ansible. In addition to Ansible, we use Terraform to deploy multiple VMs simultaneously.

The testbed consists of multiple hardware components. Depending on the application, the cluster can be scaled up or down to match the overall costs to the available budget. 48 Raspberry Pi 4B (called modules) with a PoE HAT perform the computations. Half of the modules have 2 GB of memory, and the other half has 4 GB. The switch powers all modules via PoE. Figure 2.2.1 shows the data flow in the testbed and the connectivity between all components. The monitoring strategy comprises the Python package *psutil* (CPU, memory, disk storage), *Wireshark* (network), external powers (energy), and the

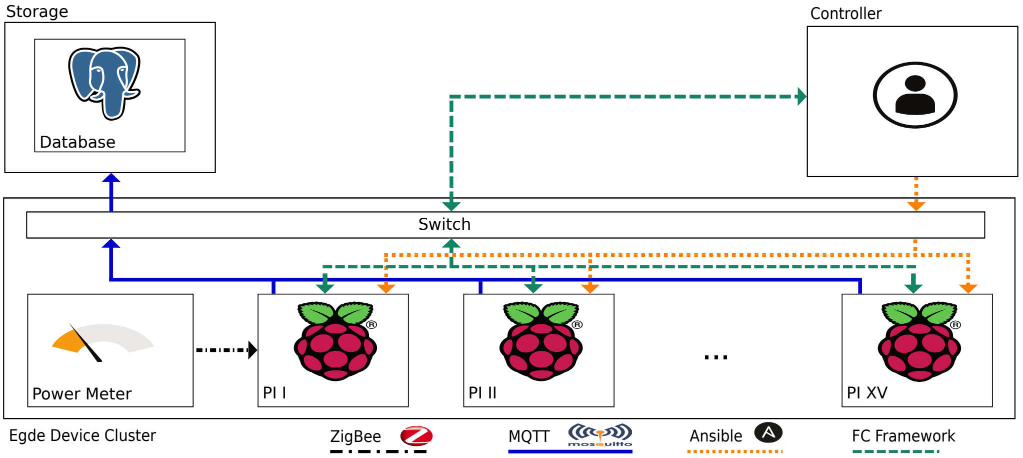


Figure 2.2.1: Our testbed used for the FC experiments. We built it from scratch including hardware and software (monitoring and deployment) setups.

PoE readings (energy) from the switch. A *PostgreSQL* database stores all measurements for later evaluations. The external power meters measure the electricity consumption of three Raspberry Pis and the PoE switch. Those measurements serve as a reference to validate the energy readings from the PoE switch.

3

Summary of Publications

This chapter summarizes all published papers for this publication-based dissertation individually. Overall, this dissertation is based on three accepted peer-reviewed publications. Additionally, there is a fourth paper currently under submission (see Appendix D). We highlight each publication's key idea, outline the achievements, and summarize the author's contributions.

Section 3.1 outlines our publication on FL in the context of NILM and building management systems. We leverage the publicly available building-level office environment dataset (BLOND). Section 3.2 provides findings of our experiments about the energy footprint of FL with and without DP as an extension. The experiments run on a testbed of 48 Raspberry Pis and we measure their electricity consumption with an external power meter. Section 3.3 compares an FL (LSTM) and an FA (linear regression) approach with each other by using real-world data from battery electric vehicles. Additionally, we capture their hardware and energy utilization to identify trade-offs between privacy, hardware and energy constraints.

3.1 Federated Office Plug-Load Identification For Building Management Systems

Reference: R. Schwermer, J. Buchberger, R. Mayer, and H.-A. Jacobsen. "Federated Office Plug-Load Identification for Building Management Systems." In: e-Energy '22. Virtual Event: Association for Computing Machinery, 2022, pp. 114–126. ISBN: 9781450393973. DOI: 10.1145/3538637.3538845

Full-text version enclosed: Appendix A

Summary:

Energy usage within buildings contributes to 40 % of the overall energy consumption in the European Union and the United States. In addition to thermal energy, buildings also rely on electricity. One consumer of electricity are office devices (e.g., monitor and projectors) and unregulated plug-in devices, like mobile phones and USB chargers. Especially the latter is steadily on the rise. In 2018, electricity consumption for European households accounted for 25 % of the total EUs energy footprint. FL offers a solution to harness this data to improve energy efficiency while complying with regulatory frameworks, such as the GDPR.

This paper uses a high-frequency energy dataset of office appliances (BLOND) to train four appliance classifiers (CNN, LSTM, ResNet, and DenseNet). We chose those ML architectures as a basis because they are widely adopted. Knowing which device is running in the offices helps to improve energy management systems. At the time of submission, we were the first one to leverage this dataset in an FL context. We investigate the effect of different data distributions (entire dataset, IID, and non-IID) and training methods on four performance metrics (accuracy, F1 score, precision, and recall). Our findings reveal that a non-IID setup leads to a decrease of up to 44 % in all performance metrics for specific model architectures. However, the LSTM model, trained with non-IID labels, can attain F1 scores similar to those achieved through central training.

Author's contributions: Conceived and developed the approach. Devised optimisations. Conducted analysis and experimental evaluation. Wrote the paper.

3.2 Energy vs Privacy: Estimating the Ecological Impact of Federated Learning

Reference: R. Schwermer, R. Mayer, and H.-A. Jacobsen. "Energy vs Privacy: Estimating the Ecological Impact of Federated Learning." In: *Proceedings of the 14th ACM International Conference on Future Energy Systems.* e-Energy '23. Orlando, FL, USA: Association for Computing Machinery, 2023, pp. 347–352. DOI: 10.1145/3575813.3597344

Full-text version enclosed: Appendix B

Summary:

More and more stakeholders are concerned about the ecological impact of ML and its associated network traffic. The current research in FL focuses on improving ML accuracy, and experiments run on VMs or on one machine with simulated clients. We quantified the network traffic and energy consumption of FL clients under different network constraints and privacy-enhancing techniques, such as DP.

At the time of submission, we were the first to build a testbed consisting of 48 Raspberry Pis, which we used to measure more real-world-like network and energy readings. We evaluated a convolutional neural network trained on the MNIST dataset under different network constraints, with DP and with an increasing amount of participating clients. We compared network and energy estimations with actual measurements.

We quantify the network traffic, energy consumption, and training time for each experiment. The results show the importance of experiments on physically separated nodes and the need to improve software-based power monitoring. The estimated energy consumption deviates up to 35 % from the measured ones. Also, the total energy consumption of an FL system scales linearly with its number of clients and with DP-enabled a clients energy consumption increases by up to 300 %.

Author's contributions: Conceived, developed, and implemented the approach. Devised optimisations. Conducted analysis and experimental evaluation. Wrote the paper.

3.3 Federated Computing in Electric Vehicles to Predict Coolant Temperature

Reference: R. Schwermer, E.-A. Bicer, P. Schirmer, R. Mayer, and H.-A. Jacobsen. "Federated Computing in Electric Vehicles to Predict Coolant Temperature." In: *Proceedings of the 24th International Middleware Conference Industrial Track*. Middleware Industrial Track '23. Bologna, Italy: Association for Computing Machinery, 2023, pp. 8–14. ISBN: 9798400704277. DOI: 10.1145/3626562.3626829. URL: https://doi.org/10.1145/3626562.3626829

Full-text version enclosed: Appendix C

Summary:

Reducing greenhouse gas emissions in mobility is paramount to achieving a carbon-neutral society. However, battery-electrical vehicles (BEV) introduce unique engineering challenges to protect expensive electrical components from overheating. A centralized architecture for model-driven predictions of coolant temperatures poses privacy and legal issues. Another challenge is the competition of resources between the on-board applications in a vehicle.

Therefore, we introduce FC to help transform the mobility sector. We evaluate the performance of two FC approaches (linear regression and ML) on hardware and privacy metrics by leveraging a real-world dataset from BEVs. It contains measurements from 35 BEVs from a Bavarian car manufacturer.

Our findings show trade-offs between hardware utilization and model accuracy. The linear regression model yields the best performance and prediction metrics. FC with ML shows up to 761 % variances when comparing vehicle-specific models with models trained with the entire fleet. Clustering the data into velocity profiles based the Worldwide Harmonised Light Vehicles Test Procedure framework partly improves prediction performance.

Author's contributions: Conceived, developed, and implemented the approach. Devised optimisations. Conducted analysis and experimental evaluation. Wrote the paper.

4

Discussion

This chapter discusses our results in the larger context of the applicability of FC in the energy domain and its effect on the environment. Specifically, we look at the building and mobility sector as two use cases for distributed entities. Both domains get smarter by incorporating more sensors and connectivity, which makes them suitable for data-driven applications and business models. Also, they contribute about 33 % to the EU energy consumption in the building and mobility sectors. The EU requires the reduction of its share of GHG emissions, and adding more functionalities might oppose those goals. We conclude the chapter by highlighting that deploying FC in environments with higher complexity and energy costs is possible. Additionally, we point out the energy consumption of FL systems and compare it with centralized approaches, which leads to trade-offs between privacy level and energy consumption.

A building can either have residential or commercial use. For both scenarios, it is possible to improve a wide range of specific tasks, which focus on the entire building as a system (e.g., load forecast), on the occupants (e.g., demand forecast), all appliances and devices in a building (e.g., NILM) or on a concrete machine (e.g., heating, ventilation or air conditioning). Other works used FL to investigate its usability in the building sector. Wang et al. combines FL and NILM [57, 58]. Other authors combine FL and electricity measurements to develop load forecasting models [59, 60] or demand forecasting for private households [61]. Some prior art investigates the impact of heating, ventilation, and air conditioning [62]) or multiple private households with smart meters on a local or global energy management system [63, 64]. All of those works focus on improving the prediction performance of their respective ML models. They especially capture the effect of unevenly distributed labels on the systems' ML accuracy metrics. However, they lack an actual implementation on multiple devices to track the hardware and network requirements of the FL system. We capture those challenges in the context of building management systems and deploy a use case by leveraging the BLOND dataset. To the best of our knowledge, we are the first to combine FL with a high-frequency measurement dataset capturing the energy consumption of office appliances. Our findings highlight the impact of non-IID data on the performance of four different ML architectures (CNN, LSTM, ResNet, and DenseNet). Even an IID of data results in an average F1 score reduction of 0.14 points or 15 percentage points. Strategies to decrease the difference between centralized and FL training include no stopping (the training does not stop when the validation loss does not decrease for five consecutive epochs.), a cosine adjustment of the learning rate over time and changing the number of local rounds before aggregating the updates on the server side. Only the LSTM achieves in a non-IID setup an F1 score close to the reference of the centralized approach.

FC aims to improve input privacy, solving the copy problem. A widely adopted approach to further increase privacy levels is the combination of FC with DP. DP artificially adds noise to a dataset without changing its statistical properties. However, DP is computationally expensive, and FL increases the total overhead of the system due to its distributed nature.

FL experiments mainly run on simulated systems with one device hosting the server and all clients. Table 4.0.1 provides an overview of multiple FL use cases and their respective experiment environment. Prior art quantified the required energy consumption of an FL system, achieving a given accuracy threshold. Their system consists of multiple GPUs [65, 66]. Additionally, they calculated the generated GHG emissions of the entire training process. However, their testbed is limited to a few devices, neglects network traffic, and does not incorporate additional privacy-enhancing techniques like DP. Therefore, we deploy FL experiments and centralized ML on an IoT testbed to quantify the entire system's energy footprint. Our findings highlight the difference between centralized ML and FL systems with and without DP on energy consumption. We also capture the behavior of the FL framework *Flower* on the scalability and training time. Training time

Table 4.0.1: Overview of environments used for FL experiments. The prominent approach is to emulate server and clients on a single device.

FL Environment / Hardware	References
Single device	[57, 58, 60, 67, 68, 69, 70, 71]
Virtual Machines	[59, 72]
Physically distributed devices	[34, 73]
Hybrid	-
Unknown	[61, 62, 63, 67, 74]

per round increases linearly with the number of participating clients for the Flower framework. The same applies to the servers' energy consumption regardless of the aggregation strategy. DP running on the clients increases the training time and energy consumed by 300 % and 280 %, respectively.

FC enables the generation of insights from distributed data sources without having to access them directly. However, labels on the client side can be non-IID, which introduces challenges concerning aggregating insights from multiple clients. Additionally, the energy consumption of FC with or without privacy-enhancing techniques becomes crucial for comparing FC and traditional approaches, besides its model generation performance. Additionally, it is challenging to create generally applicable strategies to know how high an FC system's network traffic and energy consumption will be in advance. Those metrics depend on the distribution of the system, the aggregation frequency, and the client update size. The participating clients consume the majority of energy, but the aggregation strategy on the server also affects the systems total energy consumption. The differences between aggregation strategies are minor. However, this becomes critical for peer-to-peer systems where the clients also run aggregation algorithms. An aggregation strategy with a minimal higher energy consumption could lead to an exponential increase in energy consumption for a peer-to-peer system.

As computational capabilities within vehicles continue to rise, the prospect of offloading computational tasks from the cloud to the edge becomes more feasible. The escalating volume of data generated within vehicles intensifies the competition for on-board hardware resources. Given the sensitivity of data from privately owned cars, there is a need for innovative solutions. Some existing approaches integrate FL with mobility applications to address this challenge. For distributed and non-stationary vehicles in the Internet of

Vehicles (IoV), issues such as connection loss and the heterogeneity of individual datasets pose challenges, as discussed by Ji et al. in their survey [75]. To tackle this, Li et al. and Ye et al. have developed a peer-to-peer architecture incorporating an aggregation strategy to accommodate vehicles dropping out during training rounds [46, 76]. These approaches exhibit comparable accuracy to centralized ML architectures. Various IoV applications include stress level identification using training on roadside units [77], image classification [78], and object detection [79]. Tan et al. offer an overview of other IoV use cases leveraging FL [80], for instance using charging data of electrical vehicles to better predict future energy demand. Beyond model performance, it is crucial to balance ML models' energy consumption with a battery's state of charge to ensure a positive customer experience [81, 82]. Notably, there is a gap in the literature regarding the utilization of time series data for developing a privacy-preserving coolant temperature prediction model.

To the best of our knowledge, our work is the first to leverage real-world measurements from BEVs in an FC environment. We optimize the model performance of a linear regression and an ML model and quantify the computational resources required for the training. Both aspects are crucial to deploying a service into production. Model training, which consumes too much memory, CPU, and disk storage capacities, interferes with the demand for other services, such as street sign detection or algorithms for autonomous driving. A high available driving distance for BEVs is a selling point, and any service running in a vehicle should use as little energy as possible. Our experiments on an IoT testbed highlight trade-offs between privacy, model performance, hardware utilization, and energy consumption. The linear regression model delivers the most accurate models, and its hardware utilization is also lower than that of the ML approach. However, its privacy level is lower due to its deterministic nature. It is prone to reverse-engineering the raw data based on the clients' updates.

More and more FL and FA systems have emerged, and it is paramount to have a standardized form to describe them to easily compare them. Multiple survey papers in this area exist. We cluster each survey into the following categories: Quantitative analysis [42, 83], tutorial [84, 85, 86, 87, 88], domain-specific [89, 90, 91], and taxonomy [92, 93, 94]. However, those surveys mainly focus on FL and lack a coherent definition of what belongs to FL. We distilled FC characteristics after looking at over two hundred FC use cases. Those characteristics include a description of the basic building blocks (client selection, aggregation, and communication) required for all FC systems and widely used extensions (privacy enhancements and compression). Our survey clusters a wide range of FC systems based on our FC system taxonomy. We identified often-used combinations of basic building blocks, widely used FL frameworks, and a need for FC systems running on actual distributed hardware. The favorite aggregation strategy is FedAvg, and about 50 % of surveyed papers do not specify which framework they use for their experiments. There is also a lack of more advanced client selection strategies that consider resource availability on client loss or the impact of clients on the overall loss of the merged model.

Our findings and contributions add to the energy informatics and FC knowledge pool. We highlight the importance of running FC experiments on physically distributed hardware to capture otherwise invisible effects. Those insights include network traffic, energy consumption, and scalability effects when increasing the number of participating clients. Our experiments focus on ARM-based edge devices such as Raspberry Pis. There needs to be more experimenting with computing architecture, such as x86 or RISC-V CPUs.

5

Conclusions

Legal constraints and personal sentiments of decision-makers hinder the development of cross-entity data sharing. FC can help to speed up the shift towards a data-driven economy by leveraging already existing data silos. It enables stakeholders to either monetize their existing data or get access to new data sources to improve current services or create new ones. While FC might help with new business models, it is important to quantify its energy footprint in advance to avoid adverse scaling effects.

The network traffic over the internet is increasing steadily, and the same applies to the energy consumption of ICT. Therefore, it is crucial to consider the impact of FC systems on those metrics at the beginning of the development process of new algorithms. Our IoT testbed enables us to capture the energy footprint of the server and all clients. We identified deviations between a centralized training approach and FC. Combining FC with additional privacy-enhancing techniques, such as DP, significantly increases energy costs for privacy. With our testbed, we also highlight scalability issues concerning the systems' total energy consumption and training time. Our experiments enable others to better understand the potential impact of FC with and without DP on the energy footprint of their system. They shed light on the ever-increasing ecological effects of ICT and software. An increase in data privacy comes with higher energy costs and complexity. Our energy footprint benchmarks help to better balance privacy and energy consumption. However, there needs to be a rule of thumb or any experience finding a suitable balance of those two metrics for specific use cases in advance.

Our experiments on combining FC with building management systems and battery electric vehicles highlight the possibility of achieving similar model prediction performance compared to a centralized approach. This insight enables stakeholders from the industry to develop data-driven business models without interfering with data privacy laws. Such models can increase energy efficiency in the building and mobility sector or reduce maintenance and failure rates. A decrease in repairs reduces service costs and waste. Those improvements help achieve the Eu's goals of reducing energy consumption and increasing energy efficiency, which increases energy productivity. Nevertheless, our experiments highlight the increase in complexity of leveraging distributed data sources in a privacy-preserving fashion. Therefore, our work also points to future research directions to decrease FC systems' complexity and reduce their total energy consumption.

It is also challenging to match ML architectures and aggregation strategies to a potentially unknown label distribution on the client side. The same applies to estimating which tools to use for a specific use case in advance to achieve a pre-defined performance metric threshold more quickly. A better understanding of the model architecture helps reduce iterations to fine-tune hyperparameters and save energy. Additionally, fine-tuning via iterations on distributed devices with potentially multiple different ownership increases complexity compared to a centralized approach. Therefore, besides a technical dimension, there is also an organizational one due to resource availability on the client side and the need for proper processes, pipelines, and accounting. Our literature research highlights the lack of answering economical questions for FC systems. It also shows the current focus in testing new aggregation algorithms and client selection algorithms on an emulated FC system to improve model prediction performance instead of relying on physically distributed devices.

Bibliography

[1] Transforma Insights. *Internet of Things (IoT) and non-IoT active device connections worldwide from 2010 to 2025*. Accessed November 21, 2023. URL: https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/.

[2] International Energy Agency. *Digitalisation and Energy*. Accessed November 29, 2023. 2017. URL: https://www.iea.org/reports/digitalisation-and-energy.

[3] Enerdata. *Net electricity consumption worldwide in select years from 1980 to 2022 (in terawatt-hours)*. Sept. 2023. URL: https://www.statista.com/statistics/280704/world-power-consumption/.

[4] The Radicati Group. *Forecast number of mobile devices worldwide from 2020 to 2025 (in billions)*. Accessed November 22, 2023. URL: https://www.statista.com/statistics/245501/multiple-mobile-device-ownership-worldwide/.

[5] P. Villalobos and A. Ho. *Trends in Training Dataset Sizes*. Accessed: 2023-11-22. 2022. URL: https://epochai.org/blog/trends-in-training-dataset-sizes.

[6] J. Dodge, T. Prewitt, R. Tachet des Combes, et al. "Measuring the Carbon Intensity of AI in Cloud Instances." In: *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '22. Seoul, Republic of Korea: Association for Computing Machinery, 2022, pp. 1877–1894. ISBN: 9781450393522. DOI: 10.1145/3531146.3533234. URL: https://doi.org/10.1145/3531146.3533234.

[7] E. Strubell, A. Ganesh, and A. McCallum. *Energy and Policy Considerations for Deep Learning in NLP*. 2019. arXiv: 1906.02243 [cs.CL].

[8] A. de Vries. "The growing energy footprint of artificial intelligence." In: *Joule* 7.10 (2023), pp. 2191–2194. ISSN: 2542-4351. DOI: https://doi.org/10.1016/j.joule.2023.09.004. URL: https://www.sciencedirect.com/science/article/pii/S2542435123003653.

[9] G. Kamiya. *Data Centres and Data Transmission Networks*. Accessed March 22, 2023. 2022. URL: https://www.iea.org/reports/data-centres-and-data-transmission-networks.

[10] L. Hilty, V. Coroama, M. Eicker, T. Ruddy, and E. Thiébaud (-Müller). "The Role of ICT in Energy Consumption and Energy Efficiency." In: (Jan. 2009).

[11] E. Gelenbe and Y. Caseau. "The Impact of Information Technology on Energy Consumption and Carbon Emissions." In: *Ubiquity* 2015 (June 2015). DOI: 10.1145/2755977. URL: https://doi.org/10.1145/2755977.

[12] Raspberry Pi Ltd. *Raspberry Pi Documentation - Processors*. Accessed December 18, 2023. 2023. URL: https://www.raspberrypi.com/documentation/computers/processors.html.

[13] A. Narayanan and V. Shmatikov. *How To Break Anonymity of the Netflix Prize Dataset*. 2006. DOI: 10.48550/ARXIV.CS/0610105. URL: https://arxiv.org/abs/cs/0610105.

[14] F. M. Naini, J. Unnikrishnan, P. Thiran, and M. Vetterli. "Where You Are Is Who You Are: User Identification by Matching Statistics." In: *IEEE Transactions on Information Forensics and Security* 11.2 (2016), pp. 358–372. DOI: 10.1109/TIFS.2015.2498131.

[15] H. Zang and J. Bolot. "Anonymization of Location Data Does Not Work: A Large-Scale Measurement Study." In: *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*. MobiCom '11. Las Vegas, Nevada, USA: Association for Computing Machinery, 2011, pp. 145–156. ISBN: 9781450304924. DOI: 10.1145/2030613.2030630. URL: https://doi-org.eaccess.tum.edu/10.1145/2030613.2030630.

[16] L. Rocher, J. M. Hendrickx, and Y.-A. de Montjoye. "Estimating the success of re-identifications in incomplete datasets using generative models." In: *Nature Communications* (2019). DOI: 10.1038/s41467-019-10933-3.

[17] 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)*. 2016.

[18] Personal Data Protection Commission Singapore. *Personal Data Protection Act*. 2014.

[19] State of California Department of Justice. *California Consumer Privacy Act of 2018 [1798.100 - 1798.199.100]*. 2018.

[20] The Office of the Attorney General. *Colorado Privacy Act*. 2021.

[21] The Office of the Attorney General. *The Connecticut Data Privacy Act*. 2022.

[22] The Office of the Attorney General. *Virginia Consumer Data Protection Act*. 2023. URL: https://law.lis.virginia.gov/vacode/title59.1/chapter53/.

[23] P. Ferrara and F. Spoto. "Static Analysis for GDPR Compliance." In: Jan. 2018.

[24] D. Basin, S. Debois, and T. Hildebrandt. "On Purpose and by Necessity: Compliance Under the GDPR." In: *Financial Cryptography and Data Security*. Ed. by S. Meiklejohn and K. Sako. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018, pp. 20–37. ISBN: 978-3-662-58387-6.

[25] A. Trask, E. Bluemke, B. Garfinkel, C. G. Cuervas-Mons, and A. Dafoe. *Beyond Privacy Trade-offs with Structured Transparency*. 2020. DOI: 10.48550/ARXIV.2012.08347. URL: https://arxiv.org/abs/2012.08347.

[26] S. Bose. "Evolution of ESG Reporting Frameworks." In: *Values at Work: Sustainable Investing and ESG Reporting*. Ed. by D. C. Esty and T. Cort. Cham: Springer International Publishing, 2020, pp. 13–33. ISBN: 978-3-030-55613-6. DOI: 10.1007/978-3-030-55613-6_2. URL: https://doi.org/10.1007/978-3-030-55613-6_2.

[27] M. Hähnel, B. Döbel, M. Völp, and H. Härtig. "Measuring Energy Consumption for Short Code Paths Using RAPL." In: *SIGMETRICS Perform. Eval. Rev.* 40.3 (Jan. 2012), pp. 13–17. ISSN: 0163-5999. DOI: 10.1145/2425248.2425252. URL: https://doi-org.eaccess.tum.edu/10.1145/2425248.2425252.

[28] M. Colmant, M. Kurpicz, P. Felber, et al. "Process-level Power Estimation in VM-based Systems." In: *European Conference on Computer Systems (EuroSys)*. Ed. by T. Harris and M. Herlihy. EuroSys'15: Proceedings of the Tenth European Conference on Computer Systems. Bordeaux, France: ACM, Apr. 2015, p. 14. DOI: 10.1145/2741948.2741971. URL: https://inria.hal.science/hal-01130030.

[29] M. Fahad, A. Shahid, R. R. Manumachu, and A. Lastovetsky. "A Comparative Study of Methods for Measurement of Energy of Computing." In: *Energies* 12.11 (2019). ISSN: 1996-1073. DOI: 10.3390/en12112204. URL: https://www.mdpi.com/1996-1073/12/11/2204.

[30] Maintenance Team. *NVML API Reference.* Accessed December 17, 2023. 2023. URL: https://docs.nvidia.com/deploy/nvml-api/nvml-api-reference.html#nvml-api-reference.

[31] R. Schwermer, J. Buchberger, R. Mayer, and H.-A. Jacobsen. "Federated Office Plug-Load Identification for Building Management Systems." In: e-Energy '22. Virtual Event: Association for Computing Machinery, 2022, pp. 114–126. ISBN: 9781450393973. DOI: 10.1145/3538637.3538845.

[32] R. Schwermer, R. Mayer, and H.-A. Jacobsen. "Energy vs Privacy: Estimating the Ecological Impact of Federated Learning." In: *Proceedings of the 14th ACM International Conference on Future Energy Systems.* e-Energy '23. Orlando, FL, USA: Association for Computing Machinery, 2023, pp. 347–352. DOI: 10.1145/3575813.3597344.

[33] R. Schwermer, E.-A. Bicer, P. Schirmer, R. Mayer, and H.-A. Jacobsen. "Federated Computing in Electric Vehicles to Predict Coolant Temperature." In: *Proceedings of the 24th International Middleware Conference Industrial Track.* Middleware Industrial Track '23. Bologna, Italy: Association for Computing Machinery, 2023, pp. 8–14. ISBN: 9798400704277. DOI: 10.1145/3626562.3626829. URL: https://doi.org/10.1145/3626562.3626829.

[34] K. Bonawitz, H. Eichner, W. Grieskamp, et al. *Towards Federated Learning at Scale: System Design.* 2019. DOI: 10.48550/ARXIV.1902.01046. URL: https://arxiv.org/abs/1902.01046.

[35] D. Wang, S. Shi, Y. Zhu, and Z. Han. "Federated Analytics: Opportunities and Challenges." In: *IEEE Network* 36.1 (2022), pp. 151–158. DOI: 10.1109/MNET.101.2100328.

[36] P. Kairouzet al. *Advances and Open Problems in Federated Learning*. 2021.

[37] B. Agüera y Arcas et al. *Federated Analytics: Collaborative Data Science without Data Collection.* 2020. URL: https://ai.googleblog.com/2020/05/federated-analytics-collaborative-data.html.

[38] A. R. Elkordy, Y. H. Ezzeldin, S. Han, et al. "Federated Analytics: A survey." In: (2023). DOI: 10.48550/ARXIV.2302.01326. URL: https://arxiv.org/abs/2302.01326.

[39] T. Everitt and M. Hutter. “Universal Artificial Intelligence-Practical Agents and Fundamental Challenges.” In: 2016.

[40] K. Frankish and W. M. Ramsey. *The Cambridge handbook of artificial intelligence.* 2014. ISBN: 978-0-521-87142-6.

[41] J. "Walmsley. “Classical Cognitive Science and Good Old Fashioned AI.” In: *Mind and Machine.* London: Palgrave Macmillan UK, 2012, pp. 30–64. ISBN: 978-1-137-28342-9. DOI: 10.1057/9781137283429_3. URL: https://doi.org/10.1057/9781137283429_3.

[42] A. Farooq, A. Feizollah, and M. H. ur Rehman. “Federated Learning Research: Trends and Bibliometric Analysis.” In: *Federated Learning Systems: Towards Next-Generation AI.* Ed. by M. H. u. Rehman and M. M. Gaber. Cham: Springer International Publishing, 2021, pp. 1–19. ISBN: 978-3-030-70604-3. DOI: 10.1007/978-3-030-70604-3_1. URL: https://doi.org/10.1007/978-3-030-70604-3_1.

[43] S. K. Lo, Q. Lu, C. Wang, H.-Y. Paik, and L. Zhu. “A Systematic Literature Review on Federated Machine Learning: From a Software Engineering Perspective.” In: *ACM Comput. Surv.* 54.5 (May 2021). ISSN: 0360-0300. DOI: 10.1145/3450288. URL: https://doi.org/10.1145/3450288.

[44] C. He, S. Li, J. So, et al. *FedML: A Research Library and Benchmark for Federated Machine Learning.* 2020. DOI: 10.48550/ARXIV.2007.13518. URL: https://arxiv.org/abs/2007.13518.

[45] Y. Hu, Y. Zhou, J. Xiao, and C. Wu. *GFL: A Decentralized Federated Learning Framework Based On Blockchain.* 2020. DOI: 10.48550/ARXIV.2010.10996. URL: https://arxiv.org/abs/2010.10996.

[46] Y. Li, H. Li, G. Xu, T. Xiang, and R. Lu. “Practical Privacy-Preserving Federated Learning in Vehicular Fog Computing.” In: *IEEE Transactions on Vehicular Technology* 71.5 (2022), pp. 4692–4705. DOI: 10.1109/TVT.2022.3150806.

[47] M. Ekmefjord, A. Ait-Mlouk, S. Alawadi, et al. “Scalable federated machine learning with FEDn.” In: *arXiv preprint arXiv:2103.00148* (2021).

[48] Y. Xie, Z. Wang, D. Gao, et al. *FederatedScope: A Flexible Federated Learning Platform for Heterogeneity.* 2022. arXiv: 2204.05011 [cs.LG].

[49] D. J. Beutel, T. Topal, A. Mathur, et al. *Flower: A Friendly Federated Learning Research Framework.* 2021. arXiv: 2007.14390 [cs.LG].

[50] P. Foley, M. J. Sheller, B. Edwards, et al. “OpenFL: the open federated learning library.” In: *Physics in Medicine and Biology* 67.21 (Oct. 2022), p. 214001. DOI: 10.1088/1361-6560/ac97d9. URL: https://dx.doi.org/10.1088/1361-6560/ac97d9.

[51] Umweltbundesamt. *Energieproduktivität.* Accessed December 9, 2023. 2023. URL: https://www.umweltbundesamt.de/daten/energie/energieproduktivitaet#der-begriff-der-energieproduktivitat-und-endenergieproduktivitat-seit-1990.

[52] Eurostat. *Energy statistics - an overview.* Accessed May 7, 2023. 2022. URL: https://ec.europa.eu/eurostat/statistics-explained/index.php?oldid=444923#Energy_intensity.

[53] W. S. Jevons. *The Coal Question; An Inquiry Concerning the Progress of the Nation, and the Probable Exhaustion of Our Coal Mines.* Macmillan and Co., 1865.

[54] K. Gillingham, M. J. Kotchen, D. S. Rapson, and G. Wagner. *The rebound effect is overplayed.* 2013. DOI: 10.1038/493475a. URL: https://doi.org/10.1038/493475a.

[55] K. Gillingham, D. Rapson, and G. Wagner. "The Rebound Effect and Energy Efficiency Policy." In: *Review of Environmental Economics and Policy* 10.1 (2016), pp. 68–88. DOI: 10.1093/reep/rev017. URL: https://doi.org/10.1093/reep/rev017.

[56] Umweltbundesamt. *https://de.statista.com/statistik/daten/studie/30100/umfrage/dauer-vergangener-rezessionen-in-deutschland/.* Accessed December 16, 2023. 2009. URL: https://de.statista.com/statistik/daten/studie/30100/umfrage/dauer-vergangener-rezessionen-in-deutschland/.

[57] H. Wang, C. Si, and J. Zhao. *A Federated Learning Framework for Non-Intrusive Load Monitoring.* 2021. DOI: 10.48550/ARXIV.2104.01618. URL: https://arxiv.org/abs/2104.01618.

[58] H. Wang, C. Si, J. Zhao, G. Liu, and F. Wen. *Fed-NILM: A Federated Learning-based Non-Intrusive Load Monitoring Method for Privacy-Protection.* 2021. arXiv: 2105.11085 [cs.LG].

[59] C. Briggs, Z. Fan, and P. Andras. *Federated Learning for Short-term Residential Energy Demand Forecasting.* 2021. arXiv: 2105.13325 [cs.LG].

[60] A. Taik and S. Cherkaoui. "Electrical Load Forecasting Using Edge Computing and Federated Learning." In: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC).* 2020, pp. 1–6. DOI: 10.1109/ICC40277.2020.9148937.

[61] Y. L. Tun, K. Thar, C. M. Thwal, and C. S. Hong. "Federated Learning based Energy Demand Prediction with Clustered Aggregation." In: *2021 IEEE International Conference on Big Data and Smart Computing (BigComp).* 2021, pp. 164–167. DOI: 10.1109/BigComp51126.2021.00039.

[62] Y. Guo, D. Wang, A. Vishwanath, C. Xu, and Q. Li. "Towards Federated Learning for HVAC Analytics: A Measurement Study." In: e-Energy '20. Virtual Event, Australia: Association for Computing Machinery, 2020, pp. 68–73. ISBN: 9781450380096. DOI: 10.1145/3396851.3397717. URL: https://doi.org/10.1145/3396851.3397717.

[63] S. Lee and D.-H. Choi. "Federated Reinforcement Learning for Energy Management of Multiple Smart Homes With Distributed Energy Resources." In: *IEEE Transactions on Industrial Informatics* 18.1 (2022), pp. 488–497. DOI: 10.1109/TII.2020.3035451.

[64] W. Yang, Y. Zhang, W. Yang Bryan Lim, et al. "Privacy is not Free: Energy-Aware Federated Learning for Mobile and Edge Intelligence." In: *2020 International Conference on Wireless Communications and Signal Processing (WCSP).* 2020, pp. 233–238. DOI: 10.1109/WCSP49889.2020.9299703.

[65] X. Qiu, T. Parcollet, J. Fernandez-Marques, et al. *A first look into the carbon footprint of federated learning.* 2022. arXiv: 2102.07627 [cs.LG].

[66] X. Qiu, T. Parcollet, D. J. Beutel, et al. *Can Federated Learning Save The Planet?* 2020. DOI: 10.48550/ARXIV.2010.06537. URL: https://arxiv.org/abs/2010.06537.

[67] Y. Liu, J. J. Q. Yu, J. Kang, D. Niyato, and S. Zhang. "Privacy-Preserving Traffic Flow Prediction: A Federated Learning Approach." In: *IEEE Internet of Things Journal* 7.8 (2020), pp. 7751–7763. DOI: 10.1109/JIOT.2020.2991401.

[68] S. Wang, M. Chen, W. Saad, and C. Yin. "Federated Learning for Energy-Efficient Task Computing in Wireless Networks." In: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. Virtual: IEEE, 2020, pp. 1–6. DOI: 10.1109/ICC40277.2020.9148625.

[69] M. Abdul Salam, S. Taha, and M. Ramadan. "COVID-19 detection using federated machine learning." In: *PLOS ONE* 16.6 (June 2021), pp. 1–25. DOI: 10.1371/journal.pone.0252573. URL: https://doi.org/10.1371/journal.pone.0252573.

[70] A. Priyanshu, R. Naidu, F. Mireshghallah, and M. Malekzadeh. *Efficient Hyperparameter Optimization for Differentially Private Deep Learning*. 2021. arXiv: 2108.03888 [cs.LG].

[71] X. Zhu, J. Wang, Z. Hong, T. Xia, and J. Xiao. "Federated Learning of Unsegmented Chinese Text Recognition Model." In: *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. New York, NY, USA: Institute of Electrical and Electronics Engineers (IEEE), 2019, pp. 1341–1345. DOI: 10.1109/ICTAI.2019.00186.

[72] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, et al. "Energy Demand Prediction with Federated Learning for Electric Vehicle Networks." In: *2019 IEEE Global Communications Conference (GLOBECOM)*. Waikoloa, Hawaii, USA: IEEE, 2019, pp. 1–6. DOI: 10.1109/GLOBECOM38437.2019.9013587.

[73] X. Mo and J. Xu. "Energy-Efficient Federated Edge Learning with Joint Communication and Computation Design." In: *Journal of Communications and Information Networks* 6.2 (2021), pp. 110–124. DOI: 10.23919/JCIN.2021.9475121.

[74] X. Qu, S. Wang, Q. Hu, and X. Cheng. "Proof of Federated Learning: A Novel Energy-Recycling Consensus Algorithm." In: *IEEE Transactions on Parallel and Distributed Systems* 32.8 (2021), pp. 2074–2085. DOI: 10.1109/TPDS.2021.3056773.

[75] B. Ji, X. Zhang, S. Mumtaz, et al. "Survey on the Internet of Vehicles: Network Architectures and Applications." In: *IEEE Communications Standards Magazine* 4.1 (2020), pp. 34–41. DOI: 10.1109/MCOMSTD.001.1900053.

[76] D. Ye, R. Yu, M. Pan, and Z. Han. "Federated Learning in Vehicular Edge Computing: A Selective Model Aggregation Approach." In: *IEEE Access* 8 (2020), pp. 23920–23935. DOI: 10.1109/ACCESS.2020.2968399.

[77] J. Vyas, D. Das, and S. K. Das. "Vehicular Edge Computing Based Driver Recommendation System Using Federated Learning." In: *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. 2020, pp. 675–683. DOI: 10.1109/MASS50613.2020.00087.

[78] S. Liu, J. Yu, X. Deng, and S. Wan. "FedCPF: An Efficient-Communication Federated Learning Approach for Vehicular Edge Computing in 6G Communication Networks." In: *IEEE Transactions on Intelligent Transportation Systems* 23.2 (2022), pp. 1616–1629. DOI: 10.1109/TITS.2021.3099368.

[79] A. M. Elbir, B. Soner, S. Çöleri, D. Gündüz, and M. Bennis. "Federated Learning in Vehicular Networks." In: *2022 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*. 2022, pp. 72–77. DOI: 10.1109/MeditCom55741.2022.9928621.

[80] K. Tan, D. Bremner, J. L. Kernec, and M. Imran. “Federated Machine Learning in Vehicular Networks: A summary of Recent Applications.” In: *2020 International Conference on UK-China Emerging Technologies (UCET)*. 2020, pp. 1–4. DOI: 10.1109/UCET51115.2020.9205482.

[81] S. Sudhakar, V. Sze, and S. Karaman. “Data Centers on Wheels: Emissions From Computing Onboard Autonomous Vehicles.” In: *IEEE Micro* 43.1 (2023), pp. 29–39. DOI: 10.1109/MM.2022.3219803.

[82] ZF. *ZF ProAI: The Source of Vehicle Intelligence*. Accessed March 21, 2023. 2021. URL: https://www.zf.com/products/en/cars/stories/proai.html.

[83] S. K. Lo, Q. Lu, C. Wang, H.-Y. Paik, and L. Zhu. “A Systematic Literature Review on Federated Machine Learning: From a Software Engineering Perspective.” In: *ACM Comput. Surv.* 54.5 (May 2021). ISSN: 0360-0300. DOI: 10.1145/3450288. URL: https://doi.org/10.1145/3450288.

[84] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed. “Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications.” In: *IEEE Access* 8 (2020), pp. 140699–140725. DOI: 10.1109/ACCESS.2020.3013541.

[85] Q. Yang, Y. Liu, T. Chen, and Y. Tong. “Federated Machine Learning: Concept and Applications.” In: *ACM Trans. Intell. Syst. Technol.* 10.2 (Jan. 2019). ISSN: 2157-6904. DOI: 10.1145/3298981. URL: https://doi.org/10.1145/3298981.

[86] L. Li, Y. Fan, and K.-Y. Lin. “A Survey on federated learning.” In: *2020 IEEE 16th International Conference on Control and Automation (ICCA)*. 2020, pp. 791–796. DOI: 10.1109/ICCA51439.2020.9264412.

[87] P. Kairouz, H. B. McMahan, B. Avent, et al. *Advances and Open Problems in Federated Learning*. 2021. arXiv: 1912.04977 [cs.LG].

[88] H. G. Abreha, M. Hayajneh, and M. A. Serhani. “Federated Learning in Edge Computing: A Systematic Survey.” In: *Sensors* 2 (2022). ISSN: 1424-8220. DOI: 10.3390/s22020450. URL: https://www.mdpi.com/1424-8220/22/2/450.

[89] Q. Xia, W. Ye, Z. Tao, J. Wu, and Q. Li. “A survey of federated learning for edge computing: Research problems and solutions.” In: *High-Confidence Computing* 1.1 (2021), p. 100008. ISSN: 2667-2952. DOI: https://doi.org/10.1016/j.hcc.2021.100008. URL: https://www.sciencedirect.com/science/article/pii/S266729522100009X.

[90] C. Briggs, Z. Fan, and P. Andras. “A Review of Privacy-Preserving Federated Learning for the Internet-of-Things.” In: *Federated Learning Systems: Towards Next-Generation AI*. Ed. by M. H. u. Rehman and M. M. Gaber. Cham: Springer International Publishing, 2021, pp. 21–50. ISBN: 978-3-030-70604-3. DOI: 10.1007/978-3-030-70604-3_2. URL: https://doi.org/10.1007/978-3-030-70604-3_2.

[91] J. Zhou, S. Zhang, Q. Lu, et al. *A Survey on Federated Learning and its Applications for Accelerating Industrial Internet of Things*. 2021. arXiv: 2104.10501 [cs.DC].

[92] Q. Li, Z. Wen, Z. Wu, et al. “A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection.” In: *IEEE Transactions on Knowledge and Data Engineering* (2021), pp. 1–1. DOI: 10.1109/tkde.2021.3124599.

[93] S. Abdulrahman, H. Tout, H. Ould-Slimane, et al. “A Survey on Federated Learning: The Journey From Centralized to Distributed On-Site Learning and Beyond.” In: *IEEE Internet of Things Journal* 8.7 (2021), pp. 5476–5497. DOI: 10.1109/JIOT.2020.3030072.

[94] X. Yin, Y. Zhu, and J. Hu. “A Comprehensive Survey of Privacy-Preserving Federated Learning: A Taxonomy, Review, and Future Directions.” In: *ACM Comput. Surv.* 54.6 (July 2021). ISSN: 0360-0300. DOI: 10.1145/3460427. URL: https://doi.org/10.1145/3460427.

[95] R. Schwermer, R. Mayer, and H.-A. Jacobsen. *Federated Computing – Survey on Building Blocks, Extensions and Systems.* 2024. arXiv: 2404.02779.

Appendix A

Federated Office Plug-Load Identification For Building Management Systems

Federated Office Plug-Load Identification for Building Management Systems

René Schwermer
rene.schwermer@tum.de
Technical University of Munich
Germany

Jonas Buchberger
jonas.buchberger@tum.de
Technical University of Munich
Germany

Ruben Mayer
ruben.mayer@tum.de
Technical University of Munich
Germany

Hans-Arno Jacobsen
jacobsen@eecg.toronto.edu
University of Toronto
Canada

ABSTRACT

Energy consumption in buildings is responsible for 40 % of the final energy consumption in the European Union and the United States of America. In addition to thermal energy, buildings require electricity for all kinds of appliances. Regulatory constraints such as energy labels aim at increasing the energy efficiency of large appliances such as fridges and washing machines. However, they only partially cover plug-loads. The amount of electricity consumption of unregulated plug-loads such as mobile phones, USB chargers and kettles is continuously increasing. For European households, their share of electricity consumption reached 25 % in 2018. Additional data about the plug-loads usage can help decrease the energy consumption of buildings by improving energy management systems, applying peak-shaving or demand-side management. People live and work in buildings, making such data privacy sensitive. Federated Learning (FL) helps to leverage these data without violating regulatory frameworks such as the General Data Protection Regulation. We use a high-frequency energy data set of office appliances (BLOND) to train four appliance classifiers (CNN, LSTM, ResNet and DenseNet). We investigate the effect of different data distributions (entire dataset, IID and non-IID) and training methods on four performance metrics (accuracy, F1 score, precision and recall). The results show that a non-IID setup decreases all performance metrics for some model architectures by 44 %. However, our LSTM model even with a non-IID labels achieves similar F1 scores compared to central training. Additionally, we show the importance of client selection in FL architectures to reduce the overall training time and we quantify the decrease in network traffic compared to a central training approach, the energy consumption and scalability.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; *Distributed computing methodologies.*

KEYWORDS

Federated Learning, Testbed, Plug-Load, Building Management

ACM Reference Format:
René Schwermer, Jonas Buchberger, Ruben Mayer, and Hans-Arno Jacobsen. 2022. Federated Office Plug-Load Identification for Building Management Systems. In *The Thirteenth ACM International Conference on Future Energy Systems (e-Energy '22), June 28-July 1, 2022, Virtual Event, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3538637.3538845

1 INTRODUCTION

The Green Deal from the European Union (EU) with the Paris Agreement aims to achieve zero net emissions of greenhouse gases by 2050 and reduce global warming to an average temperature increase of 2 °C [47, 73]. Several approaches have been implemented to achieve these goals. Renewable energy sources, such as wind, solar and water, replace the currently running fossil fuel-based power plants, the energy efficiency of machinery and processes is increased or the energy consumption is reduced. All these approaches can be applied to any of the three sectors mobility, electricity and heating. Improving the energy efficiency of buildings is a key tool to achieve these targets. In the EU, buildings consume the greatest share of energy and have the largest energy savings potential [54].

Tenants and residents use buildings for work or living. Depending on the number of deployed sensors, they generate sensitive data while working or living in offices or homes. Monitoring these data enables stakeholders to increase energy efficiency by either passively proposing saving strategies or actively switching actuators. Some examples are peak-shaving for commercial buildings or demand side management [17, 41, 42]. However, in different countries or states sensitive information is protected by laws such as the General Data Protection Regulation (GDPR) in the EU [72] and the Californian Consumer Privacy Act [55]. European privacy regulators scrutinise how employers collect workers' personal data and hand out fines when they violate the GDPR. Some examples are unauthorized video surveillance or collection of employee details about their health and religion [11]. These cases and regulatory

e-Energy '22, June 28-July 1, 2022, Virtual Event, USA

ACM ISBN 978-1-4503-9397-3/22/06…$15.00
https://doi.org/10.1145/3538637.3538845

uncertainties can hinder the deployment of a centralised energy management system, making it necessary to use technology to avoid privacy violations while obtaining valuable information. Federated Learning (FL) tries unravelling this contradiction between data privacy and the potential from insights based on private data. In our use case, it helps to combine the GDPR and energy transition to reduce the ecological footprint of buildings. The challenge is to develop well performing appliance classifiers in an FL system running on edge devices with an ARM architecture. In this paper, we use high-frequency electricity time-series data from an office building to train an appliance detection model in different system architectures. We evaluate four different models based on accuracy, precision, recall and F1 score. The training process is benchmarked through different hardware-related metrics. They include monitoring of memory, CPU, network usage and energy consumption during mode training.

The contributions of this study are as follows:

(1) Training and evaluating four different machine learning (ML) architectures (convolutional neural network (CNN), long short-term memory (LSTM), ResNet, DenseNet) on a high-frequency energy data set, building-level office environment (BLOND). Most appliance detection models use data with a frequency of maximum 1 Hz. By gaining knowledge into working with high-frequency data, we help to expand the usage of electricity data beyond appliance detection, e.g., for condition monitoring in industry processes.
(2) Investigating the effect of training data distributions (central, IID and non-IID) and variations of model training (early stopping, learning rate scheduler and aggregation strategy) on model metrics. Our results show that it is possible to train ML models on non-IID data while achieving a model performance that can compete with a centrally trained model.
(3) Edge devices have limited computational and energy resources and are installed at remote locations with network constraints. Our results show the importance of running FL on physically separated devices to capture the effect of different setups on network and CPU load, training time and energy consumption.
(4) Highlighting the importance of plug-loads on energy savings in buildings and providing some examples on how to leverage our models to improve energy efficiency and maintenance costs. Knowing where saving potentials are and how to leverage them can help to decrease the energy consumption of private and commercial buildings.

Finally, all models and source code are made open source to enable other groups to reproduce and improve the pipeline. We use the BLOND data set, which is already open-source [33].

The remainder of the paper is organised as follows. First, we discuss related work in Section 2. We explain FL in Section 3 and Section 4 presents some context for plug-loads in buildings from an energy perspective. Our experiments consist of three parts (data preparation, model execution and evaluation). Figure 1 gives an overview on how these components work together. The paper structure follows the same pipeline, enabling the reader to quickly browse to a specific topic. Section 5 explains the preparation of the data set. Section 6 presents the design and experiments based on one pick from three different categories (model, data distribution and hardware). We evaluate the hardware and model metrics in addition to the effect of scaling up the our setup on the training time in Section 7. Some example use-cases of our ML model are illustrated in Section 8. We present some learned lessons in Section 9. Finally, Section 10 presents the conclusion.

2 RELATED WORK

To the best of our knowledge, this is the first work on plug-load appliance detection on high-frequency energy data in an FL environment. In the next paragraph we see others who implemented appliance detection of plug-loads or FL on energy data in households, including larger appliances such as washing machines and electric vehicle charging stations. This section presents some of these works, starting with plug-load appliance detection.

Based on a 15 min frequency data set of office occupancy sensors and plug-loads, Mahdavi et al. [40] predicted user electricity consumption. Wireless energy meters measure the plug-load. Their approach does not facilitate data privacy (all plug-loads are associated with a user), and all data is stored centrally. Based on smart plugs, Reddy et al. [62] examined electricity consumption with the ability to actively interact with appliances. They used k-nearest neighbours, Naive Bayes, logistic regression and random forest algorithms to identify plug-loads and compared their respective results. Radhakrishna et al. [30] developed a smart plug. A central server stores all measurements with different levels of detail and privacy. The device has information about its electrical ratings and can act decisively to reduce potential fire hazards due to overcharging of malfunctioning plug-loads.

The following research in the context of FL on energy data runs all on simulated nodes on one machine. This allows for an easy setup. However, it neglects network constraints and possibly related computational constraints due to CPU stalls. The evaluation of hardware metrics during FL training is not well investigated in research. Other papers focus on model performance on simulated nodes [9, 16, 35, 68, 71, 76, 77, 80]. These setups do not cover the system behaviour under real-world conditions. We fill this research gap and introduce some future research questions. One approach to address this issue is to calculate a theoretical network load while training an LSTM model on data from Pecan Street Inc.'s Dataport site with TensorFlow Federated [68]. The focus in the FL energy domain is on LSTM models using either *TensorFlow* or *PyTorch* [9, 68, 71]. Taik et al. used a FL specific framework (*TensorFlow Federated*). The focus is on household applications and stationary equipment [68]. Exceptions are Yang et .al [80] who ran experiments on mobile devices and Wan et al. [77] who used a data set of heavy-machinery from a Brazilian poultry feed factory. In both cases, they clustered the data to have similar properties. Doing so might interfere with a normally not independent and identically distributed (non-IID) data set in FL, because all houses have similar characteristics. Another LSTM was trained by Briggs et al. [9] with PyTorch on eight virtual machines on the "SmartMeter Energy Consumption Data in London Households" data set and Guo et al. [16] trained multiple models (linear regression, neural network, random forest, boosting tree and AdaBoost) with PyTorch on a simulated cloud-edge environment on mostly non publicly available data.

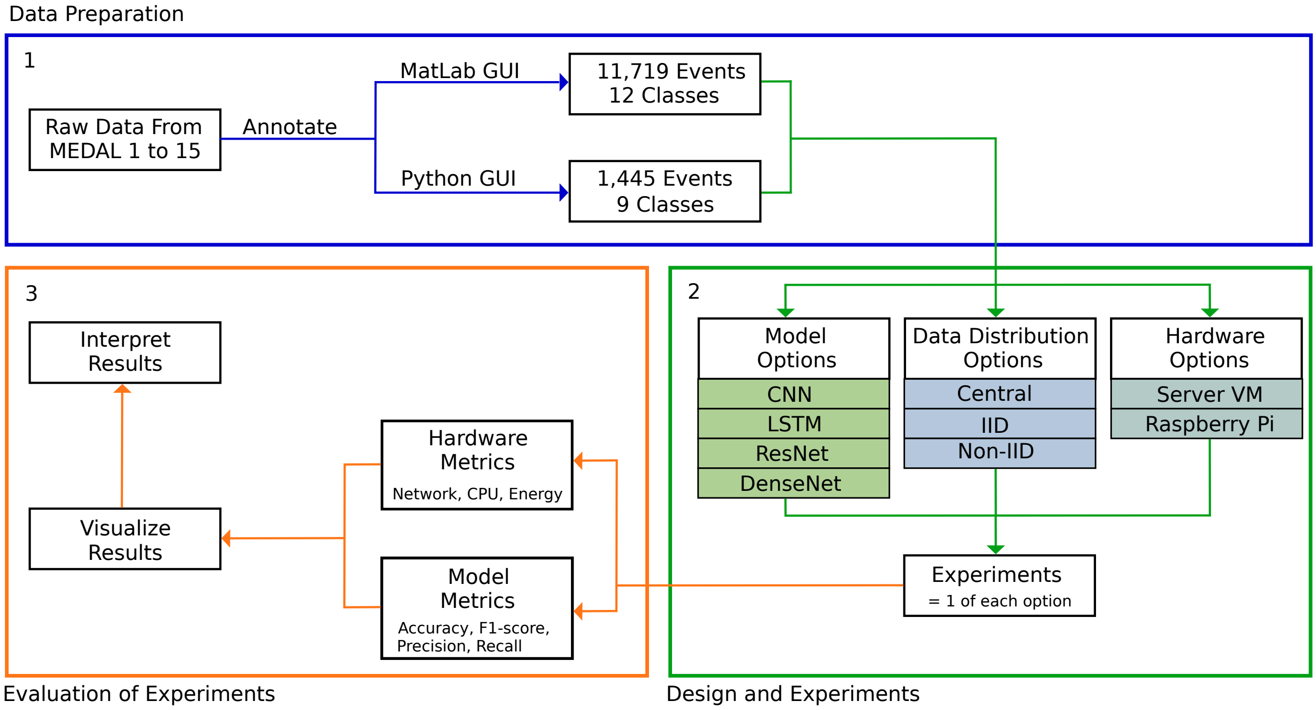


Figure 1: Pipeline of our research approach. In the first block (Prepare Data) we generate training data based on two annotation runs. Block 2 (Design and Experiment) shows how we build our experiments based on three different parameters. Finally, in block 3 (Evaluate Experiment) we introduce the model and hardware metrics we are evaluating.

Our work goes beyond these by working with high-frequency energy data in an FL system with physically separated nodes. This enables us to investigate the behavior of different model training strategies on its performance metrics. Additionally, we can quantify the network and energy usage. Furthermore, we apply new model architectures like ResNet and DenseNet to the energy domain and investigate the impact of memory usage, training time and model complexity on the energy consumption. This is especially important for remotely installed and battery powered edge devices.

3 FEDERATED LEARNING

One subcategory of privacy preserving ML is FL. This method enables data scientists to work on remote data while keeping it privately. The data stay at its origin and each device trains its local model with its individual data. Only the model parameters are transferred to a central server where each individual model is incorporated into one central model. FL aims to open up existing data silos while considering privacy concerns. Different approaches have been developed to combine multiple models such as (weighted) federated averaging (FedAvg) [8, 44], matched averaging (FedMa) [14, 78], FedPer [14], FedSGD [44], FedProx [36] and FedDane [37]. We use FedAvg.

In FL, the data scientist works with distributed data. This can lead to non-IID data because of an uneven distribution of features and labels over the connected devices. Different definitions of FL exist [31]. Horizontal FL describes a data set with the same features on multiple devices, but the labels vary. Some examples are the next word written prediction on Google's Gboard [8] and traffic flow prediction with the Caltrans Performance Measurement System data set [38]. Vertical FL uses multiple feature sets from different sources to run inference on one object. An example arises from financial sector where retailers and banks store historical data on the same person, but with different features [79]. Transfer FL is vertical FL in combination with a pre-trained model [79].

In our context of office appliance classification, one device might only have measurements for certain appliances. Training two separate models on these two data sets and averaging them can give worse results than a central approach. Other challenges include the non-IID of data over multiple devices and the system-induced bias of distributed systems. With our experiments and evaluations we give a reference case on physically separated hardware. Nodes can also drop out during a training process.

Multiple organisations, institutes or other stakeholders develop FL frameworks. Some of them have a specific focus on a certain domain, some are not for commercial use and others are not further developed. Several frameworks have been developed, such as *PySyft* from openMined [67], *TensorFlow Federated* from Google [8], *IBM FL* from IBM [39], *FedAI/FATE* from WeBank [15], *Clara SDK* from Nvidia [70], *FedML* [18], *Paddle FL* from Baidu, Fed-BioMed [66] and *Flower* [7]. *LEAF* [10] provides tools to benchmark different pre-selected models in a FL setting.

4 ELECTRICITY CONSUMPTION IN BUILDINGS

A general way to distinguish between the usage of buildings is residential and non-residential or commercial/service [48]. In this section we will compare the energy and plug-load consumption of the EU and USA. In Europe and the USA, buildings accounted for 40 % of the final energy consumption in 2021 and 2020 [2, 69]. Influencing factors are legal differences and different housing standards. Both numbers provide an order of magnitude for industrialised countries. This paper focuses on electricity and specifically on plug-loads.

4.1 Plug-Loads

A building and its equipment consume different forms of energy, particularly thermal and electrical energy. The latter includes plug-loads. This study focuses on unregulated plug-loads, which cover energy used by products and equipment powered by an ordinary alternate current (AC) plug. Plug-loads are also known as miscellaneous electronic loads or plug-and-process-load. In this context, we restrict plug-loads to small appliances such as routers, speakers, computers, monitors, tablets, printers, projectors, paper shredders and other consumer electronics. Such devices also exist in private households, which makes our results also applicable to that area. Big appliances such as washing machines, fridges and servers are excluded from our office data set. The energy consumption due to plug-loads is increasing globally. However, there are regional differences due to ownership saturation and efficiency improvements [24]. To motivate the need for a better understanding of plug-load consumption in buildings, we give an overview of their contribution to the overall energy consumption in buildings and how the consumption changed over time.

The share of electricity consumption in households within the EU ranks second in energy consumption and it increased from 21 % in 2000 to 25 % in 2018 [52], which is primarily attributed to electrical appliances. These captive uses of electricity exclude thermal uses. The share for large appliances decreased from 2000 to 2018 by 24 % from 1100 to 800 kWh per household and year. The European Ecodesign and Energy Labelling Framework Directives enforces manufacturers to label their products according to their energy usage and to continuously decrease power consumption. Other countries have similar energy labels. Figure 2 shows different examples. This image is not exhaustive, as Singapore, India, the Philippines and Jamaica also have an energy label. However, all labels focus on large appliances. Table 1 presents an overview of which energy label covers which small appliances. The focus is on televisions, computers and monitors. However, some small appliances might not be subject to registration and labelling requirements, but they might be under other mandatory standards that give a minimum value for energy efficiency. Therefore, the overall plug-load consumption increases as the energy consumption of some small appliances decreases. This can be because of rebound effects or unregulated devices such as USB chargers, phones and power supplies [53]. The energy labelling EU directive is also in place for small appliances such as computers, fans and vacuum cleaners; however it does not cover other appliances such as monitors, printers and USB chargers [59]. Therefore, the rapid growth of small appliances of 18 % within the last 12 years till 2020 cannot counterbalance the decrease in energy consumption for large appliances. All these numbers are averages from all 27 EU States [52]. However, Rudzki et al. [65] showed that consumption in office buildings could significantly vary depending on the building type and tenants. Additionally, some consumers focus on the energy efficiency on a label, but not on its actual electricity consumption and therefore potentially overestimate the energy efficiency of a product, leading to higher energy costs than expected [75].

The plug-load share in commercial buildings in the USA increased from 33 % in 2008 to 47 % in 2020 [50]. Retail buildings in the USA have a similar distribution [51]. The energy consumption of other consumers such as lighting and space heating, also decreased [1, 43]. The energy consumption for lighting decreased because of increasing share of more efficient LEDs than fluorescent lamps [23, 52]; therefore it partially explains the higher share of plug-loads. The results of Attia et al. [6] showed this relationship in the example of Egypt. Another reason for the higher share of plug-loads at the overall energy consumption is the increase in small appliances such as routers, set-top boxes and smart speakers, which is also suggested by an International Energy Agency (IEA) study [24].

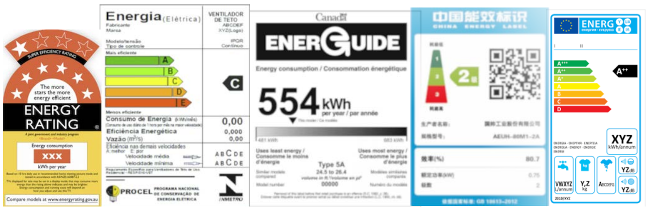


Figure 2: Examples of energy labels from Australia and New Zealand [63], Brazil, Canada [56], China, and European Union [74].

Table 1: Energy labels in different parts of the world and small appliance they cover (Australian and New Zealand [63], Brazil, Canada [56], China, European Union [74], India [58], Japan [45], Jamaica, United States of America [12], Philippines [57] and Singapore [3]).

Country / Unions	TV	Printer	Monitor	PC	Game Console
Australia & New Zealand	+	-	-	-	
Brazil	+	-	-	-	-
Canada	-	-	-	-	-
China	+	+	+	-	-
European Union	+	-	+	+	+
India	+	+	-	+	-
Japan	+	-	-	+	-
Jamaica	-	-	-	-	-
USA	+	-	-	-	-
Philippines	-	-	-	-	-
Singapore	+	-	-	-	-

For detecting events, we leverage changes in current load waves that occur when an appliance switches its state. Four general states exist, which are categorised from type I to type IV [81]: On/Off state (e.g. light bulbs), multi-state (e.g. washing machines), continuously varying (e.g. laptops and mobile phones) and constant energy consumption (e.g. fire alarms and landlines).

4.2 Appliance Data Sets

Different publicly available data sets cover the area of appliance detection, e.g. REDD [32], BLUED [5], WHITED [27], PLAID [13] and IDEAL [61]. Renaux et al. [64] and Ahajjam et al. [4] presented an overview of available non-intrusive load monitoring (NILM) data sets with some of their features and characteristics. Due to the mobility and heterogeneity of small appliances, metadata is sometimes unavailable or hard to track in some data sets. The different model training approaches in this study run on the BLOND data set according to Kriechbaumer et al. [33].

The BLOND data set contains high-frequency data of plug-bars in multiple offices in a university. It comprises of voltage and current measurements over a sequence of 213 and 50 consecutive days with either a frequency of 50 or 250 Hz, respectively. The power plug bars used for the measurements have six slots metered individually. To avoid any interference or overlapping signals between the sockets, all sockets produce an independent current signal by measuring with a Hall-effect-based integrated circuit [33]. A redesigned power socket combined with a Raspberry Pi takes the measurements. Such power sockets are called MEDAL units and a total of 15 of these units are deployed. Please refer to [33] for a detailed description of the data sampling setup. Two example event snippets for an USB charger and a monitor are given in Figure 3. Each event snippet comprises of 25,600 measurements with a sampling rate of 6.4 kHz, resulting in an event window of 4 s. and the meta data for each appliance is available. The measurements for the BLOND data set come from an office environment. However, it covers small appliances such as monitors, laptops and USB chargers, which are also used in residential houses. Therefore, a transferring of our results to a non-office environment is possible.

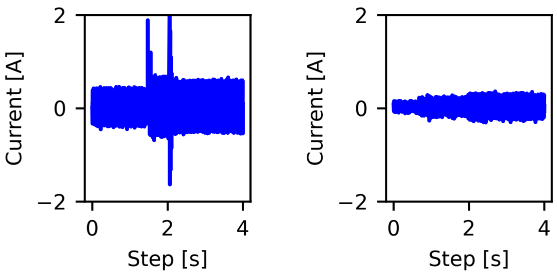


Figure 3: Pre-processed events of an USB charger and a computer monitor with a measurement frequency of 6.4 kHz (25,600 samples). The X-axis shows the time in seconds and the Y-axis shows the current in ampere.

5 DATA PREPARATION

The following subsections explain the annotation of the training data. We explain in detail which events come from which annotation and which category (e.g. USB charger or laptop) they represent.

5.1 Event Annotation

The event snippets used for training are state transition events between an off/on switch either way. Event annotation or event detection is performed manually using MATLAB [28] and Python tools. Both tools have a simple graphical user interface (GUI) to plot the data and zoom in and out. Each electrical phase is plotted separately. After annotating a specific time series, the script stores the start and endpoint of the event in a file. There is no information available on when an event happens. Therefore, the annotator has to go through the time series and look for peaks and changes in the time series of the current waves. F The training data set comes from two annotation runs that are two years apart (Figure 1). Each annotation was performed by a different person. For the first one, the annotator used the MATLAB tool. It covers the time between October, 2016 and December, 2016 resulting in 11,719 event snippets. However, they have a huge bias towards monitors, which make up 75 % of the events. Therefore, a second annotation using the Python GUI focused on obtaining additional events from underrepresented appliances. This led to a total of 1,445 additional events from laptops, dev boards, PCs, kettles, printers, projectors, screen motors, USB-chargers and daylight. However, for the kettle and daylight, only few events are available. The total labeled power of all appliances is 11.2 kW. Daylight contributes with 10 W about 0.1 % to it and is therefore be neglected. The two kettles contribute 3.8 kW, but they are rarely used. Therefore, we removed these two classes of devices from the training and testing data set. Figure 4 shows the distribution of all events from both annotation runs. The majority of the events are laptop and monitor events because almost every uses these devices in offices. Only one kettle exists in the kitchen. The printer and projector also have their dedicated room and are not used as frequently as laptops or monitors.

5.2 Features

The features for training the four ML models belong to either the power or spectral category. Power features are basic or advanced calculations from the electrical engineering domain. Spectral features are often transformed into the frequency space using a Fourier transform, and some of them originate in the audio domain. The power features allow us to classify steady-state appliances (Type I). Additionally, spectral features can help to classify transient-state appliances. A steady state is a time period in which the variables describing the current process do not change. An example is a light bulb. A transient state occurs during the change between two steady states of a process or system. For example, this state occurs in laptops, stoves or workstations (Type II to Type IV).

6 DESIGN AND EXPERIMENTS

The next step after preparing the data is the experimental design. First, we present how a reference model is trained on a central server. This section focuses on the methods we used to identify a good performing model with access to the entire data set. Then, we elaborate on the FL experimental environment. For the FL setups, we alter the model's training process with three different variations.

- No-stopping:
  The training does not stop when the validation loss does not decrease for five consecutive epochs.
- Learning rate scheduler:
  Cosine annealing is initialised with an initial value and the number of training epochs. The learning rate follows a cosine trajectory that reduces the learning rate on each epoch. Compared to the scheduling based on client loss, the advantage of this approach is that cosine annealing creates an equal reducing policy on each client.
- Aggregation strategy:
  Instead of aggregating model parameters after an epoch, we can aggregate them after different batches. For example, an aggregation step of four means we average all model parameters after every fourth batch. This approach has the advantage that all clients perform synchronised training since they have the same batch size on each client. Nasirigerdeh et al. [46] used this approach and found that the FL model achieved comparable performance to a centrally trained model.

6.1 Reference Model

We build a well-performing reference model for each model architecture (CNN, LSTM, ResNet and DenseNet) that has access to the entire data set. Hyperparameter tuning, feature chaining and cross-validation aided in finding the optimal model parameters based on our parameter space. The performance metrics, namely accuracy, F1 score, precision and recall, are to be optimised. Fixed parameters are the batch size, maximum number of epochs and the train/validation/test split which are set to 128, 20 and 80 % / 10 % / 10 %, respectively. Each model uses cross entropy for the loss calculation. The sampler is a weighted random sampler and the scheduler is set to ReduceLROnPlateau (factor=0.1, patience=3).

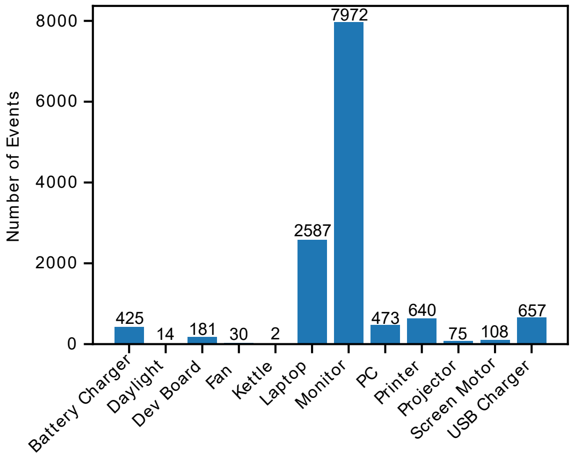


Figure 4: Number of events per appliance in the training data set based on two feature annotation runs.

The baseline CNN uses a simple architecture, which is adopted from the LeNet architecture presented in 1998 by Lecun et al. [34]. This architectural design is extended with activation functions and batch normalization. The numbers behind the model name in Table 3 give the number of blocks and the numbers in brackets give the number of filters/hidden units. The best size of the first unit is obtained via hyperparameter tuning and all consecutive units 1.5 times larger than the previous one. The entire source code with the model architecure will be made open source. For the working principles of LSTM, ResNet and DenseNet model refer to [20], [19] or [22], respectively. Table 2 summarises the performance metrics of the central training for all four model metrics (accuracy, F1-score, precision and recall). The model metrics vary slightly between the experiments and model architectures. We focus on the F1 score as the leading performance metric. It combines the precision and recall metrics and it is a widely used score for evaluating ML classifications.

Four second level parameters are optimised during the hyperparameter tuning: learning rate (lr), weight decay (wd), number of blocks and number of filters/hidden units. Their respective minimum and maximum values are (0.001, 0.1), (0, 0.001), (1, 4) and (10, 30). To reduce the computational overhead we use an optimal latin hypercube based on the enhanced stochastic evolutionary [26]. Optimal latin hypercube is a method to distribute multiple parameter combinations evenly in a multidimensional space. It belongs to the category of design of experiments, and its goal is to uniformly distribute N parameter combinations in an m-dimensional space, where m represents the number of hyperparameters. The best feature set, with respect to the F1 score, is identified with forward feature chaining. Seven features are available. First, each feature is used individually. Then, the best performing feature is combined with each leftover feature and the performance is evaluated again. In theory, this can lead to N! trained models. The results listed in Table 3 show the feature combined with the highest F1 score. These feature sets might change for a non-IID FL setup. However, in our case it is not feasible to run hyperparameter tuning and feature chaining on computationally weak edge devices. To increase the confidence in the results, we run ten-fold cross-validation with the best hyperparameters and feature sets which yield a maximum difference of 1 % in all four performance metrics. The best performing features are always spectral features. Power features rank second or third. Electrical signals are sine curves. By transforming these signals into the frequency domain we can extract additional information.

6.2 Edge Device Testbed

Our edge device testbed comprises of 16 Raspberry Pis model 4B with 2 GB of memory. They are powered over a Netgear Power-over-Ethernet switch. It measures the individual power consumption of all devices. An external power meter measures the total energy consumption of the switch to identify possible deviations of the power readings from the switch's web interface. In addition to the power and energy readings, each module runs a local monitoring script to track different hardware-related metrics. These include measurements of memory, CPU and network with a sampling frequency of 1 Hz. The external and switch power meters have a

Experiment	Accuracy				F1				Precision				Recall			
	CNN	LSTM	ResNet	Dense	CNN	LSTM	ResNet	Dense	CNN	LSTM	ResNet	Dense	CNN	LSTM	ResNet	Dense
Central	.96	.89	.95	.91	.96	.86	.95	.89	.96	.85	.95	.92	.96	.89	.95	.91
IID	.87	.82	.80	.79	.83	.79	.79	.73	.84	.78	.78	.68	.84	.82	.82	.79
Non-IID	.34	.50	.22	.28	.31	.44	.19	.19	.34	.45	.17	.17	.34	.50	.22	.28
Non-IID + [1]	.27	.79	.26	.35	.26	.79	.23	.27	.33	.82	.23	.26	.27	.79	.26	.35
Non-IID + [1, 2]	.39	.81	.33	.35	.33	.80	.31	.27	.40	.81	.32	.26	.39	.81	.33	.35
Non-IID + [1, 2, 3]	.50	.86	.28	.41	.50	.84	.26	.38	.54	.82	.31	.48	.50	.86	.28	.41
Non-IID + [1, 2, 4]	.27	.77	.23	.22	.25	.77	.19	.18	.28	.79	.16	.16	.27	.77	.23	.22
Non-IID + [1, 2, 5]	.29	.70	.28	.34	.26	.68	.25	.31	.27	.74	.25	.40	.29	.70	.28	.34

Table 2: Performance metrics (accuracy, F1-score, precision and recall) for all model architectures and experiments (1 = No-Stopping, 2 = Cosine learning rate, 3 = Aggregation step 1, 4 = Aggregation step 4, 5 = Aggregation step 10).

Model	Optimiser	Features
CNN 4	SGD	MFCC, DCS, AOT
(19, 28, 42, 63)	(lr=0.055, wd=0.0)	
LSTM 1	SGD	MFCC, AC Power
(23)	(lr=0.045, wd=0.001)	
ResNet 4	SGD	MFCC, COT
(20, 30, 45, 67)	(lr=0.052, wd=0.001)	
DenseNet 3	SGD	MFCC, DCS, AOT
(13, 18, 27)	(lr=0.075, wd=0.001)	

Table 3: Overview of the hyperparameters of the four tested model architectures with the best testing results.

measurement frequency of one sample per two minutes. The measured parameters enable the evaluation of the hardware load for each module individually and network traffic caused by a training process. We use Raspberry Pis to simulate edge device, because they have a small form factor, are cheap and computationally weak. Additionally, the BLOND data set is generated with Raspberry Pis. Therefore, we use this device for our experiments.

7 EVALUATION OF EXPERIMENTS

The following two subsections discuss the model and hardware measurements of our experiments. Subsection 7.1 includes a comparison of the four performance metrics (accuracy, precision, recall and F1 score), whereas Subsection 7.2 focuses on network, CPU, energy utilisation and training time during the training process. All experiments were run with four model architectures (CNN, LSTM, ResNet and DenseNet) and either the entire data set, an IID or non-IID data. For the latter we split the entire data set into random subsets. In the non-IID setup each Raspberry Pi only has access to the data from one MEDAL unit.

7.1 Model Metrics

We use centrally trained models that have access to the entire data set for reference. In a best case scenario, the FL models achieve at least the same model performance or are even better than centralised training. During all experiments we calculate the models accuracy, F1 score, precision and recall on the test data set.

The experiments on the virtual machines and Raspberry Pis do not yield the same performance results. It is necessary to use Fourier transformations to calculate the spectral feature (e.g. MFCC). PyTorch implements a short-time Fourier transform for that. However, at the time of writing this paper, this implementation does not work on ARM architectures. Therefore, we used the *Librosa* library on the Raspberry Pis. This results in lower performance metrics for the CNN and LSTM architectures trained on the Raspberry Pi; the LSTM architecture is particularly affected by this library change. Its F1 score in a non-IID FL setting on an ARM device reduces by 35 % points. The performance metrics for the ResNet and DenseNet architecture stay the same. Therefore, ensuring the availability of all required packages and libraries on the target architecture is essential before the model training. Figure 5 shows an overview of our results with the experiments running on virtual machines with an x86 architecture.

The best F1 score is achieved with central training. The models have all data available. For the IID data, we split the entire data set into equally sized subsets. Each subset contains a random selection of the entire data set. This setup represents our first FL experiment. After every epoch we use the FedAvg algorithm to merge all 15 model parameters into one global model. Then, this model is transferred back to each node and the training continues. The validation and test sets are the same for all experiments to allow for a better comparison between different setups. The server performs the evaluation on each aggregation step. The F1 score for the IID setup is 15 % worse than the central training.

The data distribution in our next experiments is non-IID. The data come from 15 MEDAL units. Now, each node only has its independent data for training. In some offices only projectors or kettles are present and others measure the electricity usage of laptops and monitors. Figure 4 shows the overall presence of all appliance categories. They are unevenly distributed among the nodes in the non-IID experiments. This scenario reduces the F1 score drastically by 64 % compared to the IID setup. The LSTM model is least affected by this change. The F1 score for the LSTM model decreases by 44 %. The model training uses early stopping. It stops the training when the validation loss is not decreasing between five consecutive epochs. Without early stopping, the ResNet and DenseNet models perform slightly better and the LSTM model achieves an F1 score as high as in the IID setup. With the cosine annealing learning rate scheduler, the F1 score improves for the CNN and ResNet models.

In the last three experiments we do not aggregate the model parameters after an epoch, but rather after every batch or every fourth

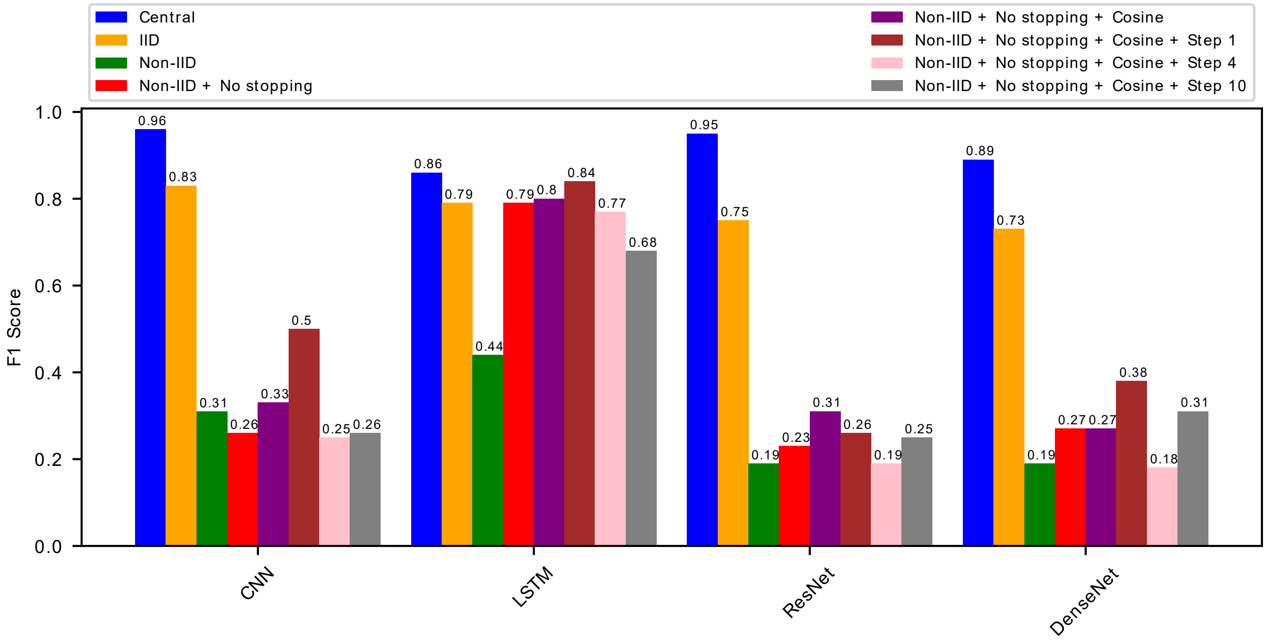


Figure 5: Overview of F1 score for all experiments and model architectures.

and tenth batch, improving the F1 score for CNN and DenseNet by 51 % and 33 %, respectively. The LSTM model achieves an F1 score almost as high as the central training. A decrease in aggregation frequency also decreases the F1 score. Therefore, we recommend using an LSTM model architecture with no early stopping, cosine annealing learning rate and model aggregation after every batch for high-frequency electricity data.

7.2 Hardware Metrics

This subsection describes our findings with respect to the hardware usage of the edge devices. First, we elaborate on why we chose the FL framework *Flower*. Then, we evaluate the network load, CPU usage and training time in detail. Finally, we compare the energy consumption between the central and FL non-IID experiments.

The FL experiments run on an unaltered network. Each module is connected to a switch with a 1 Gbit/s Ethernet cable. The switch itself has four glass fibre cables with 10 Gbit/s each. We use the FL framework *Flower*. This framework uses the gRPC protocol for communication and it only requires to install some Python packages. It allows for an easy setup on ARM architectures such as Raspberry Pis or NVIDIA Jetson Nanos and is generic towards *PyTorch* models. The other FL frameworks have some disadvantages. For example, *PySyft* has a huge overhead due to Docker, a SQLite database and other third party packages, *IBM FL* community version is not available for commercial use and *Clara* from NVIDIA focuses on the healthcare domain.

We compare three scenarios with different data distributions. In the first scenario one Raspberry Pi has the entire data set and trains a single ML model. The other two scenarios have an IID and non-IID setup. For the former we split the entire data set into random subsets. In the non-IID setup each Raspberry Pi only has access to the data from one MEDAL unit.

For the network load we use the transfer of the raw data to a central location as a reference. Therefore, a 0 in the radar chart in Figure 7 represents the transfer of the entire data set of 5.3 GB. This is the case for the central training. The FL experiments reduce the network traffic to 0.2 GB. The network load is reduced by 97 %. Therefore, it gets a score of 97.

We measure the idle time with the CPU pressure stall information (PSI). This metric describes how contended the CPU's resources are. It indicates how long processes had to wait for CPU resources. The higher this number is, the longer processes have to wait and the more overloaded is the CPU. A CPU PSI of below 1000 μs is considered as idle.

The training time for the central setup varies per model. The CNN and the LSTM model architecture need approximately 3 h. The DenseNet needs 1 hour more and the ResNet model architecture takes 4.5 h to train. All these models have different levels of complexities and it is hard to normalize these results. Therefore, a direct comparison is not possible. However, the training time seems to correlate with the models memory usage. The CNN and the LSTM are the smallest models. They require 450 MB and 550 MB, respectively. The DenseNet consumes 600 MB and the ResNet 650 MB. For the non-IID setup this looks differently. The training time for 20 epochs decreases to 0.5 h and 1 h for all model architectures for the IID and non-IID case, respectively. The reduction in training

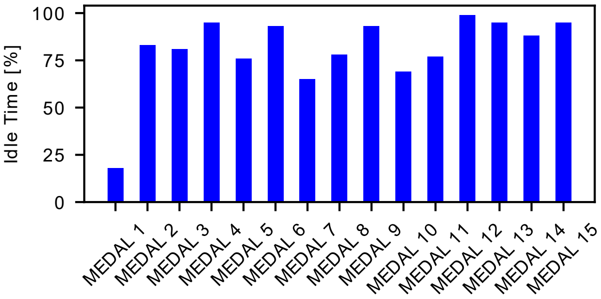


Figure 6: Idle time in percent of an LSTM training with non-IID data.

time is due to the smaller data sets per Raspberry Pi. Additionally, the total compute power is 15x higher. However, due to the non-IID nature of the data set some devices are idle for up to 95 % of the time. Figure 6 shows the idle time of each device / MEDAL unit in percent for a FL training of an LSTM model for 20 epochs. The whole training took 1 hour. To speed up the training process, we could remove the device with the data from the MEDAL 1 unit. This device has 3,618 training samples. The other devices have an average of 684 training samples. To reduce the over all training time we could not consider the first device in the client selection process. However, this potentially reduces the overall model performance.

During our experiments we monitor the energy consumption of all edge devices individually. For the energy evaluation we first subtract the idle power of the edge devices from the measured values during training, which is 3.8 W for the Raspberry Pi with 2 GB memory. The energy consumption for training a CNN, LSTM, DenseNet and ResNet on the entire data set on a Raspberry Pi with 2 GB of memory is 7.63 Wh, 8.03 Wh, 10.25 Wh and 12.45 Wh, respectively. The normalized energy consumption by training time is the same for all model architectures and results to 2.6 Wh. Training each of the four models in a FL setup increases the total energy consumption by up to 37 % for the LSTM. The CNN, DenseNet and ResNet consume 18 %, 12 % and 5 % more energy, respectively. Therefore, it is crucial for practical applications to consider the model architecture and the training time for the energy consumption. Our findings help to find an optimum with respect to training time, energy consumption and level of privacy.

7.3 Scalability

We ran additional experiments to evaluate the influence of the number of clients in a FL setup on the overall training time. Each client runs on a Raspberry Pi 4B and trains an LSTM model on the same training data set. We use the FL framework Flower. Even though all clients always need the same amount of individual training time, the overall training time increases linearly with the number of clients. Our experiment consisted of a total number of 41 clients and we ran an experiment each time after increasing the number of clients by five.

7.4 Summary of all Metrics

This subsection summarizes the findings from Subsection 7.1 and 7.2. We use multiple radar charts to visually compare the performance of different data distributions on model and hardware metrics. The results of the training of the LSTM and the ResNet model architecture are presented in Figure 7. All values range from 0 to 100 where 100 represents the best outcome and the results are normalized to the best score. We assume that FL is completely private and subsequently give all FL based solutions a privacy score of 100, while central training gets a privacy score of 0.

The radar chart shows the trade offs between different metrics. Central training has the best F1 score, but it does not conserve data privacy. Also, it is slow on a computationally weak edge device. For the two FL data distributions, data privacy increases, but the accuracy decreases drastically. This is especially the case for non-IID labels and the ResNet and DenseNet architecture. The training time decreases by a factor of up to 6 for the ResNet model architecture.

FL aims at increasing data privacy. However, in a non-IID case our ML models performance decreases. To achieve similar model performance compared to central training, additional steps are required. In our case these are no-stopping, cosine learning rate and a different aggregation strategy. By applying these steps we found that FL can compete with a centrally trained model in terms of quality while keeping data private.

8 EXAMPLE USE CASE

With a building management system an operator can maintain its infrastructure and optimise its usage. In the following paragraphs we introduce some potential use cases for our appliance classifier in an office environment. These use cases are split into two categories: Energy and safety.

Plug-loads add thermal energy to a room. This energy is considered in the design phase of a building [49, 60]. However, the actual usage might differ from the theoretical value. Knowing which devices are running can help adjusting ventilation and cooling to avoid uncomfortable user experience and uncontrolled, wasteful window opening. Office devices consume electrical energy. With our classifier, the operator can identify appliances that run over night or the weekend and accumulate the wasted energy. The operator can improve energy awareness by showing these numbers to the users.

Office equipment must comply with safety standards; depending on the country, they must be checked frequently. Some examples are the "EN 60950:2002 Safety of IT equipment" in United Kingdom or DGUV V3 in Germany. Running inference of an appliance detection model locally in an office environment allows us to keep track of the actual usage of the equipment. Safety checks based on the equipment's actual run time instead of calendar based checks can reduce maintenance costs.

Besides the mentioned use cases, Radhakrishnan et al. [30] provided further ideas for exploiting information about plug-loads in a building and Hosseini et al. [21] gave an overview on how NILM can help to improve home energy management systems. All these use cases have touching points with legal, technical or economical aspects. Our office appliance classifier can solve technical issues and help realizing the applications discussed above.

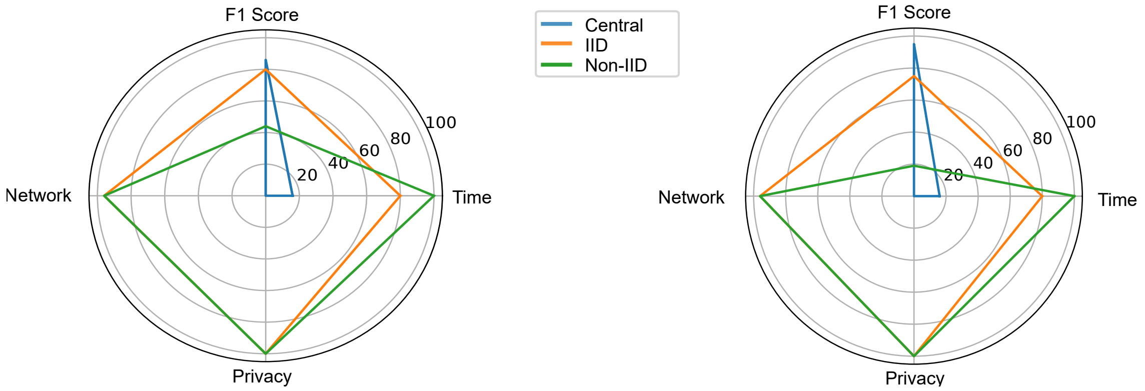


Figure 7: Radar diagram showing a rating of privacy, F1 score, network and training time on a scale from 0 to 100 of an LSTM (left) and ResNet (right) training with three different data distributions (central, IID and non-IID).

9 LEARNED LESSONS

This section gives takeaway messages based on the conducted experiments. First, we highlight our findings with respect to a central model training on a high-frequency energy data set. Then we show our findings with respect to training the same models in an FL system.

(1) Our four developed ML model architectures trained perform well in classifying On/Off states in a high-frequency energy data set with a few features (see Subsection 6.1). These are electrical engineering (AC power) and spectral features (MFCC, AOT, DCS and COT). The equations for all these features are given in the Appendix.
(2) Data distribution (IID or non-IID) in a distributed FL system has a tremendous effect on the model's performance. All performance metrics (accuracy, F1 score, recall and precision) decrease for all four model architectures by at least 50 % when going from a central to non-IID scenario. However, avoiding early stopping in the training or introducing learning rate schedulers and different aggregation strategies help to boost the performance metrics (see Subsection 7.1).
(3) Network traffic in FL experiments decreases compared to transferring only the raw data to a central location. However, this depends on the FL framework and its implementation (see Subsection 7.2).
(4) The training time of all our FL experiments decreases compared to central training by a factor of up to 6 (see Subsection 7.2). This is due to the smaller data set per edge device. The entire training is as fast as the slowest edge device with the largest data set. To avoid drop outs in the training, it is helpful to only choose clients for the training which have a data set of similar size.
(5) Changing from a central training approach to a FL setup increases the energy consumption. Depending on the model architecture the energy consumption increases by up 37 %.

10 CONCLUSION

Energy informatics is an emerging field that tries to solve issues related to energy or operational efficiency. To do so data are required. We developed four model architectures CNN, LSTM, ResNet and DenseNet to leverage high-frequency energy data, which are privacy-sensitive, from an office environment to classify appliances. If all data are centrally stored, an employer could generate profiles of the working staff. However, regulatory frameworks like the GDPR forbid such approaches. FL helps to generate insights with the high-frequency data set, without violating privacy concerns. We ran multiple experiments centrally to identify the best performing model architecture for a data set with high-frequency electricity measurements based on four metrics accuracy, F1 score, precision and recall. To achieve better performance metrics for the FL setups, we applied three different strategies. A cosine learning rate scheduler and the absence of early stopping increased all four performance metrics equally. Additionally, we achieved with a batch wise aggregation strategy even better performance. However, our results show that all FL models perform worse than a central training approach. Just the LSTM model achieves an F1 score close to the one from the central training. The training time and the network load decreased for the FL training compared to a central training. The former decreases due to smaller data sets per client and the network load required for the training process with the `Flower` framework reduces by 97 % when compared to the size of the entire data set of 5.3 GB.

In future work, we will investigate the impact of compression techniques on the network load and the model accuracy to further decrease the network usage. In addition to that, we will alter the network of all or individual edge device to account for a heterogeneous network landscape (e.g., 4G mobile network or instead of glas fiber connection a 15 MB/s DSL connection). We will also look into transfer learning to pre-train a model on a smaller data set to account for the non-IID environment of the BLOND data set.

ACKNOWLEDGMENTS

This paper is funded by the Bavarian Ministry of Economic Affairs, Regional Development and Energy under the REMORA project with the identifier DIK0111/03.

REFERENCES

[1] EIA (U.S. Energy Information Administration). 2020. Annual Energy Outlook 2020. (2020).
https://www.eia.gov/outlooks/aeo/.

[2] U.S. Energy Information Administration. 2021. How much energy is consumed in U.S. buildings? (2021).
https://www.eia.gov/tools/faqs/faq.php?id=86&t=1.

[3] National Environment Agency. 2021. The Energy Label. https://www.nea.gov.sg/our-services/climate-change-energy-efficiency/energy-efficiency/household-sector/the-energy-label. (July 2021).

[4] Mohamed Aymane Ahajjam, Daniel Bonilla Licea, Chaimaa Essayeh, Mounir Ghogho, and Abdellatif Kobbane. 2020. MORED: A Moroccan BuildingsâĂŹ Electricity Consumption Dataset. *Energies* 13, 24 (2020). https://doi.org/10.3390/en13246737

[5] K. Anderson, A. Ocneanu, Diego Benitez, Derrick Carlson, A. Rowe, and M. Berges. 2012. BLUED: A fully labeled public dataset for event-based non-intrusive load monitoring research. *Proceedings of the 2nd KDD Workshop on Data Mining Applications in Sustainability (SustKDD)* (01 2012), 1–5.

[6] Shady Attia, Mohamed Hamdy, and Sherif Ezzeldin. 2017. Twenty-year tracking of lighting savings and power density in the residential sector. *Energy and Buildings* 154 (2017), 113–126. https://doi.org/10.1016/j.enbuild.2017.08.041

[7] Daniel J. Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, Pedro P. B. de GusmÃčo, and Nicholas D. Lane. 2021. Flower: A Friendly Federated Learning Research Framework. (2021). arXiv:cs.LG/2007.14390

[8] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub KoneÄŊnÃ¡, Stefano Mazzocchi, H. Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. 2019. Towards Federated Learning at Scale: System Design. (2019). arXiv:cs.LG/1902.01046

[9] Christopher Briggs, Zhong Fan, and Peter Andras. 2021. Federated Learning for Short-term Residential Energy Demand Forecasting. (2021). arXiv:cs.LG/2105.13325

[10] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub KoneÄŊnÃ¡, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. 2019. LEAF: A Benchmark for Federated Settings. (2019). arXiv:cs.LG/1812.01097

[11] CMSn. 2021. GDPR Enforcement Tracker. https://www.enforcementtracker.com/. (July 2021).

[12] Federal Trade Commission. 2021. Title 16 Commercial Practices. (2021).

[13] Leen De Baets, Mario Berges, Tom Dhaene, Chris Develder, Jingkun Gao, and Dirk Deschrijver. 2020. PLAID 2017. (Jan 2020). https://doi.org/10.6084/m9.figshare.11605215.v1

[14] Sannara Ek, François Portet, Philippe Lalanda, and German Vega. 2020. Evaluation of Federated Learning Aggregation Algorithms Application to Human Activity Recognition. In *UbiComp/ISWC '20: 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and 2020 ACM International Symposium on Wearable Computers*. ACM, Virtual Event Mexico, France, 638–643. https://doi.org/10.1145/3410530.3414321

[15] WeBank AI Group. 2018. Federated Learning White Paper V1.0. (2018).

[16] Yunzhe Guo, Dan Wang, Arun Vishwanath, Cheng Xu, and Qi Li. 2020. Towards Federated Learning for HVAC Analytics: A Measurement Study *(e-Energy '20)*. Association for Computing Machinery, New York, NY, USA, 68âĂŞ73. https://doi.org/10.1145/3396851.3397717

[17] Lukas Haefner. 2018. Demand Side Management. (2018). https://doi.org/10.1365/s40702-017-0363-9

[18] Chaoyang He, Songze Li, Jinhyun So, Xiao Zeng, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, Xinghua Zhu, Jianzong Wang, Li Shen, Peilin Zhao, Yan Kang, Yang Liu, Ramesh Raskar, Qiang Yang, Murali Annavaram, and Salman Avestimehr. 2020. FedML: A Research Library and Benchmark for Federated Machine Learning. (2020). arXiv:cs.LG/2007.13518

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. (2015). https://doi.org/10.48550/ARXIV.1512.03385

[20] Sepp Hochreiter and JÃijrgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (11 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735 arXiv:https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf

[21] Sayed Saeed Hosseini, Kodjo Agbossou, Sousso Kelouwani, and Alben Cardenas. 2017. Non-intrusive load monitoring through home energy management systems: A comprehensive review. *Renewable and Sustainable Energy Reviews* 79 (2017), 1266–1274. https://doi.org/10.1016/j.rser.2017.05.096

[22] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2016. Densely Connected Convolutional Networks. (2016). https://doi.org/10.48550/ARXIV.1608.06993

[23] IEA. 2020. Lighting. (2020).
https://www.iea.org/reports/lighting.

[24] IEA. 2021. Appliances and Equipment. (2021).
https://www.iea.org/reports/appliances-and-equipment.

[25] Aashish Kumar Jain, Syed Shahbaaz Ahmed, Prahalathan Sundaramoorthy, Raghavendran Thiruvengadam, and Vineeth Vijayaraghavan. 2017. Current peak based device classification in NILM on a low-cost embedded platform using extra-trees. In *2017 IEEE MIT Undergraduate Research Technology Conference (URTC)*. 1–4. https://doi.org/10.1109/URTC.2017.8284200

[26] Ruichen Jin, Wei Chen, and Agus Sudjianto. 2005. An efficient algorithm for constructing optimal design of computer experiments. *Journal of Statistical Planning and Inference* 134, 1 (2005), 268–287. https://doi.org/10.1016/j.jspi.2004.02.014

[27] Matthias Kahl, Anwar Haq, Thomas Kriechbaumer, and Hans-Arno Jacobsen. 2016. WHITED - A Worldwide Household and Industry Transient Energy Data Set.

[28] Matthias Kahl, Thomas Kriechbaumer, Daniel Jorde, Anwar Ul Haq, and Hans-Arno Jacobsen. 2019. Appliance Event Detection - A Multivariate, Supervised Classification Approach. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems (e-Energy '19)*. Association for Computing Machinery, New York, NY, USA, 373âĂŞ375. https://doi.org/10.1145/3307772.3330155

[29] Matthias Kahl, Anwar Ul Haq, Thomas Kriechbaumer, and Hans-Arno Jacobsen. 2017. A Comprehensive Feature Study for Appliance Recognition on High Frequency Energy Data. In *Proceedings of the Eighth International Conference on Future Energy Systems (e-Energy '17)*. Association for Computing Machinery, New York, NY, USA, 121âĂŞ131. https://doi.org/10.1145/3077839.3077845

[30] Krishnanand Kaippilly Radhakrishnan, Hoang Duc Chinh, Manish Gupta, Sanjib Kumar Panda, and Costas J. Spanos. 2020. Context-Aware Plug-Load Identification Toward Enhanced Energy Efficiency in the Built Environment. *IEEE Transactions on Industry Applications* 56, 6 (2020), 6781–6791. https://doi.org/10.1109/TIA.2020.3016621

[31] Peter Kairouz, H. Brendan McMahan, Brendan Avent, AurÃľlien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, AdriÃă GascÃşn, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub KoneÄŊnÃ¡, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, TancrÃĺde Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Oezguer, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian TramÃĺr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. 2021. Advances and Open Problems in Federated Learning. (2021). arXiv:cs.LG/1912.04977

[32] J Kolter and Matthew Johnson. 2011. REDD: A Public Data Set for Energy Disaggregation Research. *Artif. Intell.* 25 (01 2011).

[33] Thomas Kriechbaumer and Hans-Arno Jacobsen. 2021. BLOND, a building-level office environment dataset of typical electrical appliances. https://dataserv.ub.tum.de/index.php/s/m1375836. (2021).

[34] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324. https://doi.org/10.1109/5.726791

[35] Sangyoon Lee and Dae-Hyun Choi. 2022. Federated Reinforcement Learning for Energy Management of Multiple Smart Homes With Distributed Energy Resources. *IEEE Transactions on Industrial Informatics* 18, 1 (2022), 488–497. https://doi.org/10.1109/TII.2020.3035451

[36] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated Optimization in Heterogeneous Networks. (2020). arXiv:cs.LG/1812.06127

[37] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smithy. 2019. FedDANE: A Federated Newton-Type Method. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. 1227–1231. https://doi.org/10.1109/IEEECONF44664.2019.9049023

[38] Yi Liu, James J. Q. Yu, Jiawen Kang, Dusit Niyato, and Shuyu Zhang. 2020. Privacy-Preserving Traffic Flow Prediction: A Federated Learning Approach. *IEEE Internet of Things Journal* 7, 8 (2020), 7751–7763. https://doi.org/10.1109/JIOT.2020.2991401

[39] Heiko Ludwig, Nathalie Baracaldo, Gegi Thomas, Yi Zhou, Ali Anwar, Shashank Rajamoni, Yuya Ong, Jayaram Radhakrishnan, Ashish Verma, Mathieu Sinn, Mark Purcell, Ambrish Rawat, Tran Minh, Naoise Holohan, Supriyo Chakraborty, Shalisha Whitherspoon, Dean Steuer, Laura Wynter, Hifaz Hassan, Sean Laguna, Mikhail Yurochkin, Mayank Agarwal, Ebube Chuba, and Annie Abay. 2020.

IBM Federated Learning: an Enterprise Framework White Paper V0.1. (2020). arXiv:cs.LG/2007.10987

[40] Ardeshir Mahdavi, Farhang Tahmasebi, and Mine Kayalar. 2016. Prediction of plug loads in office buildings: Simplified and probabilistic methods. *Energy and Buildings* 129 (2016), 322–329. https://doi.org/10.1016/j.enbuild.2016.08.022

[41] D. Mariano-HernÃąndez, L. HernÃąndez-Callejo, A. Zorita-Lamadrid, O. Duque-PÃľrez, and F. Santos GarcÃŋa. 2021. A review of strategies for building energy management system: Model predictive control, demand side management, optimization, and fault detect diagnosis. *Journal of Building Engineering* 33 (2021), 101692. https://doi.org/10.1016/j.jobe.2020.101692

[42] Rodrigo Martins, Holger C. Hesse, Johanna Jungbauer, Thomas Vorbuchner, and Petr Musilek. 2018. Optimal Component Sizing for Peak Shaving in Battery Energy Storage System for Industrial Applications. *Energies* 11, 8 (2018). https://doi.org/10.3390/en11082048

[43] Kurtis McKenny, Matthew Guernsey, Ratcharit Ponoum, and JEff Rosenfeld. 2008. Commercial Miscellaneous Electric Loads: Energy Consumption Characterization and Savings Potential in 2008 by Building Type. (2008).

[44] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agueera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. (2017). arXiv:cs.LG/1602.05629

[45] Trade Ministry of Economic and Industry. 2021. Energy Efficiency and Conservation. https://www.meti.go.jp/english/policy/energy_environment/energy_efficiency/index.html. (July 2021).

[46] Reza Nasirigerdeh, Mohammad Bakhtiari, Reihaneh Torkzadehmahani, Amirhossein Bayat, Markus List, David B. Blumenthal, and Jan Baumbach. 2021. Federated Multi-Mini-Batch: An Efficient Training Approach to Federated Learning in Non-IID Environments. (2021). arXiv:cs.LG/2011.07006

[47] United Nations. 2015. Paris Agreement. (2015).

[48] Deutsches Institut Fur Normung. 2003. DIN 4108-6:2003-01 Thermal protection and energy economy in buildings - Part 6: Calculation of annual heat and energy use. (2003).

[49] Deutsches Institut Fur Normung. 2010. DIN V 18599 Beiblatt 1:2010-01 Energy efficiency of buildings - Calculation of the net, final and primary energy demand for heating, cooling, ventilation, domestic hot water and lighting - Supplement 1: Balancing of demand and consumption. (2010).

[50] NREL. 2020. Assessing and Reducing Plug and Process Loads in Office Buildings. (2020).

[51] NREL. 2020. Assessing and Reducing Plug and Process Loads in Retail Buildings. (2020).

[52] Odysse-Mure. 2018. Sectoral Profile - Households. (2018). https://www.odyssee-mure.eu/publications/efficiency-by-sector/households/heating-consumption-per-m2.html.

[53] OECD/IPEEC. 2019. Building Energy Performance Gap Issues - An Intnernational Review. (December 2019).

[54] European Court of Auditors. 2020. Energy efficiency in buildings: greater focus on cost-effectiveness still needed. (2020).

[55] State of California Department of Justice. 2018. California Consumer Privacy Act of 2018 [1798.100 - 1798.199.100]. (2018).

[56] Government of Canada. 2021. The EnerGuide label. https://www.nrcan.gc.ca/energy-efficiency/energuide-canada/energuide-label/13609. (July 2021).

[57] Department of Energy. 2017. Energy Labeling Efficiency Standards as of June 2017. https://www.doe.gov.ph/energy-labelling-efficiency-standards?page=6&withshield=1. (July 2017).

[58] Bureau of Energy Efficiency. 2021. Standards & Labeling. https://beeindia.gov.in/content/standards-labeling. (July 2021).

[59] Council of European Union. 2017. Directive 2009/125/EC and Regulation (EU) 2017/1369. (2017). https://ec.europa.eu/growth/single-market/european-standards/harmonised-standards/ecodesign_en.

[60] Official Journal of the European Union. 2018. DIRECTIVE (EU) 2018/844 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 30 May 2018 amending Directive 2010/31/EU on the energy performance of buildings and Directive 2012/27/EU on energy efficiency. (2018).

[61] Martin Pullinger, Jonathan Kilgour, Nigel Goddard, Niklas Berliner, Lynda Webb, Myroslava Dzikovska, Heather Lovell, Janek Mann, Charles Sutton, Janette Webb, and Mingjun Zhong. 2020. The IDEAL household energy dataset, electricity, gas, contextual sensor data and survey data for 255 UK homes. https://doi.org/10.1038/s41597-021-00921-y

[62] Raghunath Reddy, Niranjan Keesara, Vishal Garg, and Vikram Pudi. 2017. Plug Load Identification Using Regression Based Nearest Neighbor Classifier. In *Proceedings of the Eighth International Conference on Future Energy Systems (e-Energy '17)*. Association for Computing Machinery, New York, NY, USA, 101âĂŞ110. https://doi.org/10.1145/3077839.3077853

[63] GEMS Regulator. 2021. Energy Rating Label. https://www.energyrating.gov.au/. (July 2021).

[64] Douglas Paulo Bertrand Renaux, Fabiana Pottker, Hellen Cristina Ancelmo, AndrÃľ Eugenio Lazzaretti, Carlos Raiumundo Erig Lima, Robson Ribeiro Linhares, Elder Oroski, Lucas da Silva Nolasco, Lucas Tokarski Lima, Bruna Machado Mulinari, JosÃľ Reinaldo Lopes da Silva, JÃžlio Shigeaki Omori, and Rodrigo Braun dos Santos. 2020. A Dataset for Non-Intrusive Load Monitoring: Design and Implementation. *Energies* 13, 20 (2020). https://doi.org/10.3390/en13205371

[65] Arkadiusz Rudzki, Zuzsanna Paciorkiewicz, and Tomasz Augustyniak. 2015. Energy Consumption in Office Buildings: a Comparative Study. (2015). https://www.oswbz.org/wp-content/uploads/2017/03/ENERGY-CONSUMPTION-IN-OFFICE-BUILDINGS.pdf.

[66] Arkadiusz Rudzki, Zuzsanna Paciorkiewicz, and Tomasz Augustyniak. 2021. An open-source federated learning framework. (2021). https://fedbiomed.gitlabpages.inria.fr/.

[67] Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, and Jonathan Passerat-Palmbach. 2018. A generic framework for privacy preserving deep learning. (2018). arXiv:cs.LG/1811.04017

[68] Afaf TaÃŕk and Soumaya Cherkaoui. 2020. Electrical Load Forecasting Using Edge Computing and Federated Learning. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. 1–6. https://doi.org/10.1109/ICC40277.2020.9148937

[69] Susanna Tenhunen. 2021. Energy Performance of Buildings Directive 2010/31/EU: Fit for 55 revision - Implementation in action. (2021). https://doi.org/10.2861/05615

[70] Jesse Tetreault, Rahul Choudhury, Brad Genereaux, Kristopher Kersten, and Jiahui Guan. 2020. Scalable and Modular AI Deployment Powered by NVIDIA Clara Deploy White Paper. (2020).

[71] Ye Lin Tun, Kyi Thar, Chu Myaet Thwal, and Choong Seon Hong. 2021. Federated Learning based Energy Demand Prediction with Clustered Aggregation. In *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*. 164–167. https://doi.org/10.1109/BigComp51126.2021.00039

[72] European Union. 2016. 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). (2016).

[73] European Union. 2019. The European Green Deal - Communication From the Commission to the European Parliament, the European Council, the Council, the European Economic and Social Committee and the Committee of the Regions. (2019).

[74] European Union. 2021. Energy Label. https://europa.eu/youreurope/business/product-requirements/labels-markings/energy-labels/index_en.htm. (July 2021).

[75] Signe Waechter, Bernadette Suetterlin, and Michael Siegrist. 2015. The misleading effect of energy efficiency information on perceived energy friendliness of electric goods. *Journal of Cleaner Production* 93 (2015), 193–202. https://doi.org/10.1016/j.jclepro.2015.01.011

[76] Haijin Wang, Caomingzhe Si, and Junhua Zhao. 2021. A Federated Learning Framework for Non-Intrusive Load Monitoring. (2021). https://doi.org/10.48550/ARXIV.2104.01618

[77] Haijin Wang, Caomingzhe Si, Junhua Zhao, Guolong Liu, and Fushuan Wen. 2021. Fed-NILM: A Federated Learning-based Non-Intrusive Load Monitoring Method for Privacy-Protection. (2021). arXiv:cs.LG/2105.11085

[78] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. 2020. Federated Learning with Matched Averaging. In *International Conference on Learning Representations*. https://openreview.net/forum?id=BkluqlSFDS

[79] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu. 2019. *Federated Learning*. Morgan & Claypool Publishers. https://books.google.de/books?id=JdPGDwAAQBAJ

[80] Wenqi Yang, Yang Zhang, Wei Yang Bryan Lim, Zehui Xiong, Yutao Jiao, and Jiangming Jin. 2020. Privacy is not Free: Energy-Aware Federated Learning for Mobile and Edge Intelligence. In *2020 International Conference on Wireless Communications and Signal Processing (WCSP)*. 233–238. https://doi.org/10.1109/WCSP49889.2020.9299703

[81] Ahmed Zoha, Alexander Gluhak, Muhammad Ali Imran, and Sutharshan Rajasegarar. 2012. Non-Intrusive Load Monitoring Approaches for Disaggregated Energy Sensing: A Survey. *Sensors* 12, 12 (2012), 16838–16866. https://doi.org/10.3390/s121216838

A FEATURE EQUATIONS

The following equations show how to calculate the feature we used during our training processes. The variables V and I are the voltage and current, respectively. The power factor is described by cos(phi). The AC Power feature represents four features. These are the active (P), reactive (Q), apparent (S) and distortion power (D).

$$P = V_{rms} \times I_{rms} \times cos(\phi) \tag{1}$$

$$Q = V_{rms} \times I_{rms} \times sin(\phi) \quad (2)$$

$$S = V_{rms} \times I_{rms} \quad (3)$$

$$D = \sqrt{S^2 - P^2 - Q^2} \quad (4)$$

Mel-Frequency Cepstrum Coefficients (MFCC)

$$MFCC = \sum_{k=1}^{K} log_k \times cos[n(k - 0.5)\frac{\pi}{K}], \forall n \in 1, ..., N \quad (5)$$

Device Current Signature (DCS) The DCS is a feature for appliance classification presented by Jian et al. [25].

Current Over Time (COT)

$$COT = [I_1, I_2, ..., I_n], \forall I_n \in C \quad (6)$$

Admittance Over Time (AOT) is a power feature presented by Kahl et al. [29].

$$AOT = [\frac{I_1}{U_1}, \frac{I_2}{U_2}, ..., \frac{I_n}{U_n}], \forall I_n, U_n \in C \quad (7)$$

Appendix B

Energy vs Privacy: Estimating the Ecological Impact of Federated Learning

Energy vs Privacy: Estimating the Ecological Impact of Federated Learning

René Schwermer
rene.schwermer@tum.de
Technical University of Munich
Germany

Ruben Mayer
ruben.mayer@uni-bayreuth.de
University of Bayreuth
Germany

Hans-Arno Jacobsen
jacobsen@eecg.toronto.edu
University of Toronto
Canada

ABSTRACT

The increasing usage of edge devices and stricter data privacy regulations motivate the use of federated learning (FL). At the same time, more and more stakeholders are concerned about the ecological impact of machine learning and its associate network traffic. The current research in FL does not investigate the impact of different network constraints and privacy-enhancing techniques, such as differential privacy, on the network traffic and energy consumption of the clients. Most experiments run either on virtual machines or on one machine with simulated clients. In such environments, it is challenging to measure each client's network and energy usage. Therefore, we built our "Distributed Edge Device Testbed" (DEDT) and evaluate a convolutional neural network trained on the MNIST data set under different network constraints on DEDT, with differential privacy and with an increasing amount of participating clients. For each experiment, we quantify the network traffic, energy consumption, and training time. The results show the importance of experiments on physically separated nodes and the need to improve software-based power monitoring. The estimated energy consumption deviates by up to 35 % from the measured ones. The accuracy of the estimated network traffic depends on the monitored network interface and gives an error of 18 % for virtual machines in combination with monitoring the Ethernet interface. The training time also increases linearly with the number of participating clients.

CCS CONCEPTS

• **Computing methodologies → Machine learning**; • **Security and privacy → Privacy-preserving protocols**.

KEYWORDS

Federated Learning, Distributed Systems, Energy

ACM Reference Format:
René Schwermer, Ruben Mayer, and Hans-Arno Jacobsen. 2023. Energy vs Privacy: Estimating the Ecological Impact of Federated Learning. In *The 14th ACM International Conference on Future Energy Systems (e-Energy '23), June 20–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3575813.3597344

1 INTRODUCTION

The increase in regulatory constraints such as the General Data Protection Regulation in the European Union [31], the California Consumer Privacy Act in the USA [18], and the Personal Data Protection (Amendment) Act in Singapore [27] increase pressure on companies and other stakeholders working with sensitive data. However, current artificial intelligence (AI) models require big data [19, 25]. Storing all data centrally leads to an increase in administrative overhead to comply with these new regulations and it falls short of preserving data privacy [1, 6]. Another challenge is the increasing amount of internet traffic and its associated energy consumption for the data transmission network. The former increased from 2015 to 2021 by 440 % and the latter also increased in the same time frame by up to 60 % from 220 TWh to 340 TWh [9]. The majority of the internet traffic is video streaming. However, these numbers illustrate the importance of decreasing the network traffic of all applications to reduce the energy footprint of the data transmission network. Federated learning (FL) can be a solution to this problem by increasing data privacy and decreasing network traffic of data science applications.

Most FL experiments in the literature run either on virtual machines (VM), on simulated nodes on one machine, or the platform is not mentioned. However, data transfer between multiple VMs over shared memory is much faster and less error prone than data transfer over the internet (wired or mobile network). Furthermore, it is hard to pinpoint parts of the energy consumption of a server to its hosted VMs, so that a detailed analysis of energy consumption becomes difficult. One approach to solving the latter is PowerAPI, which currently only works on x86 architectures [3]. Therefore, we built the Distributed Edge Device Testbed (DEDT) to evaluate FL experiments on edge devices under real-world conditions to investigate the impact of different network constraints and additional privacy-enhancing techniques such as differential privacy (DP) on hardware performance. For a detailed description of the testbed components see Appendix C.

Our contributions are:

(1) Quantifying the impact of different emulated networks and privacy-enhancing techniques like DP on the energy consumption.
(2) Insights and guidelines on how accurate estimated network traffic and energy consumption are compared to measured metrics on physically separated IoT devices in an FL context.
(3) Quantifying changes in the systems behavior when scaling the number of participating clients. Those include network traffic, energy consumption, and training time.

This paper is structured as follows. First, we describe how FL works and which challenges it faces in Section 2 and we give an

e-Energy '23, June 20–23, 2023, Orlando, FL, USA
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0032-3/23/06…$15.00
https://doi.org/10.1145/3575813.3597344

overview of different FL systems. Section 3 shows how we design our experiments. Section 4 presents the performance and usability of DEDT with different FL experiments. We evaluate our FL experiments and compare the measured metrics with the estimated ones. A summary of the lessons learned is provided in Section 5, and Section 6 presents the conclusion of our findings.

2 FEDERATED LEARNING

The current prominent ML approach is to transfer raw data from remote machines to a central location to further process it. In FL no raw data is transferred. Instead, each remote client receives the instructions on what to run, e.g. train an ML model, and it only sends its local ML model parameters/gradients or other statistical metrics to a central location for further processing. During the entire process, the data scientist at the central server has no access to the raw data. However, FL systems face a multitude of challenges. Those are mainly due to an uneven label distribution on the participating clients, network constraints or risks of privacy attacks. In this paper we focus on challenges with respect to hardware and network utilization.

We qualitatively describe privacy by tackling two prominent problems in an information flow: Copy and bundling problem. The former describes the risk of shared data being unlimitedly duplicated [20, 29]. The latter describes the unintentional sharing of information contained in a data set. FL tries to solve the copy problem and avoids sending raw data at a higher cost of energy due to a larger pool of distributed devices compared to central training. DP tackles the bundling problem and tries to keep the input of a process private by artificially adding noise to the raw data. However, it is computationally heavy. Different aggregation algorithms can increase ML performance. We evaluate the impact of FedAvg [16], QFedAvg [14], FedAdam [24] and FedAvgM [8] on the hardware behavior. Some challenges of FL systems can be investigated purely on one device and others need additional hardware to better mimic the systems behaviour. However, coordinating and maintaining multiple devices increase the organizational overhead.

The current focus in FL research is on improving model accuracy or training time [15, 28, 38]. In most cases it is not clear which hardware environment was used for the experiments. Bousbiat et al. [4] identified a lack of evaluation of communication overhead and required processing resources in FL applications. Additionally, Table 1 gives an overview on which environments are used for FL experiments. Current research focuses on virtual FL nodes for simulating a distributed system. This makes it hard to evaluate the effect of network communication induced by the FL training and to measure energy consumption on a client level. Bonawitz et. al [2] run an FL system on mobile phones. However, they do not give any measurements for energy and network usage. Therefore, this paper illustrates the capabilities of FL in a real-world use case without quantifying its ecological footprint. The goal of this overview is to highlight the lack of FL experiments on physically distributed devices and not to emphasize that all FL experiments should run in such an environment. With our work we enable other researchers to estimate their systems network traffic and, depending on the hardware, also their energy consumption, independently of their desired environment.

Table 1: Overview of environments used for different FL experiments.

FL Environment / Hardware	References
Single device	[15, 28, 32–34]
Virtual Machines	[5, 26]
Physically distributed devices	[2]*
Hybrid	-
Unknown	[7, 13, 15, 23, 30]

Some researches developed optimization problems to balance energy consumption depending on completion time or wireless communication [35, 37]. The given constraints (e.g. fixed CPU frequency and data transmission size) limit this approach to certain edge cases. Additionally, Qui et. al [21, 22] ran FL experiments with different data sets and aggregation strategies on multiple graphic cards and measured the energy consumption. The goal was to quantify the required energy to achieve certain ML model accuracies. The energy consumption of the FL setups was up to 11x higher when compared to a central ML approach. Our experiments follow a similar approach but differ in hardware (48 CPU edge devices instead of a few GPUs). We also introduce DP and monitor the network traffic. Qui et. al converted the resulting energy in Watt seconds to CO2 equivalents. In our experiments we focus on the energy usage in Watt seconds and do not give any CO2 equivalents, because the latter highly depends on the local energy mix. The power consumption of FL on Raspberry Pis might vary from that on other platforms. However, Raspberry Pis are common hardware for IoT applications and Ubuntu is a widely adopted OS. Therefore, our testbed is a feasible reference for real-world scenarios.

3 DESIGN OF EXPERIMENTS

We organize our experiments into three categories: FL extension, network and environment. Each category has multiple options and each experiment consists of one pick per category. FL extensions are either with or without DP. To enable DP we use the `Opacus` Python package with its default settings and secure mode off [36]. The network category consists of a pure network (no artificial changes, which translate to 1 Gbit/s) and three networks emulated with `netEm` (15 Mbit/s and 8 Mbit/s Ethernet network [17] and 4G mobile [12]). The environment category is either a single device, VMs or multiple devices.

In our experiments we use two types of data distributions. First, all clients have the same data set available. Second, each client samples the entire data set based on a Gaussian distribution along a random number between zero and nine. Furthermore, a round refers to the moment when all clients send their updates to the aggregation server after they trained for one epoch. The Power-over-Ethernet (PoE) switch delivers the ground truth for the power measurements. We focus on quantifying the energy consumption of the entire system (edge device) and not of specific software artifacts. All experiments run with a different number of participating clients (2, 5, 10, 20 and 40) to evaluate the scalability of FL. Central ML training with and without DP is our reference scenario.

Table 2: Overview of the energy consumption in Ws/round for an FL training with and without DP, FedAvg and clients 2 GB of memory without idle load. E_{Client} and E_{System} provide the energy consumption of one client and the entire FL system, respectively.

Nr. of Clients	DP	E_{Server} [Ws/round]	E_{Client} [Ws/round]	E_{System} [Ws/round]
Central	-	-	-	2841
2	-	432	1757	3946
5	-	451	1727	9086
10	-	516	1943	19,946
20	-	1203	2230	45,803
40	-	2750	1928	79,870
47	-	4081	2056	100,713
Central	+	-	-	4380
2	+	1083	4594	10,271
5	+	1121	4585	24,046
10	+	2164	5501	57,174
20	+	4096	5485	113,796
40	+	7432	5177	214,512
47	+	10,457	4849	238,360

4 EVALUATION

For the power and network metrics, we compare the measurements from the testbed with an estimator. We use the following methods to estimate the power and network usage in an FL context.

(1) Power: Software-based power estimator based on CPU cycles.
(2) Network: Monitoring local loopback network interface on one Raspberry Pi hosting multiple simulated FL clients or monitoring the Ethernet interface for the VM environment.

4.1 Energy Metrics

The PoE switch and the software-based power estimator return the power measurements and estimations in milli Watt (mW). To calculate the energy consumption, we use the time difference between two readings in seconds. The resulting energy unit for evaluation is Watt seconds (Ws). We normalize all results by the number of training rounds giving us Ws/round.

First, we run reference scenarios with one device hosting the entire data set and training only one central model. This approach consumes 6075 Ws/round and 6690 Ws/round for the Raspberry Pi model with 2 GB and 4 GB, respectively. This includes the energy consumption of the entire device. With DP, the energy consumption per round increases to 9925 Ws/round and 10,004 Ws/round, respectively. Therefore, increasing data privacy results in about 56 % more energy consumption. The energy consumption is not dependant on the network constraints.

To estimate the energy $E_{Client,FL}$ consumed only by the FL process, we deduct the power drawn of the devices in idle mode from the power drawn during FL training. For example, a Raspberry Pi with 2 GB of memory has an idle load of 3.3 W. In that case the central training consumes 2841 Ws/round. Table 2 provides an overview of the energy consumption per client and round for different FL systems. The numbers exclude the idle load. Depending on the number of participating clients the energy consumption $E_{Client,FL}$ varies by up to 25 %, and with DP it increases by up to a factor of 2.6.

The servers' energy consumption per aggregation strategy in Ws/round for two clients without DP from high to low is FedAvg (564), QFedAvg (421), FedAvgM (394) and FedAdam (378). It scales linearly with the number of clients for all aggregation strategies. The clients' energy consumption is independent of the server aggregation algorithm. Lastly, a Gaussian label distribution on client side reduces the average client energy consumption by 30 % and by 20 % for the server due to shorter training.

To put our findings into perspective, we compare them to the battery capacity of current high-end mobile phones by converting Ws into milli Ampere hours (mAh). We consider the evaluation from above with two clients. Without and with DP this translates to 1753 Ws/round and 4544 Ws/round per client, respectively. Table 3 summarizes the comparison. To avoid negative user experience due to FL training, Bonawitz et al. [2] trained their models only when the mobile phone was connected to a power source and it was idle.

Table 3: Simplified examples to get an idea of the training cost with and without DP incurred on a state-of-the-art mobile phone.

Mobile Phone	Battery	FL Rounds	
	[mAh]	No DP	DP
iPhone14	3279	312	123
Galaxy S23 Ultra	5000	490	188

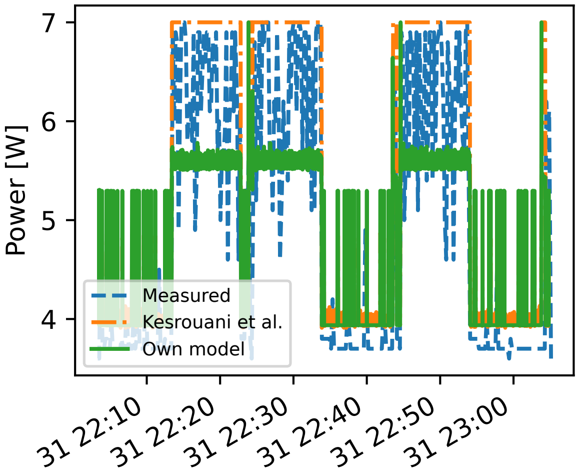


Figure 1: Three different power metrics for one client. The ground truth are the measured power readings. The values for the other two readings are calculated and are based on the Raspberry Pis CPU utilization. All calculated power metrics are capped at 7 W.

It is not always practical to directly measure the power load of an edge device. Therefore, we compare our measurements with two software-based power meters to show how accurate they are and to give guidelines on when to use them. For our estimations, we capped all calculated values at the highest measured value of 7 W. We assume that the majority of users use their edge devices without overclocking them.

Both Raspberry Pi power models mentioned in Appendix A vary in their prediction performance. In general, the Pi model I over-predicts the actual power drawn and the Pi model II under-predicts it at high loads. An example for those relations is shown in Figure 1. The highest error for the Pi model I and II are +35 % and -20 % for runs with DP, respectively. The performance of the power models is independent of the network. However, it depends on the total amount of available memory. The maximum error without DP is +12 % and -20% for the Pi model I and model II, respectively. The lower the CPU load, the better the prediction performance of the Pi model II. This is especially the case when more than 10 clients are involved in FL training, because each client has to wait longer for a round to finish which reduces the CPU load. We explain the relation between number of clients and training time in more detail in Section 4.3.

4.2 Network Metrics

We use `tshark` to monitor each devices individual network traffic. For the experiments with physically distributed clients and with multiple simulated FL clients on one machine we monitor the Ethernet interface (eth0) and the local loopback interface (lo), respectively. This interface is a message channel with only one end point. The sender and receiver are identical in the loopback. The evaluation only considers traffic between a client and the server by filtering based on the respective IP addresses and port. We evaluate the network traffic size and the number of messages.

For the clients, the network traffic (24 MB/round) and the number of messages per round (15,000 messages/round) for all emulated networks are constant, independent of DP. They are 30 % higher than for the pure setup. The number of messages per round for the pure network reduces to 13,000. For a pure network the number of large messages (21 % of all messages are 2962 byte) is higher when compared to the emulated 4G network (11 % 2962 byte).The aggregation strategy has an impact on network traffic. For two clients and FedAvg, the server's traffic per round is 36 MB. For QFedAvg and FedAdam the traffic changes by +16 % and 55 %, respectively, while for FedAvgM it stays the same.

While running FL on the VMs, we monitor the eth0 Ethernet interface. For up to ten clients, the measured network traffic is comparable to the one on the physically distributed devices. The network traffic per client and round on the Raspberry Pis is 23 MB/round. For two, five and ten clients running on VMs the measured network traffic per round and client is almost the same with 19 MB/round, 17 MB/round and 17 MB/round, respectively. However, the number of messages between the VMs is much smaller (about 94 % less messages when compared to the distributed training on the Raspberry Pis). Therefore, monitoring the Ethernet interface on the VMs gives a good estimate of the actual expected network traffic in terms of bandwidth, but not in terms of number of messages.

We run experiments with two simulated clients and all four network settings (pure, 15 Mbit/s, 8 Mbit/s and 4G) with and without DP and monitor the lo interface. The results show a clear mismatch between the monitored network traffic between devices and the monitored traffic on the local loopback interface. The average network traffic is 6 MB/round and the number of messages is 500. The lo interface captures only 26 % of the network traffic and 4 % of the messages measured in fully distributed setup.

4.3 Time and Scalability

Our testbed consists of 48 Raspberry Pis and is therefore limited to represent large scale FL applications with multiple hundred or even thousands of devices. Still, with our experiments, we provide a starting point and identify some trends with respect to training time and energy consumption. Each aggregation step on the server takes 30 s/round, independently of DP running on the clients or not. The training time on the clients increases with DP from 600 s/round to 1600 s/round. The training time is independent of the network profile. The training time per round increases linearly with the number of participating clients even though all clients have the same data set and the same hardware. This is due to the clients training out of sync after every batch of eight clients. Every 8th new batch introduces a training delay of 85 sec.

5 LESSONS LEARNED

In this section, we summarize some lessons learned:

(1) Differential privacy running on the clients increases the training time and the consumed energy by 300 % and 280 %, respectively.
(2) Training time per round increases linearly with the number of participating clients for the `Flower` framework. The same applies to the servers' energy consumption regardless of the aggregation strategy.
(3) Depending on the software-based power model and the number of participating clients the calculated energy consumption can be off by up to +35 %. In any case, the underlying power model should cap all estimations at the respective maximum power load of the device.
(4) Using the Ethernet interface to measure network traffic between virtual machines gives a good estimate of the expected network bandwidth consumption, but underpredicts the number of messages by 94 %.
(5) Using the local loopback interface to measure network traffic on one machine hosting multiple simulated clients does not give a good estimate for the actual network traffic.

6 CONCLUSION

The measurements obtained from our FL experiments on DEDT quantify the accuracy of estimated power and network metrics for different network constraints with and without DP. Our results show that it is difficult to estimate the power and network usage in a real-world scenario with a software-based power meter or local experiments. However, running FL on VMs and simultaneously monitoring the Ethernet interface give a good estimate of the actual network load. Due to high error rates this approach is not applicable to simulating multiple clients on one machine and monitoring the local loopback interface. Using CPU cycle based power meters on Raspberry Pis is easy to implement and it can have high prediction accuracies. Furthermore, FL specific extensions like DP further increase energy consumption. Out scalability experiments indicate a linear training time increase depending on the amount of participating clients and an increase in training time by a factor of three between the FL runs with and without DP.

ACKNOWLEDGMENTS

This paper is funded by the Bavarian Ministry of Economic Affairs, Regional Development and Energy under the REMORA project with the identifier DIK0111/03.

REFERENCES

[1] David Basin, Søren Debois, and Thomas Hildebrandt. 2018. On Purpose and by Necessity: Compliance Under the GDPR. In *Financial Cryptography and Data Security*, Sarah Meiklejohn and Kazue Sako (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 20–37.

[2] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečný, Stefano Mazzocchi, H. Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. 2019. Towards Federated Learning at Scale: System Design. arXiv:1902.01046 [cs.LG]

[3] Aurelien Bourdon, Adel Noureddine, Romain Rouvoy, and Lionel Seinturier. 2013. PowerAPI: A Software Library to Monitor the Energy Consumed at the Process-Level. *ERCIM News* 2013 (2013).

[4] Hafsa Bousbiat, Roumaysa Bousselidj, Yassine Himeur, Abbes Amira, Faycal Bensaali, Fodil Fadli, Wathiq Mansoor, and Wilfried Elmenreich. 2023. Crossing Roads of Federated Learning and Smart Grids: Overview, Challenges, and Perspectives. arXiv:2304.08602 [cs.LG]

[5] Christopher Briggs, Zhong Fan, and Peter Andras. 2021. Federated Learning for Short-term Residential Energy Demand Forecasting. arXiv:2105.13325 [cs.LG]

[6] Pietro Ferrara and Fausto Spoto. 2018. Static Analysis for GDPR Compliance. In *ITASEC 2018 - Italian Conference on Cyber Security*. CEUR Workshop Proceedings, Milan, Italy, 10. http://ceur-ws.org/Vol-2058/#paper-10

[7] Yunzhe Guo, Dan Wang, Arun Vishwanath, Cheng Xu, and Qi Li. 2020. Towards Federated Learning for HVAC Analytics: A Measurement Study. In *Proceedings of the Eleventh ACM International Conference on Future Energy Systems* (Virtual Event, Australia) *(e-Energy '20)*. Association for Computing Machinery, New York, NY, USA, 68–73. https://doi.org/10.1145/3396851.3397717

[8] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. 2019. Measuring the Effects of Non-Identical Data Distribution for Federated Visual Classification. arXiv:1909.06335 [cs.LG]

[9] IEA. 2022. Data Centres and Data Transmission Networks. https://www.iea.org/reports/data-centres-and-data-transmission-networks

[10] Fabian Kaup, Philip Gottschling, and David Hausheer. 2014. PowerPi: Measuring and modeling the power consumption of the Raspberry Pi. In *39th Annual IEEE Conference on Local Computer Networks*. IEEE, Edmonton, Canada, 236–243. https://doi.org/10.1109/LCN.2014.6925777

[11] Kamar Kesrouani, Houssam Kanso, and Adel Noureddine. 2020. A Preliminary Study of the Energy Impact of Software in Raspberry Pi devices. In *2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. IEEE, Bayonne, France, 231–234. https://doi.org/10.1109/WETICE49692.2020.00052

[12] A. S. Khatouni, M. Trevisan, and D. Giordano. 2019. Data-Driven Emulation of Mobile Access Networks. In *2019 15th International Conference on Network and Service Management (CNSM)*. IEEE, 1515 South Park Street, Halifax, Nova Scotia, Canada, 1–6. https://doi.org/10.23919/CNSM46954.2019.9012691

[13] Sangyoon Lee and Dae-Hyun Choi. 2022. Federated Reinforcement Learning for Energy Management of Multiple Smart Homes With Distributed Energy Resources. *IEEE Transactions on Industrial Informatics* 18, 1 (2022), 488–497. https://doi.org/10.1109/TII.2020.3035451

[14] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. 2020. Fair Resource Allocation in Federated Learning. arXiv:1905.10497 [cs.LG]

[15] Y. Liu, J. J. Q. Yu, J. Kang, D. Niyato, and S. Zhang. 2020. Privacy-Preserving Traffic Flow Prediction: A Federated Learning Approach. *IEEE Internet of Things Journal* 7, 8 (2020), 7751–7763. https://doi.org/10.1109/JIOT.2020.2991401

[16] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2023. Communication-Efficient Learning of Deep Networks from Decentralized Data. arXiv:1602.05629 [cs.LG]

[17] D. Moss. 2014. Modelling the network performance of DSL connections using netem.

[18] State of California Department of Justice. 2018. California Consumer Privacy Act of 2018 [1798.100 - 1798.199.100].

[19] D. E. O'Leary. 2013. Artificial Intelligence and Big Data. *IEEE Intelligent Systems* 28, 2 (2013), 96–99. https://doi.org/10.1109/MIS.2013.39

[20] Aman Priyanshu, Rakshit Naidu, Fatemehsadat Mireshghallah, and Mohammad Malekzadeh. 2021. Efficient Hyperparameter Optimization for Differentially Private Deep Learning.

[21] Xinchi Qiu, Titouan Parcollet, Daniel J. Beutel, Taner Topal, Akhil Mathur, and Nicholas D. Lane. 2020. Can Federated Learning Save The Planet? https://doi.org/10.48550/ARXIV.2010.06537

[22] Xinchi Qiu, Titouan Parcollet, Javier Fernandez-Marques, Pedro Porto Buarque de Gusmao, Yan Gao, Daniel J. Beutel, Taner Topal, Akhil Mathur, and Nicholas D. Lane. 2022. A first look into the carbon footprint of federated learning. arXiv:2102.07627 [cs.LG]

[23] Xidi Qu, Shengling Wang, Qin Hu, and Xiuzhen Cheng. 2021. Proof of Federated Learning: A Novel Energy-Recycling Consensus Algorithm. *IEEE Transactions on Parallel and Distributed Systems* 32, 8 (2021), 2074–2085. https://doi.org/10.1109/TPDS.2021.3056773

[24] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H. Brendan McMahan. 2021. Adaptive Federated Optimization. arXiv:2003.00295 [cs.LG]

[25] Y. Roh, G. Heo, and S. E. Whang. 2021. A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective. *IEEE Transactions on Knowledge and Data Engineering* 33, 4 (2021), 1328–1347. https://doi.org/10.1109/TKDE.2019.2946162

[26] Yuris Mulya Saputra, Dinh Thai Hoang, Diep N. Nguyen, Eryk Dutkiewicz, Markus Dominik Mueck, and Srikathyayani Srikanteswara. 2019. Energy Demand Prediction with Federated Learning for Electric Vehicle Networks. In *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, Waikoloa, Hawaii, USA, 1–6. https://doi.org/10.1109/GLOBECOM38437.2019.9013587

[27] Personal Data Protection Commission Singapore. 2014. Personal Data Protection Act.

[28] A. Taïk and S. Cherkaoui. 2020. Electrical Load Forecasting Using Edge Computing and Federated Learning. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. IEEE, Virtual, 1–6. https://doi.org/10.1109/ICC40277.2020.9148937

[29] Andrew Trask, Emma Bluemke, Ben Garfinkel, Claudia Ghezzou Cuervas-Mons, and Allan Dafoe. 2020. Beyond Privacy Trade-offs with Structured Transparency. https://doi.org/10.48550/ARXIV.2012.08347

[30] Ye Lin Tun, Kyi Thar, Chu Myaet Thwal, and Choong Seon Hong. 2021. Federated Learning based Energy Demand Prediction with Clustered Aggregation. In *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE, Jeju Island, Korea, 164–167. https://doi.org/10.1109/BigComp51126.2021.00039

[31] European Union. 2016. 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).

[32] Haijin Wang, Caomingzhe Si, and Junhua Zhao. 2021. A Federated Learning Framework for Non-Intrusive Load Monitoring. arXiv:2104.01618 [eess.SP]

[33] Haijin Wang, Caomingzhe Si, Junhua Zhao, Guolong Liu, and Fushuan Wen. 2021. Fed-NILM: A Federated Learning-based Non-Intrusive Load Monitoring Method for Privacy-Protection. arXiv:2105.11085 [cs.LG]

[34] Sihua Wang, Mingzhe Chen, Walid Saad, and Changchuan Yin. 2020. Federated Learning for Energy-Efficient Task Computing in Wireless Networks. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. IEEE, Virtual, 1–6. https://doi.org/10.1109/ICC40277.2020.9148625

[35] Zhaohui Yang, Mingzhe Chen, Walid Saad, Choong Seon Hong, and Mohammad Shikh-Bahaei. 2021. Energy Efficient Federated Learning Over Wireless Communication Networks. *IEEE Transactions on Wireless Communications* 20, 3 (2021), 1935–1949. https://doi.org/10.1109/TWC.2020.3037554

[36] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. 2021. Opacus: User-Friendly Differential Privacy Library in PyTorch. *arXiv preprint arXiv:2109.12298* (2021), 18.

[37] Xinyu Zhou, Jun Zhao, Huimei Han, and Claude Guet. 2022. Joint Optimization of Energy Consumption and Completion Time in Federated Learning. In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, Bologna, Italy, 1005–1017. https://doi.org/10.1109/ICDCS54860.2022.00101

[38] X. Zhu, J. Wang, Z. Hong, T. Xia, and J. Xiao. 2019. Federated Learning of Unsegmented Chinese Text Recognition Model. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, Portland, OR, USA, 1341–1345. https://doi.org/10.1109/ICTAI.2019.00186

A RASPBERRY PI POWER MODEL

Raspberry Pis do not come with any power meter capabilities. One approach to indirectly measure the power drawn is to leverage a CPU utilization based model. Such models reach accurracies of up to 97.5 %. Kaup et al. [10] and Kesrouani et al. [11] generated power models for model 2 B and model 3 B+, respectively. By using the same approach, we developed our own CPU based power model for the Raspberry Pi model 4. We generated models for CPU loads from 0 % to 50 % (P [W] = 0.02992 * u + 3.9364) and from 51 % to 100 % (P [W] = 0.031 * u + 5.2947). The variable u describes the quotient between two types of total numbers of CPU cycles at two consecutive time stamps: c_{busy} and c_{total}. The numerator is the difference between $c_{busy}[t]$ and $c_{busy}[t-1]$ and the denominator is the difference between $c_{total}[t]$ and $c_{total}[t-1]$. c_{busy} is the sum of c_{user}, c_{nice}, and c_{system}.

Table 4: CNN architecture used to train an MNIST predictor. Padding is always equal to zero.

Layer	Input	Kernel	Stride	Output
Conv2d	1x28x28	3x3	1	32x26x26
ReLU	32x26x26	-	-	32x26x26
Conv2d	32x26x26	3x3	1	64x24x24
ReLU	64x24x24	-	-	64x24x24
Pool2d	64x24x24			64x12x12
Dropout	64x12x12	-	-	64x12x12
Flatten	64x12x12	-	-	9216
Linear	9216	-	-	128
ReLU	128	-	-	128
Droput	128	-	-	128
Linear	128	-	-	10

B MODEL ARCHITECTURE FOR EXPERIMENTS

Table 4 provides details about the CNN architecture used in our FL experiments. This architecture yields a pickled model size of 4.8 MB. This a rather large model for the MNIST use case and we chose it to have more load on the network. The learning rate is set to 1.0 and the optimizer is Adadelta.

C TESTBED DESIGN

Figure 2 provides an holistic overview of the edge device testbed setup. Its hardware consists of the following components. The external power meter verifies the power readings of the power-over-Ethernet (PoE) switch.

(1) PoE switch (S5500-48T8SP)
(2) Raspberry Pi model 4 with PoE hat
(3) Jetson Nano
(4) External power meter with ZigBee communication capabilities

We split the software components into operational and monitoring related categories. The former consists in our case of the following parts:

(1) Ubuntu 20.04 LTS
(2) Watchdog: Power cycle Raspberry Pi if it is not responding after a given interval to avoid manual interactions with it when it freezes
(3) Network time protocol to synchronous time stamps on each device
(4) Ansible (DevOps) to ease deployment of experiments
(5) PostgreSQL running on an VM to store all measurements
(6) netem to emulate different network environments
(7) FL framework

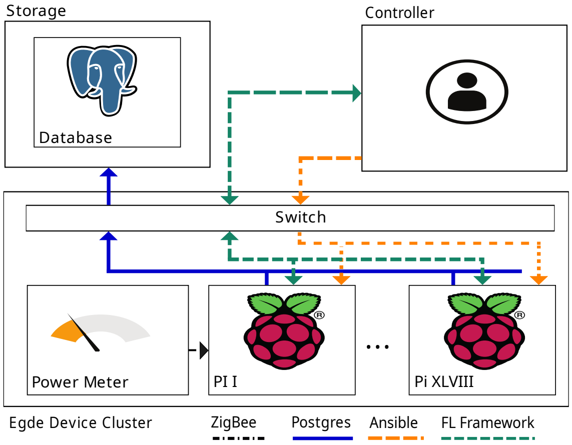


Figure 2: General overview of DEDT with its different communication protocols.

With psutil we monitor each devices' CPU and memory usage. A batch scripts takes care of starting and killing the network monitoring via tshark and we extract the switch's power readings with its API. The measurements are either stored in a PostgreSQL database running on an VM or are processed with the commercial product Weights and Biases. The data to the PostgreSQL database is either sent via Pub/Sub (Mosquitto) or each client directly connects to it.

Appendix C

Federated Computing in Electric Vehicles to Predict Coolant Temperature

Federated Computing in Electric Vehicles to Predict Coolant Temperature

René Schwermer
rene.schwermer@tum.de
Technical University of Munich
Munich, Germany

Ekin-Alp Bicer
ekin-alp.bicer@bmw.de
BMW Group
Munich, Germany

Pascal Schirmer
pascal.schirmer@bmw.de
BMW Group
Munich, Germany

Ruben Mayer
ruben.mayer@uni-bayreuth.de
University of Bayreuth
Bayreuth, Germany

Hans-Arno Jacobsen
jacobsen@eecg.toronto.edu
University of Toronto
Toronto, Canada

ABSTRACT

Reducing greenhouse gas emissions in mobility is paramount to achieving a carbon-neutral society. However, battery-electrical vehicles (BEV) introduce unique engineering challenges to protect expensive electrical components from overheating. A centralized architecture for model-driven predictions of coolant temperatures poses privacy and legal issues. Additionally, the applications in a vehicle compete for the available resources and must use them as sparingly as possible. Therefore, we introduce a new federated computing (FC) use case to help transform the mobility sector. We evaluate the performance of two FC approaches (linear regression and machine learning) on hardware and privacy metrics by leveraging a real-world dataset from BEVs. Our findings show trade-offs between hardware utilization and model accuracy. The linear regression model yields the best performance and prediction metrics. FC with ML shows up to 761 % variances when comparing vehicle-specific models with models trained with the entire fleet and clustering the data into velocity profiles partly improves prediction performance.

CCS CONCEPTS

• **Security and privacy**; • **Computing methodologies** → **Distributed computing methodologies**; • **Applied computing**;

KEYWORDS

Electric Vehicle, Federated Computing, Systems

ACM Reference Format:
René Schwermer, Ekin-Alp Bicer, Pascal Schirmer, Ruben Mayer, and Hans-Arno Jacobsen. 2023. Federated Computing in Electric Vehicles to Predict Coolant Temperature. In *24th International Middleware Conference Industrial Track (Middleware Industrial Track '23), December 11–15, 2023, Bologna, Italy.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3626562.3626829

Middleware Industrial Track '23, December 11–15, 2023, Bologna, Italy

ACM ISBN 979-8-4007-0427-7/23/12...$15.00
https://doi.org/10.1145/3626562.3626829

1 INTRODUCTION

The carbon dioxide emissions caused by the transportation and mobility sector in 2020 was 27.06 % of the European Union's (EU) total emissions, and vehicles drove 56.93 % of its carbon dioxide emissions [2]. The share of battery and plug-in electric (BEV and PHEV) vehicles at the new registrations increased in the EU from 3 % to 17.8 % between 2019 and 2021 [1]. Three drivers of this trend are the public's interest in reducing the personal carbon dioxide footprint, legal restrictions, and government subsidies [13, 44].

However, BEVs introduce unique engineering challenges due to higher proportions of costly electrical components, such as semiconductor parts. Keeping those parts at their optimum temperature is crucial to reduce failure rates. BEVs' thermal management system (TMS) holds the vehicle components at optimum temperatures. Figure 1 illustrates a simplified TMS of a BEV. The coolant and refrigerant are liquids. The latter can change its phase to gaseous. This complexity makes it challenging to predict the coolant temperature and its effect on all attached hardware components' performance, efficiency, and lifespan.

Vehicles generate different types of data, from time series data to images. Sending entire datasets to a central server can introduce network bottlenecks and cause privacy concerns by customers. Additionally, regulatory constraints make it more challenging to

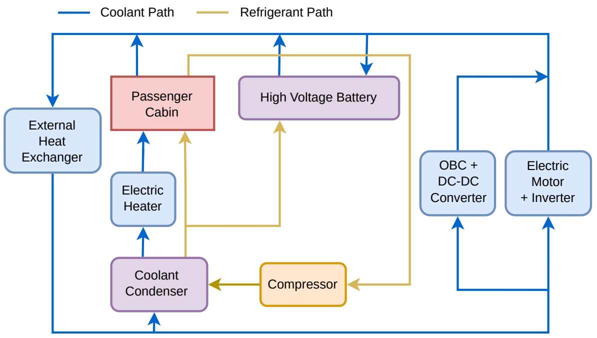


Figure 1: Simplified illustration of an BEV TMS. The On Board Charger (OBC) and DC-DC converter convert AC to DC for the battery and high voltage (400 V) to low voltage (12 V) for applications, such as the radio, respectively.

deploy centralized pipelines due to legal risks and compliance efforts [5, 11]. Since 2016 the General Data Protection Regulation has protected customers in the European Union [35]. Similar legal frameworks are the California Consumer Privacy Act in the USA [24] and the Personal Data Protection (Amendment) Act in Singapore [31]. A strategy to enable privacy-preserving computations is federated computing (FC). It keeps the data at its origin and shifts the computational workload to the source instead. However, computational resources and energy in a BEV are scarce. Therefore, we evaluate the impact of two different FC approaches on multiple hardware metrics (memory, storage, network, and energy) to find a trade-off between coolant temperature prediction accuracy and hardware requirements. Finally, with an FC-enabled architecture in combination with the Extended Vehicle (ExVe) or ADAXO (Automotive Data Access - Extended and Open) data standard [25], automotive manufacturers could learn from one another without sharing their datasets and tailor models more quickly to specific geographical locations or customer behaviors.

Our contributions are:

- Quantifying trade-offs between multiple hardware metrics (memory, storage, network, and energy) and privacy in a BEV context by leveraging physically distributed hardware to mimic real-world scenarios as closely as possible and using a real-world data set from test vehicles.
- Investigating the impact of two FC approaches (FA and FL) on model transferability for multiple vehicles with different engine types.
- Quantifying the impact of velocity profiles on the prediction performance.
- Evaluating the network on physically distributed edge devices and virtual machines in the context of FC to highlight differences.

This paper is structured as follows. First, we describe in detail the motivation for this paper Section 2. Then we describe related work in FC and temperature prediction (Section 3.1, and Section 3.2) in the automotive context. Section 4 describes the dataset and our pre-processing steps. The evaluation of our experiments concerning privacy, hardware utilization, and model transferability (FC) is in Section 5. Section 6 presents the conclusion of our findings.

2 MOTIVATION

Monitoring complex cooling circuits in a vehicle is challenging. An abundance of temperature sensors allows fine-grained measurements of the system. However, each sensor imposes design and engineering challenges [14, 37]. The space in a vehicle is limited, and neighboring components transfer heat from or to each other over the air or by direct contact. Additionally, temperature sensors themselves impact the transfer of excess heat. They also increase initial costs, potential error sources, and maintenance efforts due to broken or drifted sensors [37]. Therefore, each sensor should generate as many insights as possible. One approach to achieve this is data-driven models.

Knowing the temperatures of core components, such as semiconductor parts or capacitors, is crucial to the decrease downtime of vehicles due to broken parts. For example, a 10 °C higher temperature than estimated can reduce a capacitor's lifetime by half [3, 28]. There are three different non-ML approaches to assessing those temperatures in advance [38]. The most accurate approach is complex numerical simulation with high computing costs. More straightforward numerical simulations with complexity-reducing assumptions give a lower accuracy but a faster compute time. Lastly, assuming a constant temperature reflects the absolute worst case and results in over-engineered components. Enriching simulations with data-driven models enables engineers to reduce the feature space and the number of available design choices. Simulations with different customer mission profiles can increase a vehicle's efficiency by considering the model's output in the design and production phases. A car drives up to 200,000 km in long term tests under different conditions to test its performance and durability. Such test drives are expensive, and being able to run those with more accurately simulated components reduces failure rates and time needed.

A coolant temperature model could also enable adaptive thermal management and remote diagnostics [4, 8]. If we know in advance when a component is getting hotter than a given threshold, the car can adjust the cooling intake or flow. Improvements to the models can come with over-the-air updates. The models' hardware requirements should be as lows as possible. Computational resources in vehicles increase and with it the number of services competing for those resources [20]. Therefore, our prediction model should be as lean as possible in every way (storage, memory, network, and energy) [22].

3 STATE OF THE ART

To the best of our knowledge, there is no study combining FC and analytical feature engineering (AFE) to predict the average coolant temperature of a BEV by simultaneously measuring its computational footprint. However, there is research on individual parts of our toolchain (FC in vehicles and temperature prediction).

3.1 Federated Learning in Vehicles

One subcategory of privacy-preserving machine learning (ML) is FC. This method enables data scientists to work on remote data while keeping it private. The data stay at its origin, and each device trains its local model. Each client only transfers its model parameters to a central server, where each model is aggregated into one primary model and distributed back to each client. FC allows server-client or peer-to-peer architectures. One of the first aggregation strategies is FedAvg, which averages all updated model parameters. One challenge of FC is unevenly distributed labels (non-Independent and Identically Distributed). FC comprises federated learning (FL) and federated analytics (FA) [10, 39, 42]. The former applies iterative optimizations with multiple rounds, such as ML. In contrast, FA runs for only one round. It leverages statistical calculations such as average and sum. An example use case is the fit of linear regression models.

The connectivity of vehicles, their available network capabilities, and computational resources are increasing, introducing the term Internet-of-Vehicle (IoV). A challenge for distributed and non-stationary vehicles or IoV is connection loss and their heterogeneous individual datasets [17]. Therefore, Li et al. and Ye et al. developed a peer-to-peer architecture with an adopted aggregation strategy to consider vehicles dropping out during a training round [21, 41]. Their approaches achieve similar accuracy when compared to a centralized ML architecture. Other IoV use cases are stress level identification in vehicles with training on roadside units [36], image classification [23], and object detection [9]. Additionally, Tan et al. provide a short overview of other IoV use cases which leverage FL [34]. Besides model performance, it is also necessary to balance the ML models' energy consumption with a battery's state of charge to avoid a negative customer experience [32, 43]. However, no work leverages time series data to develop a coolant temperature prediction model in a privacy-preserving fashion.

3.2 Temperature Prediction

In the past works, some data driven temperature predictions for BEVs have been made, but there are few works that attempt to model the complete TMS. Moreover, to the best of the authors' knowledge, federated training of models for EV TMS has not been proposed. Billert et al. predicted high voltage battery coolant temperatures using quantile neural networks [6, 7]. Padros et al. generate neural network models to predict the important electric motor temperatures together with the coolant temperature using Karhunen-Loeve expansion as a pre-processing step [26]. Wallscheid et al. investigate ML to predict the temperatures within the permanent magnet synchronous motors. However, the coolant temperature is not analyzed [37, 38]. Park et al. leverage artificial neural networks to train a model for TMS subsystems, then uses the subsystem models to develop control strategies [27].

4 DATASET

Our dataset consists of time series measurements of 35 test BEVs from an automotive manufacturer. The vehicles drive around Germany with different routes. A mobile data recorder tracks all measurements with a frequency of 10 Hz and loads the logs into a local database. The vehicle category comprises one model (iX3) and two engine types (C and D). We have data from 10 and 23 vehicles for $iX3_C$ and $iX3_D$, respectively. The measured features capture the behavior of multiple temperatures (e. g., ambient, inverter, and oil), battery (e. g., state-of-charge, current, and voltage of high voltage system), and movement-related (torque, velocity) metrics (Table 1). However, not all metrics are always available. For example, test vehicles measure some metrics more frequently than a vehicle sold to a customer. Therefore, we focus on the feature mentioned in Table 1.

Analytical feature engineering. To increase the precision and transferability of the model without requiring vast amounts of data, we generate new features using the underlying equations that govern the physical processes within the TMS. The temperatures within the BEV depend on the generated and dissipated heat from the TMS components. The heat diffuses through surrounding components to reach the coolant. Therefore, we use physical equations related to heat losses and conduction to generate additional features based on existing measurements. Eddy and Hysteresis losses represent a portion of the losses in the electric motor [30], Fourier's Law [12], and Stefan-Boltzmann Law [40] describe heat diffusion and radiation. Those features are a supplement to the main features from Table 1 since they represent a portion of the power losses and temperature relations in BEVs.

Table 1: Features used for the experiments with their respective unit and value range.

Name	Symbol	Unit	Range
Velocity	v	km/h	[0, 180]
Torque	M	Nm	[0, 400]
RPM	n	min^{-1}	[0, 15000]
HV System Voltage	V_{HVDC}	V	[300, 400]
Ambient Temp.	T_{Amb}	°C	[0, 40]
Avg. Coolant Temp.	T_C	°C	[5, 55]

Table 2 provides an overview of newly calculated features. The $\odot$ operation is an element-wise multiplication. T is the temperature, t is time, k is the thermal conductivity, c is the specific heat capacity, ρ is the density, j^* is the radiated heat loss from a black body, σ is the Stefan-Boltzmann constant.

Table 2: Empirical generation of features using physical equations. M, n and T_{Amb} describe the torque, rotation speed and ambient temperature, respectively.

	Equation	Operation
Mechanical Power [30]	$P = Mn$	$M \odot n$
Eddy Current Loss [30]	$K_e B^2 f^2$	$M^2 \odot n^2$
Hysteresis Loss [30]	$K_h B^{1.6} f$	$M^{1.6} \odot n$
Fourier's Law [12]	$\frac{\partial T}{\partial t} = \frac{k}{c\rho} \frac{\partial T^2}{\partial^2 x^2}$	T_{Amb}^2
Stefan-Boltzmann Law [40]	$j^* = \sigma T^4$	T_{Amb}^4

Data pre-processing. The dataset requires pre-processing to remove inaccurate measurements. Those include outliers, non-changing values over a long period, or NaN values. We remove those measurements, drop features containing only zeros, and normalize each column by its min/max (ML) or mean (Lasso) values. Some sensor measurements keep getting logged after the vehicle stops operating, even though the data collection should stop. We apply a stand-still filter similar to Padros et al. [26] to eliminate those measurements by removing entries where the torque and velocity equal zero. Additionally, a dataset per vehicle consists of multiple separate driving events. We separate those via the timestamp. Any time gap above 10 minutes is considered an end to a driving session.

5 EXPERIMENT EVALUATIONS

We simulate a fleet of vehicles with Raspberry Pis model 4 to get more realistic measurements than simulating a fleet on one machine or with VMs. Table 3 provides a summary of our results. The Lasso model requires fewer computational resources to train a model than an ML model. We explain each measurement in more detail in the following sections.

The Raspberry Pis emulate the available computational resources in a vehicle. We limit our application to running on a weak device because similar devices already operate in BEV. Another reason is to simulate the internal competition of services for hardware resources. During our experiments, we evaluate four hardware

Table 3: Summary of hardware metric evaluations measured during the training of the Lasso and ML model.

	Memory [MB]	Storage [kB]	Network [kB]	Energy [J]
Lasso (Pi)	110	0.274	0.38	0.37
ML (Pi)	175	20.5	86	140

metrics (memory, storage, network, and energy) for an FA and FL strategy together with their respective trade-offs (Section 5.3).

We monitor the network during each experiment with `tshark`. We filter the monitored network traffic by port to eliminate any noise. Additionally, we normalize all network- and energy-related measurements by the number of rounds to allow easier comparison between different runs. All Raspberry Pis draw their power via a Power-over-Ethernet (PoE) capable from the switch. It measures the individual power drawn.

5.1 Accuracy

We use the R² score to benchmark the prediction performance of our models. Its highest value is one. It can be arbitrarily worse and also be negative. The model training runs with up to three features from Table 2 and Table 1, and T_C is always the label. No feature correlates with another one. We either take the entire dataset of a vehicle (no WLTP) or split it by three velocity ranges (<= 50 km/h, > 50 km/h and <= 100 km/h, > 100 km/h) based on the Worldwide Harmonised Light Vehicles Test Procedure (WLTP), which describes driving profiles to measure the vehicle's emissions in real-world usage. We highlight the effect of WLTP only on locally trained and the Lasso models.

We train a Lasso model per BEV with all possible feature combinations (up to three features) to find the best-performing features. Figure 2 shows the highest respective R² score per BEV for each model and engine type. The WLTP boxplot shows the velocity profile > 100 km/h. However, the other two look similar. Dividing the data into velocity profiles improves the R² score. The most occurring features with the highest R² score are T_{Amb} and V_{HVDC}. Also, all models with the best prediction performance use three features in total. However, not all models use the same feature combination. We merge only coefficients from the same features for the FA and FL experiments. Therefore, we restrict each category to use the three most occurring features. Those are V_{HVDC}, T_{Amb}, and Stefan-Boltzmann Law. Not all vehicles have those features available, resulting in four and one BEV dropping out for $iX3_C$ and $iX3_D$, respectively.

When training only with V_{HVDC}, T_{Amb}, and Stefan-Boltzmann Law as features, the R² score ranges from 0.68 to 0.77 and 0.19 to 0.86 for the $iX3_C$ and $iX3_D$ BEVs, respectively. With WLTP, the R² score improves on average by 0.04 for both vehicle categories. Figure 4 shows one car's measured and predicted coolant temperature using a Lasso and an ML model. Both approaches have some deviations for high coolant temperatures. Their prediction performance is feasible for a real-world application in BEVs, and both methods achieve a similar output. However, this looks different for the FA and FL scenarios.

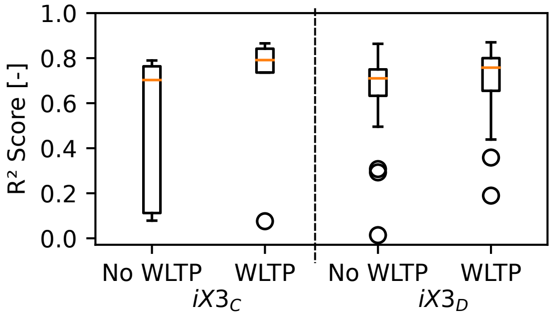


Figure 2: Maximum R² score of all BEVs with and without WLTP velocity profiles. The underlying model is a Lasso regression mode trained on three features. The feature combinations vary between the BEVs.

Each Raspberry Pi trains one Lasso model per vehicle, and a central server aggregates all model parameters. Those are three coefficients (one per feature) and the y-intercept. We use the FedAvg algorithm to aggregate all models. It averages all received model parameters. Each BEV calculates the R² score with the aggregated model by testing it with its own data set. First, we run one FA experiment each for $iX3_C$ and $iX3_D$. Figure 3 shows the percentage difference of the R² score between the individual models (local) to the aggregated ones (FA). A percentage difference of zero describes that the local and the FA achieve the same R² score. Generally, the R² scores of the FA run get worst compared to locally trained ones. On average it decreases by 18.1 % and 4.36 % for $iX3_C$ and $iX3_D$, respectively. For $iX3_C$, we excluded an extreme outlier of 254 %. However, running FA with all BEV yields for $iX3_C$ vehicles an average R² score improvement of 12 %. The median R² score for $iX3_D$ stays constant. However, an outlier increased from -12 % to -101 %. For WLTP-based FA runs with all vehicles, the average R² score decreases by 57 % and 14 % for $iX3_C$ and $iX3_D$, respectively. Therefore, categorizing the data into WLTP profiles is not a feasible strategy to improve the overall prediction performance of the vehicle fleet in a FA system.

We use an ML model coming from the energy domain [18]. The best-performing feature combination for the Lasso model is the basis for the ML models. Training locally on $iX3_C$ vehicles yield R² scores between -0.031 and 0.77. For $iX3_D$ vehicles, the R² score varies between -0.01 and 0.85 with an average of 0.69. The R² score with WLTP profiles vary by up to 100 % between them for $iX3_C$ and are more constant for $iX3_D$. The right plot in Figure 3 provides an overview of our FL experiments. The FL with FedAvg yield in general worst R² score compared to each vehicle training its own model. We also tested QFedAvg [19], FedAvgM [15], FedAdam [29] with their default settings, which all gave worst results than the FedAvg aggregation strategy. FL for $iX3_C$ and $iX3_D$ yields on average an R² score of 0.41 % (median 0.56) and 0.48 % (median 0.43), respectively. FL with both vehicle categories reduces the R² score on average by 50 % compared to a vehicle-specific FL.

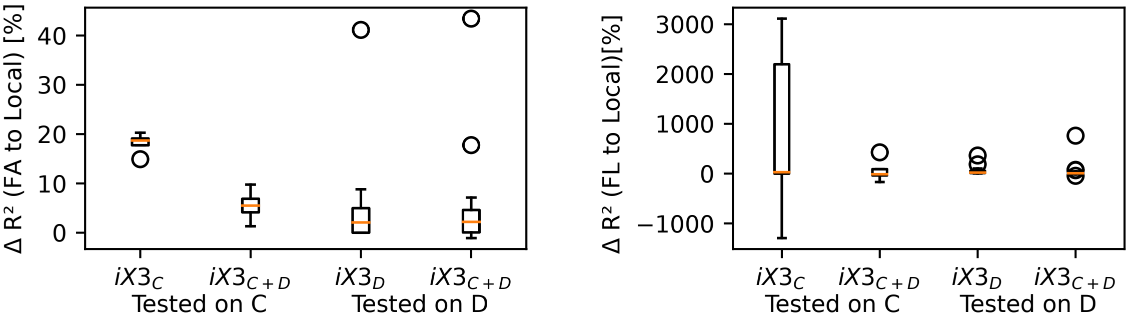


Figure 3: Comparison of the R² scores between the locally trained and FA (left) and FL (right) models. The aggregated models consist of parameters coming from only one engine type (C or D) or from both engine types (C + D). The lower Δ R² the better.

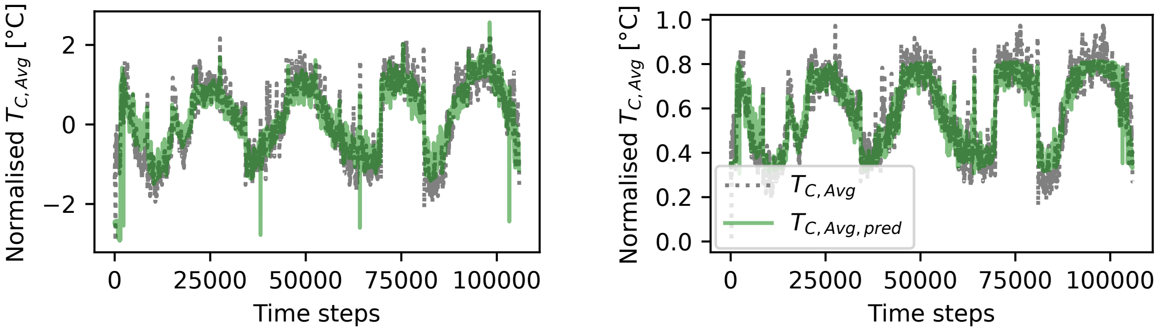


Figure 4: Prediction performance of a Lasso and a ML model trained with the same features (V_{HVDC}, T_{Amb}, and n) on one vehicle. Left: Lasso linear regression model. Right: ML model consisting of CNN and LSTM components.

5.2 Hardware

Memory. Loading the entire dataset into memory consumes 100 MB of memory and takes 1.7 s. We limit memory utilization with three strategies. We are first reading only the required columns instead of the entire dataset (40 MB and 0.8 s execution time). Second, change the data type from Float64 to Float32 (20 MB), and third, read the data in batches of 10,000 in Float32 format (0.04 MB and 0.017 s). Therefore, the maximum memory reduction and data reading time increase are 99 % and 97 %, respectively. Another system to further reduce the memory footprint is a shift from Python to a more resource-efficient programming language such as C++. However, there is a trade-off between development speed, code complexity, and memory footprint. Also, all existing FL frameworks are in Python. Therefore, we limit ourselves to Python.

The additional memory consumption of the Lasso and ML model training is 110 MB and 170 MB, respectively. Additionally, the memory consumption increases for the ML model linearly by about 0.6 MB per round, and it stays constant for the Lasso model. The ML model requires 54 % more memory over a more extended period, which can block the execution of other tasks.

Storage. Due to the storage demands of other applications, the respective model size should be as small as possible. For the Lasso regression model, we compare the storage requirements of four different data types: Pickle, skops, open neural network exchange (ONNX) format, or predictive model markup language (PMML). Their respective model size is 541 B, 7732 B, 274 B, and 1259 B. Pickle files are not secure and their documentation encourages users only to open trusted pickle files. ONNX is standard for representing machine learning algorithms, and PMML is an XML-based predictive model interchange format. We opt for ONNX with the lowest storage requirements and the possibility to use it with different frameworks, such as *scikit-learn*, *PyTorch* or *TensorFlow*.

The tested formats for the ML model are Pickle, skops, and ONNX. Their respective file size is 22 kB, 18.2 kB, and 20.5 kB. We also pick the ONNX model as our final storage format. It is 74 times larger than the Lasso model due to the higher number of parameters. The Lasso model has three coefficients and a y-intercept. In contrast, the ML model consists of 3833 parameters. Another approach to reducing model size for inference is dynamic quantization. We quantize the LSTM and linear layer to Int8. However, the resulting model size remains the same.

Network. Transferring 100 MB of raw data over the network via rsync results in about 4000 messages. The total network traffic and number of messages per client both ways for the linear regression model is 0.37 kB and six, respectively. Compared to a centralized approach, the FA approach reduces the total network traffic by 99 %. The measured network traffic on the VMs is 0.26 kB and four messages. It deviates by 30 % from the measurements made on the Raspberry Pis. Therefore, it is not an excellent indicator for estimating real-world network requirements.

The training-induced network traffic per round for the ML model is 210 times higher than the one from the linear regression model. After a round, each client sent and received a total of 86 kB. The number of messages increases by a factor of 10 compared to the linear regression model and is about 60 messages per round. The network traffic per client for 100 rounds is 8.6 MB. The measurements on the VM are 30 % and 500 % less for the network traffic and number of messages, respectively.

Energy. The training of a Lasso regression model takes 0.007 s, and the measurement frequency of the PoE switch is 10 s. We run 100 identical training rounds to bring those two together and calculate an average energy consumption per round. We excluded for all experiments the idle load of the Raspberry Pis to isolate the energy consumption of the training itself. This results in an energy consumption of 0.37 J. The total energy consumption of this FA system scales linearly with the number of participating clients.

Each FL round per client takes about 8 s and consumes 140 J. The ML model's training time and energy consumption are 1142 times slower and 378 times higher than the training of the Lasso model, respectively. The Lasso regression model is more favorable from an energy perspective than the ML approach. A typical BEV has an energy consumption of 20 kWh/100 km. Thus the range reduction per round for the FA and FL training are 0.05 cm and 18.7 cm, respectively.

5.3 Model Comparison

Figure 5 compares the differences in our performance metrics for centralized training and two FC approaches (Lasso and an ML model). A one indicates the best performance. A centralized training and model inference does not require any energy, memory, or storage on the vehicle when neglecting the energy necessary to send the data to a central server. Therefore, it gets a one on those metrics. However, even though we expect a higher model accuracy due to more data, it reaches a zero for the privacy metric. Based on our Raspberry Pi experiments, we put the Lasso and ML model into perspective to the centralized approach (see Table 3). For example, storage requirements on vehicles with central training are zero, and the highest storage demand is from the ML model (20.5 kB). Therefore, the Lasso model gets a score of 0.98 (0.274 kB). The Lasso model requires minor network traffic and a small storage capacity. Its privacy level is a bit lower than the one from the ML model. The Lasso linear regression model is deterministic, making it more prone to inference attacks when compared to an ML which has some randomness due to the optimizer [16, 33]. We recommend using a linear regression model because of its trade-offs between hardware metrics, privacy level, accuracy, and complexity.

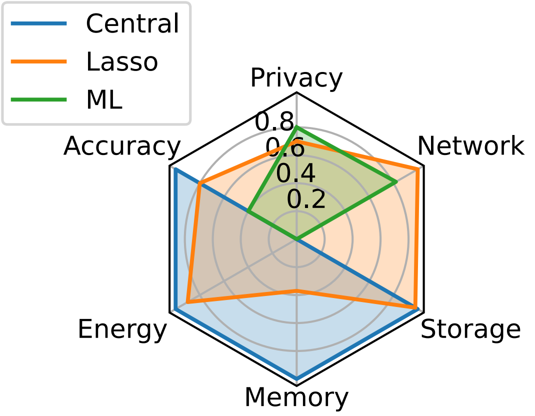


Figure 5: Trade-offs between multiple metrics for three different approaches. A one represents the best value.

6 CONCLUSION

Computational resources in vehicles, together with the number of applications competing for those resources, are increasing. Additionally, policymakers and customers get more privacy sensitivity. With our FA and FL experiments and their respective hardware requirements evaluations, we enable car manufacturers to better balance hardware and privacy needs. Our findings show that a Lasso linear regression model achieves in an FC architecture better R^2 scores than an ML model and requires fewer hardware resources. However, it is more prone to inference attacks. Clustering by WLTP velocity profiles can improve prediction performance. Additionally, we show that data-driven predictions can help engineers design the vehicle. Reducing malfunctions in electrical components increases acceptance of BEV and helps to transition the automotive sector towards a more carbon-neutral industry. In future work, we will evaluate compression techniques to reduce network traffic and try to find shared data properties for well-performing vehicles.

ACKNOWLEDGMENTS

This paper is funded by the Bavarian Ministry of Economic Affairs, Regional Development and Energy under the REMORA project with the identifier DIK0111/03.

REFERENCES

[1] European Environment Agency. 2022. New registrations of electric vehicles in Europe. https://www.eea.europa.eu/ims/new-registrations-of-electric-vehicles Accessed March 20, 2023.
[2] European Environment Agency. 2023. EEA greenhouse gases - data viewer. https://www.eea.europa.eu/data-and-maps/data/data-viewers/greenhouse-gases-viewer Accessed March 20, 2023.
[3] Arne Albertsen. 2010. Electrolytic Capacitor Lifetime Estimation. *Bodos Power Magazine* (04 2010), 52–54.
[4] Philip Arejola, Ondrej Burkacky, Johannes Deichmann, Gourav Ganguly, Asif Khan, and Martin Wrulich. 2022. The future of automotive computing: Cloud and edge.
[5] David Basin, Søren Debois, and Thomas Hildebrandt. 2018. On Purpose and by Necessity: Compliance Under the GDPR. In *Financial Cryptography and Data Security*, Sarah Meiklejohn and Kazue Sako (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 20–37.
[6] Andreas Billert, Stefan Erschen, Michael Frey, and Frank Gauterin. 2022. Predictive Battery Thermal Management using Quantile Convolutional Neural Networks. *Transportation Engineering* 10 (11 2022), 100150. https://doi.org/10.1016/j.

treng.2022.100150

[7] Andreas Billert, Michael Frey, and Frank Gauterin. 2022. A Method of Developing Quantile Convolutional Neural Networks for Electric Vehicle Battery Temperature Prediction Trained on Cross-Domain Data. 3 (05 2022), 1–1. https://doi.org/10.1109/OJITS.2022.3177007

[8] BMW. 2023. BMW Group Technology Trend Radar. https://www.bmwgroup.com/en/innovation/company/technology-trend-radar.html Accessed September 26, 2023.

[9] Ahmet M. Elbir, Burak Soner, Sinem Çöleri, Deniz Gündüz, and Mehdi Bennis. 2022. Federated Learning in Vehicular Networks. In *2022 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*. 72–77. https://doi.org/10.1109/MeditCom55741.2022.9928621

[10] Ahmed Roushdy Elkordy, Yahya H. Ezzeldin, Shanshan Han, Shantanu Sharma, Chaoyang He, Sharad Mehrotra, and Salman Avestimehr. 2023. Federated Analytics: A survey. (2023). https://doi.org/10.48550/ARXIV.2302.01326

[11] Pietro Ferrara and Fausto Spoto. 2018. Static Analysis for GDPR Compliance.

[12] Jean Baptiste Joseph Fourier. 2009. *The Analytical Theory of Heat*. Cambridge University Press. https://doi.org/10.1017/CBO9780511693205

[13] Scott Hardman, Amrit Chandan, Gil Tal, and Tom Turrentine. 2017. The effectiveness of financial purchase incentives for battery electric vehicles – A review of the evidence. *Renewable and Sustainable Energy Reviews* 80 (2017), 1100–1111. https://doi.org/10.1016/j.rser.2017.05.255

[14] Jonathan Hey, Adam C. Malloy, Ricardo Martinez-Botas, and Michael Lampérth. 2016. Online Monitoring of Electromagnetic Losses in an Electric Motor Indirectly Through Temperature Measurement. *IEEE Transactions on Energy Conversion* 31, 4 (2016), 1347–1355. https://doi.org/10.1109/TEC.2016.2562029

[15] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. 2019. Measuring the Effects of Non-Identical Data Distribution for Federated Visual Classification. arXiv:1909.06335 [cs.LG]

[16] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. 2018. Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning. In *2018 IEEE Symposium on Security and Privacy (SP)*. 19–35. https://doi.org/10.1109/SP.2018.00057

[17] Baofeng Ji, Xueru Zhang, Shahid Mumtaz, Congzheng Han, Chunguo Li, Hong Wen, and Dan Wang. 2020. Survey on the Internet of Vehicles: Network Architectures and Applications. *IEEE Communications Standards Magazine* 4, 1 (2020), 34–41. https://doi.org/10.1109/MCOMSTD.001.1900053

[18] Honghai Kuang, Qian Guo, Shengqing Li, and Hao Zhong. 2021. Short-term wind power forecasting model based on multi-feature extraction and CNN-LSTM. *IOP Conference Series: Earth and Environmental Science* 702, 1 (mar 2021), 012019. https://doi.org/10.1088/1755-1315/702/1/012019

[19] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. 2020. Fair Resource Allocation in Federated Learning. arXiv:1905.10497 [cs.LG]

[20] Xin Li, Yifan Dang, and Tefang Chen. 2018. Vehicular Edge Cloud Computing: Depressurize the Intelligent Vehicles Onboard Computational Power. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. 3421–3426. https://doi.org/10.1109/ITSC.2018.8569286

[21] Yiran Li, Hongwei Li, Guowen Xu, Tao Xiang, and Rongxing Lu. 2022. Practical Privacy-Preserving Federated Learning in Vehicular Fog Computing. *IEEE Transactions on Vehicular Technology* 71, 5 (2022), 4692–4705. https://doi.org/10.1109/TVT.2022.3150806

[22] Liangkai Liu, Sidi Lu, Ren Zhong, Baofu Wu, Yongtao Yao, Qingyang Zhang, and Weisong Shi. 2021. Computing Systems for Autonomous Driving: State of the Art and Challenges. *IEEE Internet of Things Journal* 8, 8 (2021), 6469–6486. https://doi.org/10.1109/JIOT.2020.3043716

[23] Su Liu, Jiong Yu, Xiaoheng Deng, and Shaohua Wan. 2022. FedCPF: An Efficient-Communication Federated Learning Approach for Vehicular Edge Computing in 6G Communication Networks. *IEEE Transactions on Intelligent Transportation Systems* 23, 2 (2022), 1616–1629. https://doi.org/10.1109/TITS.2021.3099368

[24] State of California Department of Justice. 2018. California Consumer Privacy Act of 2018 [1798.100 - 1798.199.100].

[25] German Association of the Automotive Industry. 2022. ADAXO: Automotive Data Access – Extended and Open: VDA concept for access to in-vehicle data.

[26] Marc Sebastián Padrós, Pascal A. Schirmer, and Iosif Mporas. 2022. Estimation of Cooling Circuits' Temperature in Battery Electric Vehicles Using Karhunen Loeve Expansion and LSTM. In *2022 30th European Signal Processing Conference (EUSIPCO)*. 1546–1550. https://doi.org/10.23919/EUSIPCO55093.2022.9909690

[27] Jonghyun Park and Youngjin Kim. 2020. Supervised-Learning-Based Optimal Thermal Management in an Electric Vehicle. *IEEE Access* 8 (2020), 1290–1302. https://doi.org/10.1109/ACCESS.2019.2961791

[28] Sam G. Parler. 2004. Deriving Life Multipliers for Electrolytic Capacitors. https://api.semanticscholar.org/CorpusID:107675698

[29] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H. Brendan McMahan. 2021. Adaptive Federated Optimization. arXiv:2003.00295 [cs.LG]

[30] Dierk Schröder. 2009. *Elektrische Antriebe - Regelung von Antriebssystemen.* https://doi.org/10.1007/978-3-540-89613-5

[31] Personal Data Protection Commission Singapore. 2014. Personal Data Protection Act.

[32] Soumya Sudhakar, Vivienne Sze, and Sertac Karaman. 2023. Data Centers on Wheels: Emissions From Computing Onboard Autonomous Vehicles. *IEEE Micro* 43, 1 (2023), 29–39. https://doi.org/10.1109/MM.2022.3219803

[33] Jasper Tan, Blake Mason, Hamid Javadi, and Richard Baraniuk. 2022. Parameters or Privacy: A Provable Tradeoff Between Overparameterization and Membership Inference. In *Advances in Neural Information Processing Systems*, Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (Eds.). https://openreview.net/forum?id=7nypt7cjNL

[34] Kang Tan, Duncan Bremner, Julien Le Kernec, and Muhammad Imran. 2020. Federated Machine Learning in Vehicular Networks: A summary of Recent Applications. In *2020 International Conference on UK-China Emerging Technologies (UCET)*. 1–4. https://doi.org/10.1109/UCET51115.2020.9205482

[35] European Union. 2016. 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).

[36] Jayant Vyas, Debasis Das, and Sajal K. Das. 2020. Vehicular Edge Computing Based Driver Recommendation System Using Federated Learning. In *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. 675–683. https://doi.org/10.1109/MASS50613.2020.00087

[37] Oliver Wallscheid. 2021. Thermal Monitoring of Electric Motors: State-of-the-Art Review and Future Challenges. *IEEE Open Journal of Industry Applications* 2 (2021), 204–223. https://doi.org/10.1109/OJIA.2021.3091870

[38] Oliver Wallscheid, Wilhelm Kirchgässner, and Joachim Böcker. 2017. Investigation of long short-term memory networks to temperature prediction for permanent magnet synchronous motors. In *2017 International Joint Conference on Neural Networks (IJCNN)*. 1940–1947. https://doi.org/10.1109/IJCNN.2017.7966088

[39] Dan Wang, Siping Shi, Yifei Zhu, and Zhu Han. 2022. Federated Analytics: Opportunities and Challenges. *IEEE Network* 36, 1 (2022), 151–158. https://doi.org/10.1109/MNET.101.2100328

[40] Mark Wellons. 2019. Stefan–Boltzmann Law. *Introduction to Quantum Mechanics 1* (2019).

[41] Dongdong Ye, Rong Yu, Miao Pan, and Zhu Han. 2020. Federated Learning in Vehicular Edge Computing: A Selective Model Aggregation Approach. *IEEE Access* 8 (2020), 23920–23935. https://doi.org/10.1109/ACCESS.2020.2968399

[42] Xuefei Yin, Yanming Zhu, and Jiankun Hu. 2021. A Comprehensive Survey of Privacy-Preserving Federated Learning: A Taxonomy, Review, and Future Directions. *ACM Comput. Surv.* 54, 6, Article 131 (jul 2021), 36 pages. https://doi.org/10.1145/3460427

[43] ZF. 2021. ZF ProAI: The Source of Vehicle Intelligence. https://www.zf.com/products/en/cars/stories/proai.html Accessed March 21, 2023.

[44] Yan Zhou, Michael Wang, Han Hao, Larry Johnson, and Hewu and Wang. 2015. Plug-in electric vehicle market penetration and incentives: a global review. (2015). https://doi.org/10.1007/s11027-014-9611-2

Appendix D

Federated Computing - Survey on Building Blocks, Extensions and Systems

This paper is currently under submission and published on arXiv [95].

Federated Computing - Survey on Building Blocks, Extensions and Systems

RENÉ SCHWERMER, Technical University of Munich, Germany
RUBEN MAYER, University of Bayreuth, Germany
HANS-ARNO JACOBSEN, University of Toronto, Canada

In response to the increasing volume and sensitivity of data, traditional centralized computing models face challenges, such as data security breaches and regulatory hurdles. Federated Computing (FC) addresses these concerns by enabling collaborative processing without compromising individual data privacy. This is achieved through a decentralized network of devices, each retaining control over its data, while participating in collective computations. The motivation behind FC extends beyond technical considerations to encompass societal implications. As the need for responsible AI and ethical data practices intensifies, FC aligns with the principles of user empowerment and data sovereignty.

FC comprises of Federated Learning (FL) and Federated Analytics (FA). FC systems became more complex over time and they currently lack a clear definition and taxonomy describing its moving pieces. Current surveys capture domain-specific FL use cases, describe individual components in an FC pipeline individually or decoupled from each other, or provide a quantitative overview of the number of published papers. This work surveys more than 150 papers to distill the underlying structure of FC systems with their basic building blocks, extensions, architecture, environment, and motivation. We capture FL and FA systems individually and point out unique difference between those two.

CCS Concepts: • **Computing methodologies** → **Distributed computing methodologies**; **Machine learning**; • **Security and privacy**;

Additional Key Words and Phrases: Federated Computing, Systems, Taxonomy

René Schwermer, Ruben Mayer, and Hans-Arno Jacobsen. Federated Computing - Survey on Building Blocks, Extensions and Systems.

1 INTRODUCTION

An increase in distributed Internet-of-Things (IoT) devices and the amount of generated data on remote, distributed devices puts stress on central processing entities and the network. Current inductive-based algorithms, such as machine learning training, run primarily on one device (i.e., a GPU), or distributed units execute the computations. In both cases, one entity or stakeholder monitors the entire process. An example is a neural network running on multiple graphical processing units trying to learn patterns in a dataset [45]. However, privacy concerns and regulatory constraints make it more challenging to deploy centralized pipelines due to legal risks, compliance efforts, and a higher consumer sensitivity [15, 49]. Such regularity constraints are put in place all around the world, and they vary in their complexity. Some examples are the General Data Protection Regulation in the European Union [132], the California Consumer Privacy Act in the USA [93], and the Personal Data Protection (Amendment) Act in Singapore [120]. One approach to tackle these issues is to shift the computational workload to the devices that generate the data in the first place. With Federated Computing (FC), data scientists and other stakeholders try to disentangle

Authors' addresses: René Schwermer, rene.schwermer@tum.de, Technical University of Munich, Munich, Germany; Ruben Mayer, ruben.mayer@uni-bayreuth.de, University of Bayreuth, Bayreuth, Germany; Hans-Arno Jacobsen, jacobsen@eecg.toronto.edu, University of Toronto, Toronto, Canada.

the contradiction between using distributed, privacy-sensitive data, regulatory frameworks, and consumer needs.

However, the literature needs to clearly define FC and how to extend it with approaches from other domains. There are diverse FC paradigms, and it is possible to unintentionally mingle neighboring techniques with FC, such as Federated Databases. FC paradigms are currently Federated Analytics and Federated Learning. However, in the future, we might see additional paradigms emerging as another branch of FC. Those FC paradigms share a similar basis but differ in certain parts. Therefore, it is paramount to have a standard definition for all existing and potential future FC paradigms to distinguish them. Such a framework allows the description of FC systems and subsequently increases comparability with other FC systems to highlight similarities and differences.

Our goal is to show which components (basic building blocks) are at least required to consider something as an FC system and which extensions currently exist. We show which kinds of FC systems exist, which problems they are trying to solve, and which combinations of essential building blocks and extensions the literature currently focuses on. Our proposed framework is expandable to allow it to grow over time or adjust with a changing view of FC systems.

Our contributions are:

(1) We present a detailed study of recent developments and trends in FC with a focus on the system level. We distinguish in our survey between FL and FA. Our survey highlights research gaps and prevalent system configurations.
(2) We develop a reference framework for FC and categorize literature accordingly. We highlight how FC basic building blocks and extension work together in different scenarios. A standardized way of describing FC systems allows to compare different works and it enables other researchers to more quickly identify bottlenecks or improvements in the future.
(3) We perform a deep discussion of underlying core technologies of FC on a low-level basis (serilization and communication protocol) and categorize existing FL frameworks. Knowing about the state-of-the art of the underlying communication allows to identify improvements with respect to network traffic and latency.
(4) We present a taxonomy to describe client selection algorithms and summarize widely adopted algorithms.

The rest of this paper is structured as follows. First, we give an overview of other surveys focusing on FC systems in Section 2 and how our paper differs from them. Then, we describe in Section 3 how FC works and we discuss its challenges, how to build an FC system, and which external factors influence the design process.

2 RELATED WORK

In this survey, we look at FC from a system perspective. Other surveys followed a similar holistic approach. However, they mainly cover individual components in an FC pipeline separate from each other and focus on Federated Learning (FL) (without covering Federated Analytics (FA)) or a specific domain. We cluster each survey into one of the following categories: Quantitative analysis, tutorial, domain-specific, and taxonomy. Table 1 provides an overview of other surveys and the respective features the capture.

A quantitative analysis describes trends and movements in a particular area by evaluating the number of publications. Farooq et al. [47] and Lo et al. [82] present such an analysis for existing FL research papers without considering FA. They highlight the recent increase in publications starting in 2017. The yearly publications increased from 25 in 2017 to 280 in 2020. Additionally, they cluster papers into different categories to distill focus areas. Most papers investigate the impact of training settings on ML model performance. The main reason to adopt FL is data privacy (62 % of papers), followed by communication efficiency (23 %). In general, their research emphasizes the increasing interest in FL. Multiple tutorial-like surveys exist as a reaction, which describe individual components of an FL pipeline together with some application examples.

Table 1. Summary of existing surveys and what kind of components in an FC system they cover. Those include FC basic building blocks (client selection, aggregation, communication) and extensions from other domains (e.g., privacy-enhancing techniques or compression).

	FC Basics			**FC Extensions**		**System Level**
Reference	Client Selection	Aggregation	CP	PET	Compression	
Quantitative Survey						
Farooq et al. [47]				✓		
Lo et al. [82]		✓		✓		✓
Tutorial Survey						
Aledhari et al. [7]	✓	✓				
Yang et al. [148]				✓		✓
Abreha et al. [5]	✓	✓		✓	✓	
Li et al. [75]		✓		✓		
Kairouz et al. [67]				✓		✓
Reddy et al. [102]	✓	✓				
Zhang et al. [155]		✓		✓		
Domain Specific Survey						
Xia et al. [143]	✓			✓	✓	
Briggs et al. [19]		✓		✓	✓	
Zhou et al. [160]				✓		
Wei et al. [142]	✓			✓		
Thapa et al. [127]				✓		
Kumar et al. [72]						✓
Dirir et al. [34]				✓		✓
Enthoven et al. [44]				✓		
Pfeiffer et al. [97]	✓	✓				
Gecer et al. [55]	✓			✓		
Zhu et al. [161]	✓	✓				
Taxonomy Survey						
Li et al. [76]				✓	✓	✓
AbdulRahman et al. [3]	✓	✓		✓	✓	
Yin et al. [151]				✓		
Bellavista et al. [16]		✓		✓	✓	
Bonawitz et al. [18]	✓			✓		✓
Our survey	✓	✓	✓	✓	✓	✓

CP = Communication Protocol, PET = Privacy-Enhancing Techniques

Aledhari et al. [7] give in their survey an overview about different FL architectures and their components. Each component with its features and different implementations is highlighted individually. A system overview is missing. Other survey have a similar approach [5, 67, 75, 148]. Their surveys allow researcher and practitioners new to the field to get a quick overview of existing approaches and algorithms.

Domain-specific surveys investigate the state-of-the art in FL from a use-case perspective [19, 143, 160]. They focus on Internet-of-Things (IoT) and edge scenarios, among others.

Taxonomy is the practice and science of categorization or classification. A taxonomy is a scheme of classification, especially a hierarchical classification, in which things are organized into groups or types. Li et al. [76] introduces a taxonomy covering multiple aspects of an FC system. For example, they highlight how to overcome different challenges by illustrating multiple optimization path. They look at each part individually and do not put them into perspective. It is not clear where in the FC system each component is used and how they interact with each other. Additionally, FL specific methods like client selection and aggregation strategies are missing. However, they describe attack vectors on FL training, such as model poisoning and inference attacks. The communication architecture focuses on a high level overview of different options. It neglects the communication protocols used in the back-end of different FL frameworks. A similar taxonomy is introduced by Abdul Rahman et al. [3]. Their focus is on basic FL building blocks, such as client selection and aggregation algorithms. With respect to communication costs, they briefly mention possible compression techniques to reduce network traffic, but they do not describe it in a broader picture of an FC system. On the application side, they give a comprehensive overview of different use cases and which aggregation algorithms and dataset were used. However, it is not clear which problem each use case was trying to solve (e.g., ML model performance, hardware/network utilization or privacy) and which FL system was deployed. The survey from Yin et al. [151] focus on identifying privacy leakages by introducing a 5W-scenario-based taxonomy. The 5W stands for different questions which give guidance in identifying and resolving attacks. They stand for: "who", "when", "where", and "why". The goal is to identify and quantify security breaches. Bellavista et al. [16] introduces a taxonomy which describes decentralized learning systems from a high-level and practitioner point of view with an emphasize on federated environments. It does not provide information about basic FL components such as client selection, aggregation and communication

Another overview of different FL components is given by Bonawitz et al. [18]. The focus is on building a centralized FL architecture with fallback aggregators similar to the hierarchical architecture. Besides the given example use case for FL they also describe how FL pipelines work in general and they mention some challenges with suggested solutions.

In our survey, we clearly separate between basic building blocks required to deploy a pure FC system and additional extensions. Besides this new taxonomy, we also add a meta layer, which describes motivation and the hardware environment. After describing each building block and extension separately, we show current trends in FC systems. The reader will have an overview at the end about which FC system is used to achieve which goal, which extensions are mostly used jointly and which FC systems are not yet investigated in sufficient depth. We separate our research and findings into FL and FA systems to highlight differences and shared characteristics.

3 FEDERATED COMPUTING DEFINITION

FC belongs to the domain of privacy-preserving computations. It introduces an information asymmetry between the server and clients creating a situation where no one know everything. Its goal is to extract information from distributed data sources without disclosing any raw data. In the process, private data stays on the device. Instead of sending raw data over the network, a node sends computing tasks to participating clients, which execute them and only send the respective

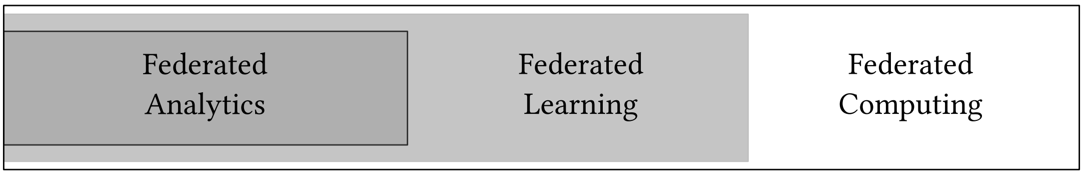


Fig. 1. Intersection of definitions for FC, FL, and FA. FC consists of FL and FA, whereas FL systems can contain FA characteristics. FA is a subset of FL and FL is a subset of FC.

computation results (e.g., machine learning model update such as weights and gradients) to the requesting server. An FC round consists of the following steps:

(1) Server selects participating clients
(2) Server sends computation task to each participating client
(3) Clients execute the computation task and send their individual update to the server
(4) Server aggregates all clients results
(5) For FL: Server distributes the aggregated result back to each client

FC includes FL and FA (see Figure 1). The difference between those two is the type of executed computing tasks and the number of aggregation rounds. FL focuses on machine learning (ML), which mainly consists of multiple aggregation rounds. The goal is to iteratively reduce a loss function and subsequently increase model performance. Conversely, FA leverages statistical operations, such as averages or sums. Each client executes them once, and the server draws conclusions from the data [134]. Some example use cases for FA include model evaluation or debugging [6, 67]. Simplified, FL consists of a combination of multiple FA steps [43]. FA emerged after FL. It started with Google using it to evaluate the accuracy of Gboard next-word prediction models by using captured data from users' typing activities on their phones. This is similar to accuracy evaluation [43]. Other FA use cases capture analytics for medical studies and precision healthcare or guiding advertisement strategies [43].

Another way to divide FC is by system focus. An FC system can focus on reasoning or learning, translating to FA and FL. Another term for the former is deductive systems and for the latter, inductive systems [45]. Deductive systems are also called "Good old-fashioned AI" and typically rely on rule-based or logical agents [52, 133]. Conversely, inductive systems try to learn based on the input data and are less prone to changes in the observed environment. An example for FL and FA are the collaborative optimization of an ML model (FL) and its subsequent distributed inference testing on client-side to obtain accuracy metrics for each client (FA). In this survey, we do not cover federated databases. Such systems also have a client-server architecture, which connects distributed databases to one another. The end user only sees one database, even though it consists of multiple ones. Other surveys describe the unique challenges and advantages of Federated Databases (FD) [13, 115]. We exclude FD from FC. FDs focus on improving query execution time and increasing availability and reliability of databases. On the other hand FC focuses on managing computations executed on any arbitrary dataset with a focus on privacy. Data in an FD system is openly available to all parties involved whereas FC limits access to client side data.

Depending on the FC system architecture, either a central server aggregates all results (central) or clients act as a server as well (peer-to-peer). Figure 2 provides an overview of three FC architectures. The first architecture in the figure shows a centralized FC approach. A server aggregates all results from the clients. The hierarchical architecture has an intermediate layer between clients and servers to increase redundancy. The devices in this intermediate layer act as servers and clients simultaneously. This layer adds robustness to the overall system. It can still generate insights even if one cluster fails. Additionally, it allows to cluster clients by categories, which could also be spatial

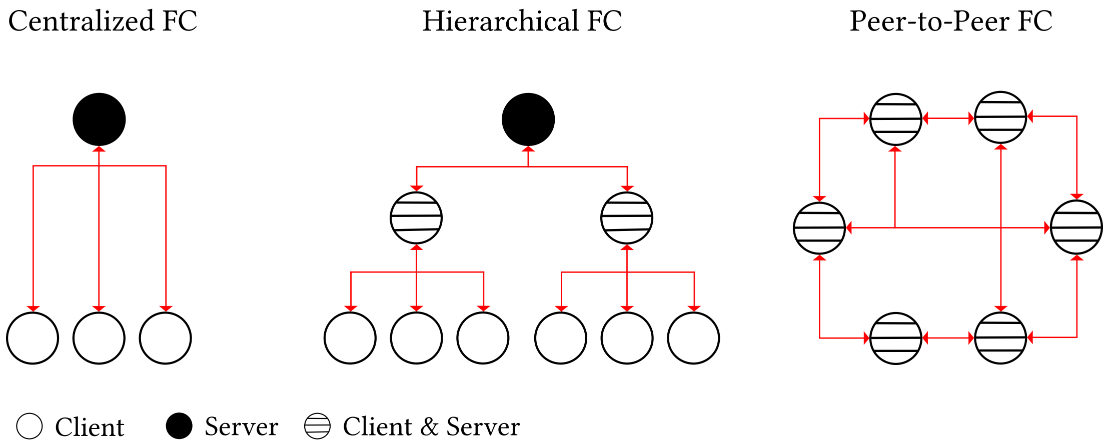


Fig. 2. Three FC architectures (centralized, hierarchical and peer-to-peer) with different aggregation server locations. Illustration inspired by [62]. A node can be a client, a server or both, which is indicated by the circle filling.

or user-based. A fully decentralized FC system runs peer-to-peer without a central server. Another term for this architecture is galaxy FC [63, 80]. Those systems are complicated to manage and it is challenging to converge to a given performance threshold if the ML optimization per client keep bouncing between multiple states instead of converging to a local or global optimum. All those architectures work for FA and FL.

3.1 Scenarios

FC systems can have different objectives, run on different devices, or with different data distributions. Table 2 summarizes commonly used ways to describe those scenarios. We structure them by a guiding question into three groups. All production-ready FC systems consist of a combination of one option per group - for example, a model-centric, horizontal, and cross-device FC system.

The first guiding question "Why" addresses the overall objective of the FC scenario. Does the user want to improve an ML model output or generated insights (model-centric), or does the user want to improve a data set and then make it available for others to train on (data-centric)? A model-centric approach tries to extract as much information as possible from a given dataset, e.g., by tuning hyperparameters of the model or tweaking the optimizer. On the other hand, data-centric is a quality-over-quantity approach and focuses on collecting and using only specific data suitable for a particular use case. It is a paradigm emphasizing that systematic design and engineering of data is essential for developing AI-based systems [64]. Therefore, performance improvements

Table 2. Three different groups of scenarios organized by a guiding question. Each scenario has a short description. An FC scenario consists of at least one scenario per group, for example model-centric, vertical, cross-silo FC.

Scenario	**Description**	**Guiding Question**
Model-Centric Data-Centric	Curating or improving output Curating or improving input	Why?
Horizontal Vertical	Same features with different users Different features with same users	How?
Cross-Device Cross-Silo	E.g. mobile phones, IoT devices E.g. hospitals, manufacturing sides	Where?

result from improving the quantity and quality of the data instead of changing the underlying model architecture. FL and FA require data. Gröger emphasizes challenges for data management, governance, and democratization in an industry context, which highlights the potential for data owner to make a data-driven approach more feasible in the future [58, 59]. The current focus in the literature and the industry is on model-centric scenarios, as seen in Section 7. This approach can start with a pre-trained ML model or from scratch.

The next group of scenarios considers the data distribution of the clients. In horizontal FC (HFC), all participating clients have the same features, but different users. An example is predicting the next words written on Google's Gboard [18]. The users per client differ, but the features (word predictions) are the same. Vertical FC (VFC) is the opposite. Each client monitors different features, but they share the same user. An example application is the finance sector, where retailers and banks store historical data on the same person but with different features [148].

Lastly, we group each scenario by the location of the participating devices. A cross-device scenario uses distributed devices with a high degree of individual ownership. Those devices could be mobile phones or wearables, such as smartwatches or home assistance systems. Cross-device scenarios work with up to thousands of devices, which all can have a different owner, generally private persons. On the other hand, cross-silo scenarios leverage data allocated on devices or entities with much less diverse ownerships. Some examples come from the healthcare and manufacturing domains. Multiple hospitals or pharmaceutical companies can collaborate and jointly generate results in an FC fashion [105, 145]. The training process still runs on dedicated hardware, which could be mobile phones. Therefore, there needs to be a clear cut between the definition of cross-device versus cross-silo scenarios. Cross-silo is mainly limited to a few hundred participating clients due to the organizational complexity, and the owners of the clients' hardware are businesses.

3.2 Problems

FC systems increase data privacy due to limiting data access. However, FC faces a multitude of problems. The assumption for FL is that each client has labelled data and therefore follows a supervised training. We cluster the problems into three categories:

(1) Improve insights or ML performance,
(2) Improve privacy or security,
(3) Improve hardware or network utilization.

The first problem is mainly due to an uneven distribution of features and labels on the clients. In an FC system, the server has no data access on client side. It only knows some meta information, such as image resolution or for time series the respective units of each column. In an ideal scenario all clients' datasets have similar statistical attributes, which yield similar results of the executed computation tasks as well. However, in real scenarios some clients' datasets might be biased towards certain labels. Therefore, the same execution task can yield different results per client. This is called non-independent and identically distributed (non-IID) data. Aggregating results based on non-IID data is a challenge due to its impact on the final result on server-side. For some cases the aggregation of individual client updates can result in worse results compared to a traditional centralized approach. Choosing a subset of available clients (Section 4.1.1) or a suitable aggregation strategy (Section 4.1.2) can improve the generated insights. Another issue in this context are clients dropping out during an FC process. This can also result in a non-IID scenario even though clients were properly selected at the beginning of the process. furthermore, the entire run can get delayed, because the server is waiting for all clients to finish their execution task.

The second problem mainly copes with the possibility of attacks on an FC system to infer raw data from the individual clients model and data leakage during the process. In general, there are two types of attacks: Black-box and white-box attacks. In black-box settings, the adversary's access is limited to the model's outputs only. The adversary can query the model with an arbitrary input x and obtains the prediction vector f(x). In white-box settings, the adversary has full access to all components of the model. The access includes the model's architecture, parameters, and hyperparameters. Also, the adversary can inspect intermediate computations and prediction vectors. Attacks can occur at different stages (input data, training and inference phase) during an FC round. An attack during the input phase tries to poison the data in such a way that the final model is impaired. This attack originates from a participating client. In the training phase, participating clients can try to infer data based on the updates they get from the server or alter the model on purpose to again impair the final model. Inference attacks happen during the training process or at the end. Their goal is to leak information about the training data and not to impair any data or models.

The third problem is due to the distributed nature of FC systems. Each client sends its updates either to a central server or other clients (peer-to-peer). Reducing this network overhead without interfering with result accuracy is one research area. Additionally, considering each clients individual hardware and data in the scheduling of the training helps to use computational resources more efficiently.

3.3 Components

FC systems consist of different components (basic building blocks), which can be enhanced with extensions from other domains (Figure 3). The goal is to a build an FC system to solve one or all above mentioned problems. There are multiple options available for each basic building block and extension. A systems' performance depends on how well all components work together for the respective use case. The following paragraphs shortly describe all required basic building blocks for a FC system and two widely-adopted types of extensions. A detailed description of all components follows in Section 4.1 and Section 4.2.

We separate those basic building blocks into hardware and software components. The hardware consists of devices hosting private data sets and an aggregation server. For this definition, the devices' computational resources are not of importance. The computational resources can vary from computationally weak edge devices to strong GPU servers. The software part consists of three components: Client selection, aggregation strategy and communication protocol (e.g., gRPC, WebSocket, HTTP). Figure 4 provides an overview on how these components work together with some options per step. To further improve different aspects of such an FC system it is possible to extend it with methods from other domains. For each building block and extension different options exist.

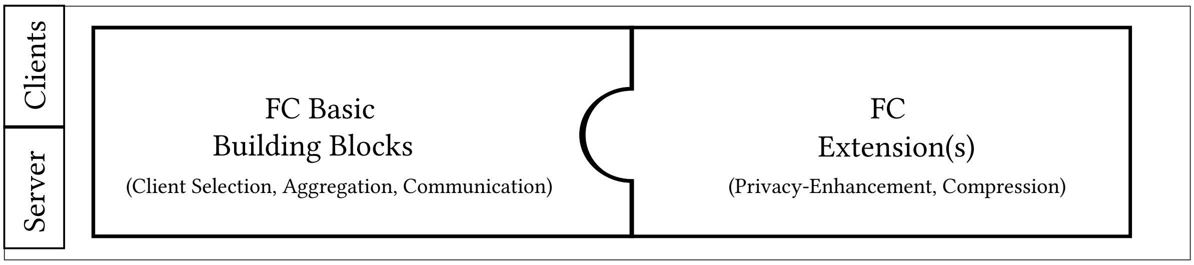

Clients
Server
FC Basic
Building Blocks
(Client Selection, Aggregation, Communication)
FC
Extension(s)
(Privacy-Enhancement, Compression)

Fig. 3. Interaction between basic building blocks for FC systems and optional extensions. The basic building blocks consist of hardware (devices hosting data set and an aggregation server) and software components (client selection, aggregation strategy and communication protocol).

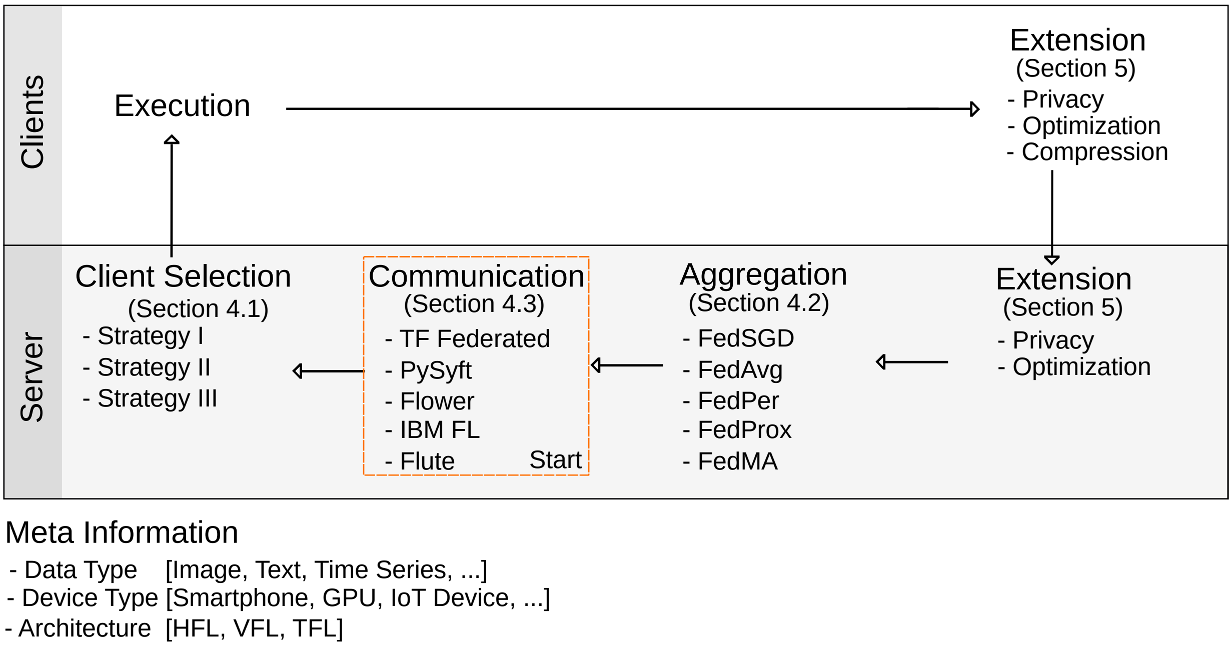


Fig. 4. Framework to describe how different building blocks in an FC system work together. The illustrated pipeline focuses on FC basic building blocks and it shows some example options for each block. Additionally, it shows some options for the systems meta information.

In the following paragraphs, we shortly explain the three software components individually, before going into detail in their respective section. FC works with distributed clients. The number of participating clients can vary from a few to multiple thousands. Following a quantity over quality approach might interfere with the above three mentioned problem categorize (Section 3.2). Therefore, it is necessary to introduce constraints on the client selection process. Those constraints can be of different nature and each of them tries to improve at least one problem. For example, taking clients with a similar data set size reduces the idle time of individual clients. This approach decreases the risk of clients dropping out during the training and helps to improve FC performance and hardware utilization. Another example for a client selection constraint is to pick only geographically close clients to reduce network failures. In Section 4.1.1 we go into detail on how different client selection algorithms work and which advantages and disadvantages they have.

Each client sends its results either to a central server or to another client (peer-to-peer). Those results have to be combined to leverage each individual clients' insights. This can either be an average of all results or more complex operations. Some other examples are to compute weighted averages or to cluster clients into different groups. Taking an average is called Federated Averaging (*FedAvg*) and it is one of the first aggregation algorithms [86]. The focus of an aggregation algorithm is to improve the generated insights. However, as we see later in Section 4.1.2, some aggregation algorithms are prone to privacy and security risks (e.g., reverse-engineering the raw data with the client updates) or they introduce bottlenecks in the hardware or network utilization.

In an FC system, each node (server or client) has its own unique IP or identifier. During a training round they have to communicate with each other and send data to one another. Engineers leveraged different network protocols and serialization methods in the context of FC. Building an FC system works in theory with all kinds of combinations of communication protocols and serilization methods. In Section 4.1.3 we explain briefly how different communication and serilization methods work. Then we introduce currently available FC frameworks with their respective architectures.

3.4 Meta Layer

Besides the hardware and software components other more holistic aspects are also important to classify an FC system. We split those aspects into three different categories:

(1) Problems (see Section 3.2),
(2) Hardware environment,
(3) Economic feasibility study

In Section 3.3 we already introduced the first meta layer "Motivation" with its three levels of motivation. Improving insights with an FC system mainly faces the potential issue with non-IID data on the clients. In contrast to a centralized approach, it is difficult to balance the distribution of features and labels during an FC run. In the literature the focus is on three different approaches to improve the generated insights. The first one works with avoiding non-IID in the first place by using a suitable client selection process (Section 4.1.1). Secondly, more and more specialized and tailored aggregation algorithms are developed to balance the clients updates (Section 4.1.2). Lastly, leveraging optimization techniques developed originally for other applications can help to reduce the negative impact of unbalanced data sets on the final results.

The next meta layer considers the environment an FC system runs in. Available options are:

(1) simulated FC system (clients and server) on one device,
(2) virtual machines (VM),
(3) physically separated devices or
(4) a combination or hybrid.

Choosing a suitable hardware environment requires a balance between development speed, maintenance and boundary conditions. For example, testing the effect of different client selection algorithms on the final result (e.g., machine learning prediction performance) does not depend on the network bandwidth. Therefore, it is feasiable to simulate all clients and the aggregation server on one device. On the other hand, tuning the way clients communicate with the server or each other requires a realistic emulation of network communication to understand the impact of different approaches on the communication behavior.

Lastly, all FC systems compete with centralized state-of-the-art approaches. Therefore, they have to have advantages to be widely used in real-world applications. Those can be legal, public relations or trust issues. However, all of them boil down to an economical impact for the respective stakeholder. Currently, the focus of the literature is on technical aspects and less on economical feasibility. However, this area is growing and some authors already mapped some existing economical frameworks on FC system to investigate different scenarios.

4 MODULES

4.1 Basic Building Blocks

The following sections describe each basic building block of an FC system individually in more detail. Those are client selection (Section 4.1.1), aggregation (Section 4.1.2), and communication (Section 4.1.3). We give an overview of existing approaches and cluster them by different criteria.

4.1.1 Client Selection. FC systems work with distributed clients and the number of those can range from just a few to multiple thousands of devices. Choosing a suitable pool of clients helps alleviating one or multiple of the challenges introduced in Section 3.2. Currently, most FC systems work in the order of magnitude of 40 clients. For those cases, the client selection is done manually and it considers either 100 % of the available clients or randomly picks a subset. However, non-IID data on the clients or geographically separated clients can introduce issues with respect to knowledge gain for example due to non-converging ML models, slower network traffic or higher latencies. Therefore, we cluster the goal of client selection algorithms into three categorize:

(1) Decrease training time,
(2) Decrease network traffic,
(3) Improve generated insights.

There are different strategies available to achieve either one or all of the above listed goals. The deployed strategy depends on the number of clients, their computational resources and the available network. Nishio et al. [92] proposes *FedCS* and focus on mobile edge devices with limited and heterogeneous computational resources. Clients are selected if their time to run one aggregation round is below a given threshold. This approach does not allow a pre-selection of clients and needs at least one round of training. Abdulrahman et al. [2] follow a similar approach with *FedMCCS*, which additionally considers CPU, memory and energy constraints. These algorithms are not suitable for FA, which only has one aggregation round. Besides computational resources, researchers select clients depending on their network connection. The experiments of Xu et al. [146] run with clients being connected all to one wireless link. Instead of having a throughput maximization approach they follow an optimization strategy to improve the systems' learning performance under finite bandwidth and energy constraints. It is also possible to have a more fluctuating set of clients which change during FL training. Here, a selection criterion can be the current loss of each client to increase convergence speed of the trained model [27].

4.1.2 Aggregation. After every round, each participating client sends an update to an aggregation server that merges all updates to a single model. A round refers to either a single event, a batch or an epoch. Such updates can be scalars, vectors, or matrices containing for example an ML model's gradients, weights, or biases for FL systems. An aggregation strategy solely focuses on how to process such updates. The server can apply statistical techniques, such as average or mean, or filter based on thresholds. All those modifications can run on an element-wise order or follow another arbitrary order. Some proposed strategies also change the client selection process. All aggregation strategies aim to address the potential issue with non-IID on the client side. The challenge is generalizing client updates on the server side by simultaneously personalizing the models on the client side. Therefore, improvements for aggregation strategies apply to either the server or client side or both.

FedSGD and *FedAvg* are the first aggregation algorithms designed for FL systems [87]. They run element-wise calculations on the input. The equations for calculating *FedSGD* (Eq. 1) and *FedAvg* (Eq. 2) on the server side differ by the number of training rounds on the client. For *FedSGD*, each client takes one step of gradient descent and then it sends its update to the server. The server takes a weighted average of all updates. *FedAvg* differs from *FedSGD* by running more iterations on each client before aggregating the results. The learning rate is given by η, K is the set of clients, t describes one time step, w is the model, n the number of data points on the client and the Nabla operator ∇ converts a field of scalars to a field of vectors.

$$FedSGD: \quad w_{t+1} \leftarrow w_t - \eta \nabla f_k(w_t) \quad \text{with} \quad \nabla f(w_t) = \sum_{k=1}^{K} \frac{n_k}{n} g_k \tag{1}$$

$$FedAvg: \quad w^k \leftarrow w^k - \eta \nabla F_k(w^k) \tag{2}$$

McMahan et al. [87] introduces three parameters to describe an aggregation strategy. The first is C, representing the fraction of clients participating in the computations on each round. It ranges from 0 to 1, with one referring to all available clients. An FC system consists of at least two clients to enable some form of aggregation and at least some protection against reverse-engineering the raw data based on the model updates. The second parameter is E, which refers to the number of training passes on each client before aggregating their updates. It is an absolute value and refers directly to the number of local training rounds. Third, the parameter B describes the size of the mini-batch in relation to the clients' dataset size. "1" refers to using the entire local dataset as one batch. Besides those three parameters, Arivazhagan et al. [10] introduced the parameter K. It describes the number of layers of a neural network trained exclusively locally. Those layers do not change after receiving an update from the server.

Tweaking those parameters allows the development of new specialized aggregation strategies designed to work in the dedicated system environment. Table 3 overviews various aggregation strategies and their respective parameter settings. Those aggregation strategies solely focus on FL and the challenge with non-IID data on the client side. For example, *FedPer* and *FedDist* focus heavily on neural networks. The former only updates a given number of layers and keeps others local. This approach aims to make the models generalizable for all clients by simultaneously keeping a certain degree of personalization. The latter aggregation strategy calculates the Euclidean distance between neurons in a neural network to identify diverging neurons. Focusing on neural networks eliminates those aggregation strategies from being used in FA systems.

Besides *FedDane*, all other aggregation strategies have a pre-defined set of clients at the beginning of the first FL training round. However, a pre-defined set of clients is still prone to clients dropping out during training. The aggregation server might replace those clients with new ones. Therefore, the set might change over time, but the defined set is not a crucial part of the aggregation strategy. At the same time, *FedDane* incorporates the change of selected clients into its core structure. It approximates the gradients using a subset of gradients from randomly sampled clients [78]. It achieves theoretically better results than *FedAvg*, but it underperforms in actual experiments and requires double the number of communication rounds due to adjusting the clients based on an optimization problem.

Conversely, *FedProx* varies the number of local training iterations per client before aggregating the results. *FedProx* differs from *FedAvg* by allowing for a variable number of training iterations on the client side based on their available dataset and computational resources. This approach results in some clients training more rounds than others. The server aggregates those partial solutions [79].

Table 3. Classification of FL aggregation strategies depending on C (number of clients participating in the training with one being 100 %), E (number of training iterations before aggregation), B (local mini-batch size with one referring to the entire local dataset as one batch), and K (number of unchanged / frozen layers).

	C	E	B	K	Note
FedSGD [87]	≤ 1	1	1	0	-
FedAvg [87]	≤ 1	> 1	≤ 1	0	-
FedPer [10]	≤ 1	> 1	≤ 1	> 0	Clients update only a subset of NN layers and train the other layers locally
FedProx [79]	≤ 1	Varies per client	≤ 1	0	-
FedMA [136]	≤ 1	> 1	≤ 1	0	-
FedAT [21]	≤ 1	> 1	≤ 1	0	Clusters clients based on latency
FedDane [78]	Varies per round	> 1	≤ 1	0	-
FedZIP [84]	≤ 1	> 1	≤ 1	0	Compresses updates
FedDist [39]	≤ 1	> 1	≤ 1	0	Calculates euclidean distance to identify diverging neurons
FedMAX [26]	≤ 1	> 1	≤ 1	0	Max entropy regularization to equalize activation vectors in an NN layer

The parameters of the aggregation strategy for *FedAT* are the same as for *FedAvg*. However, it clusters the available clients based on latency to improve the overall training time and test accuracy by having clients with similar time and dataset constraints [21].

4.1.3 Communication. FC systems work with remote clients. It describes a distributed client and server architecture with data being transferred between those nodes. This section describes FL frameworks and their principles in more detail.

Multiple organisations, institutes or other stakeholders develop FL frameworks. Some of them have a specific focus on a certain domain, some are not for commercial use and others are not further developed. Several frameworks have been developed, such as *PySyft* from openMined [108], *TensorFlow Federated* from Google [18], *IBM FL* from IBM [83], *FedAI/FATE* from WeBank [57], *Clara SDK* and *FLARE* from Nvidia [125, 126], *FedML* [62], *Paddle FL* from Baidu, *Fed-BioMed* [107], *Flower* [17], *FLUTE* from Microsoft [33], *Substra* from Owkin [54], *OpenFL* from Intel [50], *FederatedScope* from Alibaba [144] *APPFL* from the Argonne National Laboratory (USA) [109], and *Vatange6* [121]. *LEAF* [20] provides tools to benchmark different pre-selected models in a FL setting. Karimireddy et al. [68] assess the strengths and weaknesses of 14 different FL frameworks, ranging from supported data distributions and communication topologies to available built-in advanced privacy and security features.

We divide the required components into communication protocols and serialization methods. Serializations methods are either binary or contextual. The former converts an input into a series of bytes and the latter uses data formats such as JSON or XML to transfer information. Contextual serialization contains details about the data's structure and purpose, making it simple for a human reader to interpret and comprehend. Table 4 provides an overview of FC frameworks with their respective components. Not all frameworks state exactly which protocol and serialization methods they are using and instead refer to a vague statement in their documentation saying that messages are sent over the internet. All frameworks are consistently promoted for FL. However, they can be adjusted to also deploy an FA system. Eight out of twelve frameworks use *gRPC* in combination with the binary serialization Protobuf. The *gRPC* protocol has no browser support. *WebRTC* and *WebSockets* support browser integration and are therefore used for *PySyft* with its browser based

Table 4. FC frameworks with their communication protocols and serialization method(s). All frameworks state in their white papers or documentation a focus on FL. However, they might be adjustable to FA use cases.

	Protocol				**Serialization**			
	gRPC	WebSocket	HTTP	GLOO	Pickle	JSON	Protobuf	FOBS
APPFL [109]	✓		✓		✓		✓	
FedBioMed [107]	✓					✓	✓	
FedN [40]	✓		✓			✓	✓	
FedScope [144]	✓						✓	
Flower [17]	✓						✓	
Flute [33]				✓	✓			
IBM FL [83]			✓		✓	✓		
FLARE [126]	✓						✓	✓
OpenFL [50]	✓						✓	
PySyft [108]		✓					✓	
TFF [18]	✓						✓	
Vantage6 [121]		✓	✓			✓		

Duet implementation. Other outliers are the FL frameworks of IBM and Microsoft, which use *HTTP* and *GLOO* as their respective network protocol. It is not recommended to use Pickle serialization in environments with untrusted parties due to potential security issues. The documentation of Pickle emphasizes the fact that it is possible to construct malicious pickle data which could execute arbitrary code during unpickling [51]. Instead, they recommend to either use `hmac` for message authentication in Python or to switch to JSON serialization. *IBM FL* offers both serialization methods.

4.2 Extensions

4.2.1 Privacy Enhancement. Privacy enhancing techniques try to reduce data leakages in an information flow. An information flow describes the communication between a server and the clients and what each node is doing with the received information. For example, a client or a server could try to infer information about the underlying raw data with the aggregated computation outputs. All information flows have a trade-off between transparency and privacy [12, 130]. The transparency is high if a server has direct access to the raw data and consequently privacy is low. If no data is shared at all there is a high level of privacy, but no transparency and knowledge gains. Privacy-enhancing techniques allow to keep a certain level of privacy while improving transparency as well. Figure 5 illustrates this trade-off. The left plot shows a naive trade-off without leveraging any privacy-enhancing techniques. The more a user shares the lower is the privacy level. However, privacy-enhancing techniques, such as the ones mentioned in Table 5, allow to keep a certain privacy level and simultaneously increase the transparency or the gained insights.

Data leakages can be clustered into two different categories: Copy problem and bundling problem [122]. The copy problem describes the loss of control when giving somebody a copy of a dataset. There can be legal boundaries describing to which extend the data can be used. However, enforcing those constraints is challenging. The bundling problem describes information leakages due to an information content, which contains more information than the actual requested one, but they cannot be separated from each other. Therefore, it is possible to directly get additional information, which is not needed or it is possible to do backwards inference to the input data based on the output. An example is the age verification of somebody who wants to buy alcohol. The cashier asks for an ID and verifies that the customer is above the legal age for drinking. However, in the process additional information is leaked such as the name or address of the customer. Those information

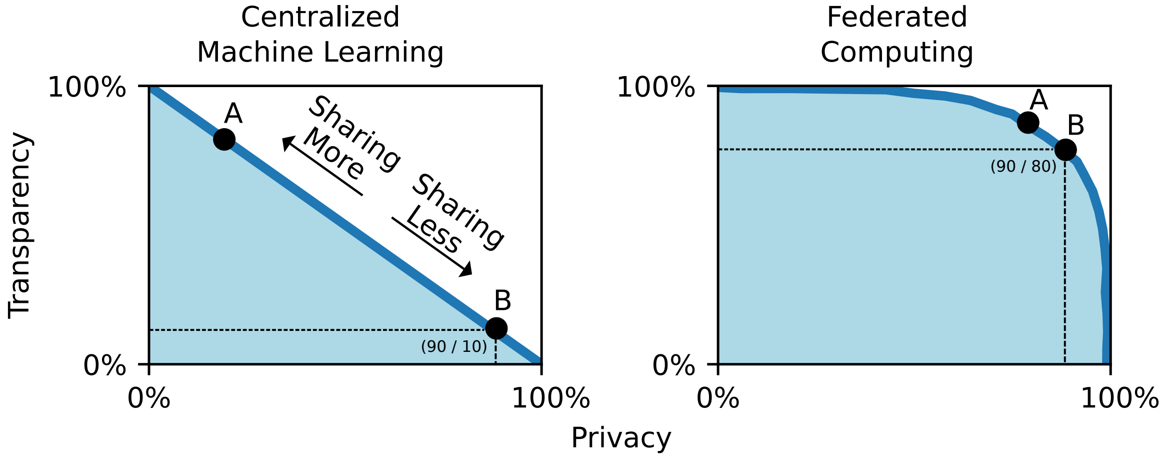


Fig. 5. Two Pareto trade-offs between data transparency and privacy. Everything on the curve represents a privacy preserving outcome. The level of privacy for Point B is in both cases the same, but the level of transparency increased in the right trade-off due to privacy-enhancing techniques [98, 130].

Table 5. Privacy enhancing techniques to either protect the input or output against privacy attacks.

Input Privacy	Output Privacy
Secure-Multi-Party Computation	Differential Privacy
Homomorphic Encryption	Student-Teacher Learning (e.g., PATE [96])
Public-Key Cryptography	
Federated Computing	

are not required to complete the age verification process. An example about backward inference is the de-anonymization of some users in a dataset from Netflix movie ratings by comparing rankings and timestamps with public information in the Internet Movie Database [91]. Other examples leveraged anonymized internet usage patterns [90] or location data [106, 153] to infer information about individuals. First, we describe FC extensions trying to solve the copy problem and then we focus on solutions to the bundling problem.

Input privacy tries to solve the copy problem. Its goal is to keep computation inputs of individuals secret from all parties involved. Table 5 shows example techniques for achieving input privacy, which mainly originate from the cryptography domain. It is possible to combine them, e.g., FC + homomorphic encryption (HE). This implies that theoretically anybody can run computation tasks on data without the need for direct access to it. This allows an information flow without the need for a trusted third-party. [130, 138]

Secure-Multi-Party Computation (SMPC) enables multiple clients to jointly compute a result without sharing their inputs. They obfuscate their inputs with random numbers, which are randomly distributed to other participating clients. Some disadvantages of this approach are an increase in network traffic due to multiple clients communicating with each other instead of only sending updates to the server and the risk of data loss due to clients dropping out. Most SMPC algorithm rely on a pair-wise collaboration. If one participant of such a pair drops out during the process the added random numbers do not cross out, resulting in a false output.

HE allows to run computations on encrypted data. This approach increases input privacy at the cost of computational complexity. The output of a computation is still encrypted and only the server is able to decrypt the generated results. So there is a trade-off between SMPC's high network overhead or HE's high computational requirements.

Output privacy tries to solve the bundling problem by preventing backwards inference or reverse-engineering of the input based on the output. This is addressed by access control or statistical disclosure control. The former imposes restrictions on who has access to the data. The latter relies on a combination of suppression, perturbation, randomization and aggregation of data [104]. A widely used approach is differential privacy (DP) and related techniques [130].

Table 6. Core components of a DP algorithm with some examples. An DP algorithm consists of one of each component. The list with examples is not exhaustive.

DP Definition	Randomization Mechanism	Sampling Technique
Pure DP	Gaussian	Poisson
Approximate DP	Laplace	Uniform
Concentrated DP		
Zero concentrated DP		
Gaussian DP		
Rényi DP		

DP is a mathematical framework that provides stringent statistical guarantees about the privacy of individuals participating in a database [36–38]. Its basic idea is to artificially add noise to a data set without changing its statistical properties. It provides provable guarantees about the amount of private data an adversary can infer by observing the outputs of an algorithm. It can either be deployed on client (user-level privacy) or server-side (record-level privacy). Table 6 provides examples for the three core components of each DP algorithm. This summary highlights the complexity of choosing a suitable DP algorithm and it is not an exhaustive list. Several DP definitions exist, with each having different theoretical privacy guarantees. In theory an DP algorithm consists of any combination of those three components.

4.2.2 Compression. Running an FC system requires continuous communication between the server and clients. The resulting network traffic can be higher when compared to transferring the raw data. Also, clients' weak network connections might result in dropouts during training. In general, there are two approaches to reduce the network traffic. First, decrease the number of aggregation rounds. Second, reduce the data transfer itself to decrease the overall network traffic. However, both approaches (aggregation frequency and update size) can negatively impact the ML model performance. Therefore, there is a trade-off between those two metrics. Since the upload speed is significantly lower than the download speed, most papers focus only on compressing the gradient updates that clients send each round and leave aside the global server updates. However, in real-world applications, additional servers balance client requests. These so-called parameter servers lead to an increased communication complexity of the global updates. Therefore, an FC system should implement compression in both directions. [124].

A server can aggregate after every 10th batch instead of after every batch or decrease the aggregation frequency even further by aggregating after every n-th epoch. However, the longer clients train locally, the more biased they become towards their local dataset. It becomes more challenging for the aggregation server to merge highly personalized models into one general model, which is helpful for all clients. An experimental study with a dataset containing electrical signals used to train four different ML architectures quantifies a reduction in overall model accuracy when increasing the aggregation steps from one batch up to ten batches [112].

Data compression techniques have a wide range of applications that are not exclusive to FC. However, those extensions boost hardware and network utilization in FC systems. Reducing network requirements makes FC use cases in network constraint environments more feasible. For example, mobile phones running on metered mobile networks or IoT devices connected with narrow bandwidth IoT (NB-IoT) or low range wide area network (LoRaWAN) benefit from smaller updates

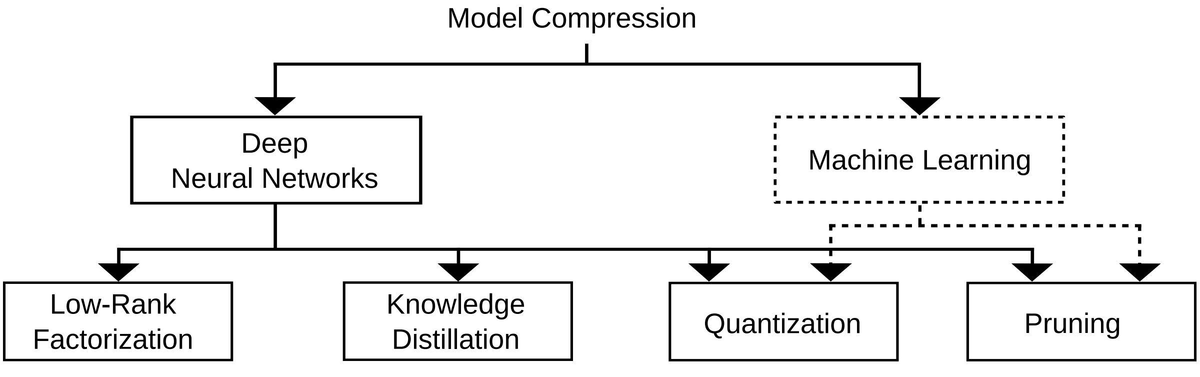


Fig. 6. Model compression techniques for deep neural networks and machine learning (e.g., Support-vector-machines and decision trees) [29].

due to cost and network constraints. NB-IoT and LoRoWAN enable smart city applications due to their long communication range, but they have constraints concerning the maximum package size and energy consumption.

Multiple different model compression techniques exist that reduce storage size (memory or disk), decrease energy consumption for inference, or increase inference speed. Figure 6 summarizes four common model compression techniques. Knowledge distillation transforms a large model (teacher) into a smaller and lighter model (student) by letting the student learn from the teacher. The student model tries to learn the generalization capabilities while staying smaller. Quantization leverages different data types with their respective precision. For example, using 32-bit floating-point numbers instead of 64-bit ones saves storage. An advanced version of quantization is the usage of clusters. A cluster consists of either the same values or multiple ranges of values. Instead of storing all numbers, the model only stores those clusters. Some adaptions of quantization in the context of FL are FedPAQ [103] and UVeQFed [118] among others [9, 25, 117]. Model pruning removes weights and neuron connections that are either below a given threshold or do not contribute much to the final result. Model pruning reduces complexity and the number of computations to run. The latter is especially beneficial for edge device scenarios with limited computational resources or battery-powered devices [65, 66, 152, 157]. Lastly, low-rank factorization factorizes a matrix into a product of two matrices with lower dimensions to reduce complexity. [29]

5 FEDERATED COMPUTING SYSTEMS

Based on our proposed framework in Section 3 we categorize a wide range of use case / application papers. Table 7 provides a few examples of our literature research. First, we identify the motivation of the paper or the challenges it tries to solve. The numbers (1), (2), and (3) refer to the definitions based on Section 3.2. The definition of the environment is given in Section 3.4 and we describe the different available scenarios in Section 3.1. The next three columns (framework, client selection, and aggregation strategy) capture the basic building blocks of the FC system. Instead of using an FC specific framework, some papers leverage ML framework such as *PyTorch* or *TensorFlow*, or they do not specify the used framework at all. A NaN indicates missing information about the framework or aggregation strategy. The first row in the table uses the FL framework *TFF*, which stands for *TensorFlow Federated*. All three examples in Table 7 run without any kind of privacy-enhancing or compression extension. Lastly, we summarize the implemented extensions. With our framework we capture an FC system completely, with all its unique components. We visualize our findings and describe them in detail in the following sections. The goal is to identify FL and FA system

Table 7. Example FL systems classified based on our framework. The numbering convention in the motivations columns follows our definitions from Section 3.2 ((1) Improve insights or ML performance, (2) Improve privacy or security, (3) Improve hardware or network utilization)

Motivation	Environment	Scenario	Framework	Selection	Aggregation	Extension
(1)	Single node	model-centric horizontal cross-device	TFF	Manual	FedAvg	None
(1)	Single node	model-centric horizontal cross-device	PyTorch	Manual	NaN	None
(3)	Single node	model-centric horizontal cross-device	NaN	Manual	NaN	None

configurations, which are either extensively implemented in the research community or are lacking further investigations. This enables the identification of research trends and gaps.

5.1 Federated Learning Systems

The focus in research is currently on FL. It covers applications from all domains, such as energy, mobility, and healthcare. Table 8 provides an excerpt of our literature research for FL systems. It shows all surveyed papers, but not all captured characteristics. Papers can appear multiple times if they have more than one motivation. For example, a paper can improve the ML performance and simultaneously try to improve hardware utilization. The y-axis in Table 8 contains the motivation and client selection approaches. Section 3.2 describes each of those motivations in detail. We separate client selection strategies into manual, resource-aware, and loss-aware. In the first strategy, the authors either use all available clients or manually define a client set for training. The other two client selection strategies define a client set for training based on the available resources on the client side or on how the loss behaves during a training round. The x-axis shows the experiment environment. It is unknown when the publication does not state the hardware used for the experiments. If the publication contains information about the hardware, it is either a single node hosting the server and clients or multiple nodes. The latter allocates one piece of hardware for the server and each client, respectively. The hardware includes edge devices, such as Raspberry Pi, dedicated servers, or GPUs. The distribution of a multi-node environment is either located in one location or spatially distributed.

The majority of publications focus on improving machine learning prediction performance by running experiments on either an unknown system or an environment with all parties (server and clients) being simulated on one device. A one-device approach with simulated clients reduces the

Table 8. Categorization of FL papers into three groups. A reference can appear in multiple groups. Environment describes the number of hardware used for an experiment. Motivation refers to three categories introduced in Section 3.2 ((1)) = Improve insights or ML performance, (2) = Improve privacy or security, (3) = Improve hardware or network utilization). Client selection is either done manually, resource-aware, or loss-aware.

Environment Motivation		Unknown	Single Node	Multiple Nodes
(1)	Manual Selection	[4, 30, 48, 73, 74, 80, 81, 89, 128, 131, 147]	[1, 24, 32, 35, 46, 60, 71, 77, 99, 110, 111, 123, 135, 135, 150, 150, 162–164]	[8, 11, 41, 70, 94, 101, 113, 157]
	Resource-aware Selection	[146]	[92]	
	Loss-aware Selection	[28]		
(2)	Manual Selection	[30, 80]	[98]	
(3)	Manual Selection	[80]	[46, 71, 137, 149, 154, 159]	[11, 70, 88, 100, 113, 114]
	Resource-aware Selection	[146]	[92]	[2]
	Loss-aware Selection	[28]		

complexity and overhead of the system. Having a realistic network traffic or monitoring the CPU cycles are not necessary to investigate the effect of different ML and aggregation strategies on the final models' accuracy. A major challenge of FL systems is an uneven distribution of labels on the clients (non-IID). Therefore, it makes sense to first focus on developing ML prototypes which achieve satisfactory accuracy before increasing the systems' complexity by introducing hardware, network, or energy constraints. Most papers do not specify which environment they run on (unknown). Figure 7 shows the distribution of motivation (a) and experiment environment (b) for the surveyed FL systems. 49 out of 50 papers focus on improving machine learning prediction performance. Eight of those 49 papers also try to improve hardware or network constraints. The next prevalent motivation is to improve hardware or network utilization (motivation (2) in Figure 7). To capture changes in those metrics, it is paramount to run the experiments on distributed devices. However, only 26 % of FL papers run their experiments on multiples nodes. Papers using one node (48 %) or an unknown environment (26 %) often test an optimization function with respect to hardware or network utilization improvements. So, instead of measuring actual utilization rates, they theoretically estimate them.

The preferred client selection algorithm and aggregation strategy are manual and *FedAvg*, respectively. There is also a clear trend towards using unspecified (unknown) aggregation strategies 12 %) or *FedAvg* (59 %). The latter is a simple aggregation strategy, and it is mainly used in its pure form. There is a huge variation of aggregation strategies. Our survey captures at least 11 different strategies, and the majority of publications work with only one aggregation strategy. Seven publications deployed at least one aggregation strategy. We visualize its distribution in Figure 7 by counting all occurrences independent of its publication. Only a few publications adopt aggregation strategies developed in other work. For example, only two publications leverage *FedAvgM*. Reasons for a lack of wide adaptations of new aggregation strategies, besides *FedAvg*, are manifold. The developed aggregation strategy might be to specialized for a specific use case or dataset. Other reasons can be a lack of documentation or source code. Having a standardized way of describing an aggregation strategy highlights the differences and advantages of specific strategies. The summary in Table 3 in Section 4.1.2 is a starting point for expanding it to more detailed definitions of the in- and outputs of each aggregation strategy to better reproduce and understand its functionality.

Also, 94 % of our surveyed FL papers capture the same scenario, which represents a model-centric, horizontal FL and cross-device architecture. The exceptions use cross-silos instead of cross-devices. They come from the healthcare domain and capture different institutions instead of multiple devices.

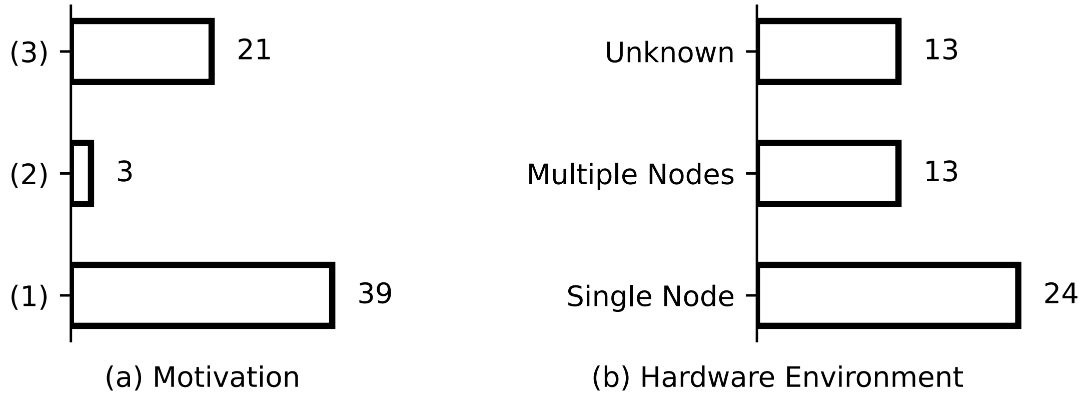
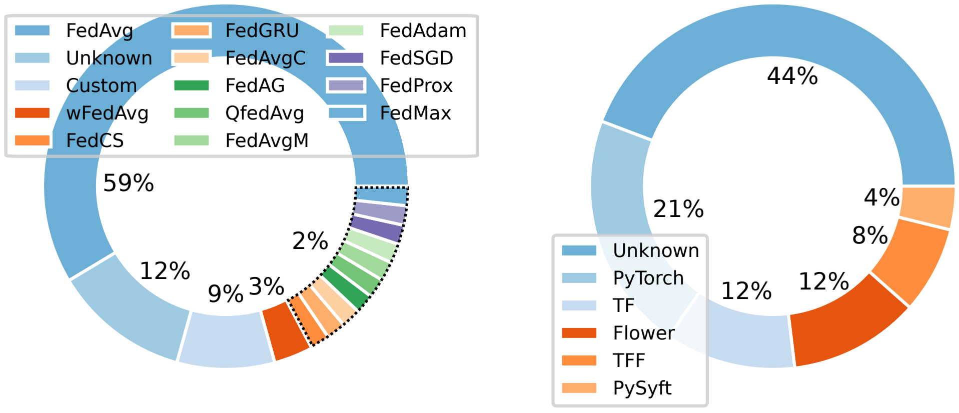


Fig. 7. Distributions of the problem definition (a) and hardware environment (b) of summarized FL systems. The numbers follow the structure introduced in Section 3.2 ((1) Improve insights or ML performance, (2) Improve privacy or security, (3) Improve hardware or network utilization).



(a) Aggregation Strategy

(b) Framework

Fig. 8. Distributions of aggregation strategies (a) and frameworks (b) for FL papers. We surveyed in total 50 FL papers published after 2020.

A cross-silo architecture is similar to a cross-device one. Differences between those two architectures are rather on legal basis instead of a technological one. However, implementing data-centric instead of model-centric or vertical FL instead of horizontal FL introduces new challenges and increases the overall complexity. Data-centric architectures improve the ML model accuracy from an FL system by altering a clients' dataset and vertical FL merges datasets with different features. Therefore, currently the focus is on model-centric and horizontal FL systems to keep the complexity low. However, this highlights a lack of research in more complex FL systems incorporating datasets with different feature sets.

Figure 8 provides an overview of deployed frameworks and aggregation strategies. Most FL papers do not specify the framework used (44 %), or they leverage frameworks widely used in ML applications, such as *PyTorch* (21 %) or *TensorFlow* (12 %). FL-ready frameworks with an integrated communication and aggregation layer are the minority. Only 24 % of the papers use an FL framework, such as *TensorFlow Federated* (4 %), *PySyft* (8 %), or *Flower* (12 %). We introduce a wide range of FL frameworks in Section 4.1.3. However, almost none achieved a wide range adaptation due to usage/license constraints or too short update cycles. For example, the FL framework from IBM has a community and enterprise edition, and only the latter is open for commercial use. Such constraints hinder adaptation. Additionally, their last version is almost 1.5 years old. Another reason for not using a specific FL framework is the ease of use. *Flower* only requires the installation of its *Python* package whereas *PySyft* requires Docker and a local database. All FL frameworks have some tutorials or installation guides, but more requirements increase complexity and potential sources of errors. Therefore, there seems to be a trend towards lean FL frameworks such as *TensorFlow Federated* and *Flower*, which use optimized serialization (Protobuf) and communication protocols (gRPC). Nevertheless 76 % of publications are most likely not built for a real-world FL deployment because they rely purely on ML frameworks, such as *TensorFlow* or *PyTorch*. The majority of publications using such ML frameworks focus on improving ML training in an FL system. Therefore, we infer that all publications with an unknown framework focusing also on ML performance use ML frameworks as well.

5.2 Federated Analytic Systems

The current driving motivation for FA is similar to FL systems' main motivation: Improving generated insights. FA systems run for one round and do not have an iterative optimization

Table 9. Categorization of FA papers into three groups. A reference can appear in multiple groups. Environment describes the number of hardware used for an experiment. Motivation refers to three categories introduced in Section 3.2 ((1)) = Improve insights or ML performance, (2) = Improve privacy or security, (3) = Improve hardware or network utilization). Client selection is either done manually, resource-aware, or loss-aware.

Motivation	Environment	Unknown	Single Node	Multiple Nodes
(1)	Manual Selection	[23, 31, 53, 56, 61, 69, 140, 141]	[14, 95, 139]	[85, 156]
(2)	Manual Selection	[22, 56]	[14, 31, 42, 69]	[85, 116, 119]
(3)	Manual Selection	[140]	[14, 42, 95, 129, 158]	[85, 156]

approach. Therefore, the focus is not on improving ML prediction performance, but rather on generating and aggregating single performance metrics. The captured scenarios for FL and FA are also similar and both focus on model-centric, horizontal FC in a cross-device environment. No surveyed FA paper works with either data-centric, vertical or cross-silo environments and the publication year for all publications are not older than 2021. This highlights the recent research interest in FA. Table 9 provides an overview of our literature research.

The type of hardware environment used for FA use cases is more evenly distributed when compared to the FL systems. Running the experiments on an unknown environment, a single node, or multiple nodes associate for 21 %, 32 %, and 47 %, respectively. There is no clear match between motivation and environment. For example, papers looking at hardware or network metrics use 63 % of the times a single node and only 25 % of those papers deploy their system on multiple nodes. This looks a bit different for papers capturing privacy or security motivations. About 45 % of them use an unknown environment, whereas 22 % and 33 % run their experiments on a single node or multiple nodes, respectively. The client selection is either manual, random, or unknown. 53 % of the papers select clients manually and 16 % use a random client selection. For both approaches the set of clients stays constant during the training. A manual client selection describes either an undefined client selection or all available clients are always selected. The latter could be a dataset which is inherently non-IID. For example, electricity measurements from multiple households which have a different dominant label per house (e.g., TV for household 1 and washing machine for household 2). However, 31 % of papers do not specify the type of client selection.

Figure 9 shows the distribution of frameworks and aggregation strategies used for FA systems. Not a single paper uses a framework specifically built for FA or FL and 74 % of papers do not mention which framework they use at all. The majority of papers use an ML framework, such as *TensorFlow*, *PyTorch* or *MATLAB*, to simulate an FA environment. It is feasiable to use an adjusted version of a framework dedicated for FL systems in an FA system. This requires only little to none changes, because in the simplest scenario an FL framework runs for only one round to mimic an FA system. This enables researchers to leverage the existing communication and serialization infrastructure. Therefore, it is not clear why no FA paper uses existing FL frameworks even when they try to improve hardware and network utilization. The field of frameworks used in FA systems is also less divers when compared to FL systems (three vs. five frameworks). However, the ratio of unknown frameworks for FA systems is 30 % percentage points higher, which introduces uncertainties when comparing FA with FL systems.

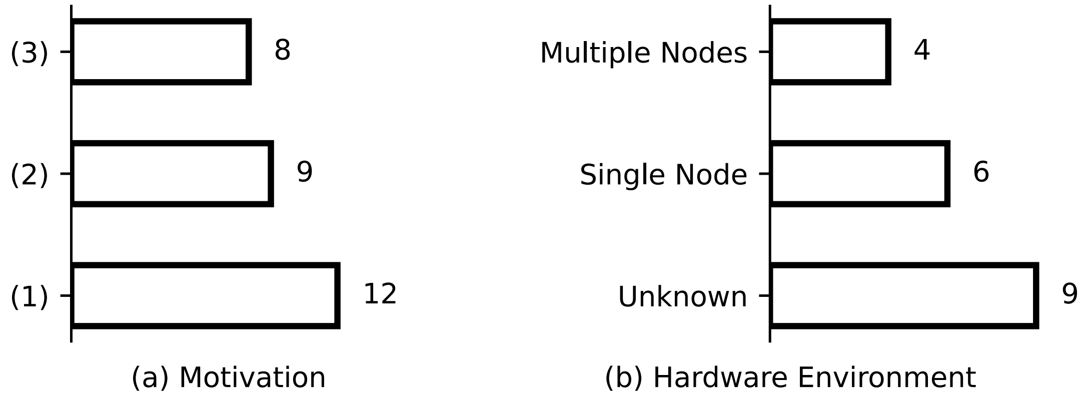


Fig. 9. Distributions of the problem definition (a) and hardware environment (b) of summarized FA systems. The numbers follow the structure introduced in Section 3.2 ((1) Improve insights, (2) Improve privacy or security, (3) Improve hardware or network utilization).

Figure 10 highlights the distribution of aggregation algorithms. Aggregation strategies are also mostly unknown (50 %) and the focus is currently on custom strategies (40 %). They often work with privacy-enhancing algorithms, such as homomorphic encryption, differential privacy or secure-multi party aggregation. The number of different aggregation strategies is much lower when compared to FL systems. A reason could be the reduced complexity of FA systems and the smaller impact of aggregation strategies on the final result, because an FA training runs for one round instead of multiple ones with an optimizer running on each client, which adds an additional layer of complexity.

However, half of the surveyed papers extend FA with privacy enhancing techniques. The privacy aspect is more present for FA systems compared to FL ones. FA training runs for only one round and often with deterministic models. The risk of de-annonymizing or reverse-engineering the raw data based on the clients' output is higher compared to statistical models with some degree of randomness. Therefore, it makes sense that about 35 % of surveyed papers combine FA systems

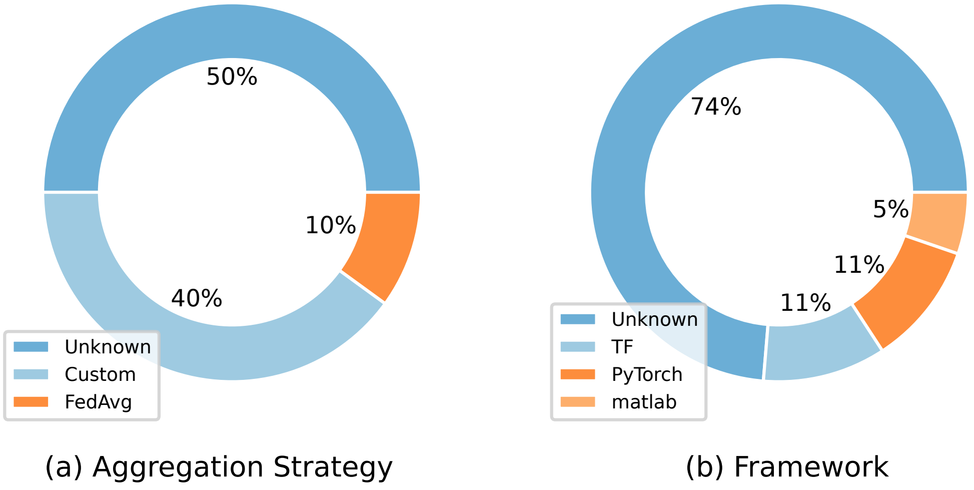


Fig. 10. Distributions of aggregation strategies (a) and frameworks (b) for FA papers. We surveyed in total 19 FA papers published after 2020.

with privacy-enhancing strategies. Improving generated insights is also easier compared to FL systems due to the reduction of system complexity. This allows researchers to focus on other topics.

6 DISCUSSION

6.1 Current Research Trends

FL got attention in the industry and academia starting in 2016 and FA emerged in 2020. However, their precursor such as Federated Databases have been studied long before that. Its main applications are in the ML domain. Research focuses currently on improving ML prediction performance by introducing new client selection (e.g., resource-aware or loss-aware client selection), aggregation algorithms or fine tuning hyperparameters of ML models. The primary challenge is a non-IID of labels on client-side leading to biased and unbalanced model updates. The network traffic and hardware consumption are for those experiments neglectable. Therefore, to test new ML-focused approaches, it is sufficient to deploy experiments on one machine which contains the server and multiple simulated clients. This strategy also reduces complexity and organizational overhead when compared to experiments running on distributed systems. The second most found motivation is improving hardware and network utilization of FL systems. Those papers solely focus on this issue. However, a few combine improving ML and hardware performance. Papers also considering the hardware or network utilization of an FL system run about half and half on either a single machine or on multiple ones. A few experiments investigate the impact of FL on those metrics theoretically by compiling optimization functions with constraints. Lastly, only a few papers extend FL systems with additional privacy-enhancing techniques, such as multi-party computation or DP. On the other side, papers describing FA systems mainly combine it with some kind of privacy-enhancing techniques. An explanation is the inherently deterministic nature of FA due to its lack of any optimizations running on client side. Therefore, it is easier to reverse-engineer raw data based on each clients individual update. To counter this, more works combine FA with techniques improving either input or output privacy.

Another focus in research is on model-centric, horizontal, and cross-device FC. This is the simplest scenario for FL and FA. Its goal is to improve ML performance by tuning hyper parameters, which is easier with a pre-defined homogeneous set of features and devices. The distribution of features per client can vary widely, but the feature set is the same for all of them. On the other side, data-centric and vertical FL and FA increases complexity. A data-centric approach aims at increasing ML performance by improving data quality. However, working with multiple distributed clients with a divers level of ownership makes it challenging to make changes on client-side. Also, vertical FL and FA work with multiple different feature sets, which can also vary per client.

About 99 % of surveyed papers incorporate a centralized architecture (see Figure 2). Only one looks at the advantages and challenges of hierarchical [18] and none at peer-to-peer systems.

6.2 Open Challenges

FL and FA often lack information about deployed systems, such as environment, framework, or aggregation strategy, which decreases reproducibility. Our proposed standardized framework to describe FC systems enables other researchers and stakeholders to identify similar systems for comparison.

FL and FA systems run mostly on one node or the environment is unknown due to the focus on ML accuracy improvements. This leads to a lack of understanding on how FC systems perform under real-world scenarios and how certain client selection and aggregation strategies affect network traffic and other hardware metrics, such as CPU utilization or energy consumption. Running an experiment on one device neglects potential latency or throughput bottlenecks of the network or the hardware itself. Therefore, ML-focused papers should be as reproducible as possible to enable other researchers to quantify their impact on the above mentioned metrics and to optimize hardware utilisation during the training process. Orchestration of multiple nodes in a real-world FC scenario is also not well researched.

Client selection algorithms primarily consider all available clients and the client set stays constant during training. Most dataset used in FC papers have a limited number of clients and reducing them could lead to an insufficient amount of (training) data. However, resource-aware or loss-aware client selection algorithms could improve ML performance and hardware utilization. Also, re-selecting a new subset of clients after an ML round increases complexity, but could potentially improve the entire systems' performance.

Not much work for FA exist. Strategies working for FL systems might achieve similar results in an FA system. FL-specific strategies, such as loss-aware client selection is not applicable in an FA context, because FA runs for one run and hence, cannot incorporate iterative optimizations. There is also a lack of using FC-specific frameworks in an FA system.

7 CONCLUSIONS

FL and FA enable the development of data-driven business models by leveraging data silos without interfering with data protection laws and by eliminating reservations from decision makers and stakeholders. Such models are paramount for improving business processes and to cope with the ever increasing complexity and velocity of changes in the global economy. Both approaches belong to FC. Over the last years researchers and practitioners expand FC systems with algorithms from other domains (e.g., encryption and compression) to minimize the effect of some of its disadvantages. However, those systems become more complex and there is currently a lack of clearly defined boundaries for such systems. Our work introduces a taxonomy capturing all moving parts in an FC systems. We differentiate between FL and FA. We summarize current research trends in FC systems and identify gaps. Additionally, we categorize existing frameworks and client selection algorithms. Currently, the majority of publications focus on FL systems with the goal to improve the accuracy of the trained machine learning models. The experiments mainly run on one device hosting the server and clients. There is a lack of research on the effect of FL Training on hardware utilization. A similar picture exists for FA systems. However, publications in this area tend to incorporate more privacy-enhancing techniques, such as differential privacy or secure-multi party computation. Our taxonomy servers as a blue print for further research on FC systems. Additionally, our comprehensive summary of existing FL frameworks highlights the focus on the combination of gRPC with Protobuf for communication and object serialization.

ACKNOWLEDGMENTS

This paper is funded by the Bavarian Ministry of Economic Affairs, Regional Development and Energy under the REMORA project with the identifier DIK0111/03.

REFERENCES

[1] Mustafa Abdul Salam, Sanaa Taha, and Mohamed Ramadan. 2021. COVID-19 detection using federated machine learning. *PLOS ONE* 16, 6 (06 2021), 1–25. https://doi.org/10.1371/journal.pone.0252573

[2] Sawsan Abdulrahman, Hanine Tout, Azzam Mourad, and Chamseddine Talhi. 2021. FedMCCS: Multicriteria Client Selection Model for Optimal IoT Federated Learning. *IEEE Internet of Things Journal* 8, 6 (2021), 4723–4735. https://doi.org/10.1109/JIOT.2020.3028742

[3] Sawsan Abdulrahman, Hanine Tout, Hakima Ould-Slimane, Azzam Mourad, Chamseddine Talhi, and Mohsen Guizani. 2021. A Survey on Federated Learning: The Journey From Centralized to Distributed On-Site Learning and Beyond. *IEEE Internet of Things Journal* 8, 7 (2021), 5476–5497. https://doi.org/10.1109/JIOT.2020.3030072

[4] Nadzurah Zainal Abidin and Amelia Ritahani Ismail. 2022. Federated Deep Learning for Automated Detection of Diabetic Retinopathy. In *2022 IEEE 8th International Conference on Computing, Engineering and Design (ICCED)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 1–5. https://doi.org/10.1109/ICCED56140.2022.10010636

[5] Haftay Gebreslasie Abreha, Mohammad Hayajneh, and Mohamed Adel Serhani. 2022. Federated Learning in Edge Computing: A Systematic Survey. *Sensors* 22, 2 (2022), 1–45. https://doi.org/10.3390/s22020450

[6] Blaise Agüera y Arcas et al. 2020. *Federated Analytics: Collaborative Data Science without Data Collection*. Google. https://ai.googleblog.com/2020/05/federated-analytics-collaborative-data.html Accessed December 5, 2023.

[7] Mohammed Aledhari, Rehma Razzak, Reza M. Parizi, and Fahad Saeed. 2020. Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Access* 8 (2020), 140699–140725. https://doi.org/10.1109/ACCESS.2020.3013541

[8] Moayad Aloqaily, Ismaeel Al Ridhawi, Fakhri Karray, and Mohsen Guizani. 2022. Towards Blockchain-based Hierarchical Federated Learning for Cyber-Physical Systems. In *2022 International Balkan Conference on Communications and Networking (BalkanCom)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 46–50. https://doi.org/10.1109/BalkanCom55633.2022.9900546

[9] Mohammad Mohammadi Amiri, Deniz Gunduz, Sanjeev R. Kulkarni, and H. Vincent Poor. 2020. Federated Learning With Quantized Global Model Updates. arXiv:2006.10672 [cs.IT]

[10] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. 2019. Federated Learning with Personalization Layers. arXiv:1912.00818 [cs.LG]

[11] Amna Arouj and Ahmed M. Abdelmoniem. 2022. Towards energy-aware federated learning on battery-powered clients. In *Proceedings of the 1st ACM Workshop on Data Privacy and Federated Learning Technologies for Mobile Edge Network* (Sydney, New South Wales, Australia) *(FedEdge '22)*. Association for Computing Machinery, New York, NY, USA, 7–12. https://doi.org/10.1145/3556557.3557952

[12] Brendan Avent, Javier Gonzalez, Tom Diethe, Andrei Paleyes, and Borja Balle. 2020. Automatic Discovery of Privacy-Utility Pareto Fronts. arXiv:1905.10862 [stat.ML]

[13] Leonardo Azevedo, Elton F. de S. Soares, Renan Souza, and Marcio Ferreira Moreno. 2020. Modern Federated Database Systems: An Overview. SciTePress, Online, 276–283. https://doi.org/10.5220/0009795402760283

[14] Eugene Bagdasaryan, Peter Kairouz, Stefan Mellem, Adrià Gascón, Kallista Bonawitz, Deborah Estrin, and Marco Gruteser. 2022. Towards Sparse Federated Analytics: Location Heatmaps under Distributed Differential Privacy with Secure Aggregation. arXiv:2111.02356 [cs.CR]

[15] David Basin, Søren Debois, and Thomas Hildebrandt. 2018. On Purpose and by Necessity: Compliance Under the GDPR. In *Financial Cryptography and Data Security: 22nd International Conference, FC 2018, Nieuwpoort, Curaçao, February 26 – March 2, 2018, Revised Selected Papers* (Nieuwpoort, Curaçao). Springer-Verlag, Berlin, Heidelberg, 20–37. https://doi.org/10.1007/978-3-662-58387-6_2

[16] Paolo Bellavista, Luca Foschini, and Alessio Mora. 2021. Decentralised Learning in Federated Deployment Environments: A System-Level Survey. *ACM Comput. Surv.* 54, 1, Article 15 (2 2021), 38 pages. https://doi.org/10.1145/3429252

[17] Daniel J. Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, Pedro P. B. de Gusmão, and Nicholas D. Lane. 2021. Flower: A Friendly Federated Learning Research Framework. arXiv:2007.14390 [cs.LG]

[18] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečný, Stefano Mazzocchi, H. Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. 2019. Towards Federated Learning at Scale: System Design. https://doi.org/10.48550/ARXIV.1902.01046

[19] Christopher Briggs, Zhong Fan, and Peter Andras. 2021. *A Review of Privacy-Preserving Federated Learning for the Internet-of-Things*. Springer International Publishing, Cham, 21–50. https://doi.org/10.1007/978-3-030-70604-3_2

[20] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. 2019. LEAF: A Benchmark for Federated Settings. arXiv:1812.01097 [cs.LG]

[21] Zheng Chai, Yujing Chen, Ali Anwar, Liang Zhao, Yue Cheng, and Huzefa Rangwala. 2021. FedAT: a high-performance and communication-efficient federated learning system with asynchronous tiers. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (St. Louis, Missouri) *(SC '21)*. Association for Computing Machinery, New York, NY, USA, Article 60, 16 pages. https://doi.org/10.1145/3458817.3476211

[22] Amit Chaulwar and Michael Huth. 2021. Secure Bayesian Federated Analytics for Privacy-Preserving Trend Detection. arXiv:2107.13640 [cs.CR]

[23] Dawei Chen, Dan Wang, Yifei Zhu, and Zhu Han. 2021. Digital Twin for Federated Analytics Using a Bayesian Approach. *IEEE Internet of Things Journal* 8, 22 (2021), 16301–16312. https://doi.org/10.1109/JIOT.2021.3098692

[24] Jiuxiang Chen, Tianlu Gao, Ruiqi Si, Yuxin Dai, Yuqi Jiang, and Jun Zhang. 2022. Residential Short Term Load Forecasting Based on Federated Learning. In *2022 IEEE 2nd International Conference on Digital Twins and Parallel Intelligence (DTPI)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 1–6. https://doi.org/

10.1109/DTPI55838.2022.9998969

[25] Shengbo Chen, Cong Shen, Lanxue Zhang, and Yuanmin Tang. 2021. Dynamic Aggregation for Heterogeneous Quantization in Federated Learning. *IEEE Transactions on Wireless Communications* 20, 10 (2021), 6804–6819. https://doi.org/10.1109/TWC.2021.3076613

[26] Wei Chen, Kartikeya Bhardwaj, and Radu Marculescu. 2020. FedMAX: Mitigating Activation Divergence for Accurate and Communication-Efficient Federated Learning. arXiv:2004.03657 [cs.LG]

[27] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. 2020. Client Selection in Federated Learning: Convergence Analysis and Power-of-Choice Selection Strategies. https://doi.org/10.48550/ARXIV.2010.01243

[28] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. 2020. Client Selection in Federated Learning: Convergence Analysis and Power-of-Choice Selection Strategies. arXiv:2010.01243 [cs.LG]

[29] Tejalal Choudhary, Vipul Mishra, Anurag Goswami, and Jagannathan Sarangapani. 2020. A comprehensive survey on model compression and acceleration. *Artif. Intell. Rev.* 53, 7 (10 2020), 5113–5155. https://doi.org/10.1007/s10462-020-09816-7

[30] Zhou Chuanxin, Sun Yi, and Wang Degang. 2020. Federated Learning with Gaussian Differential Privacy. In *Proceedings of the 2020 2nd International Conference on Robotics, Intelligent Control and Artificial Intelligence* (Shanghai, China) *(RICAI '20)*. Association for Computing Machinery, New York, NY, USA, 296–301. https://doi.org/10.1145/3438872.3439097

[31] Graham Cormode and Igor L. Markov. 2021. Bit-efficient Numerical Aggregation and Stronger Privacy for Trust in Federated Analytics. arXiv:2108.01521 [cs.CR]

[32] Shuang Dai, Fanlin Meng, Qian Wang, and Xizhong Chen. 2023. FederatedNILM: A Distributed and Privacy-Preserving Framework for Non-Intrusive Load Monitoring Based on Federated Deep Learning. In *2023 International Joint Conference on Neural Networks (IJCNN)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 01–08. https://doi.org/10.1109/IJCNN54540.2023.10191549

[33] Dimitrios Dimitriadis, Mirian Hipolito Garcia, Daniel Madrigal, Andre Manoel, and Robert Sim. 2022. FLUTE: A Scalable, Extensible Framework for High-Performance Federated Learning Simulations. https://www.microsoft.com/en-us/research/publication/flute-a-scalable-extensible-framework-for-high-performance-federated-learning-simulations/

[34] Ahmed Mukhtar Dirir, Khaled Salah, and Davor Svetinovic. 2021. *Towards Blockchain-Based Fair and Trustworthy Federated Learning Systems.* Springer International Publishing, Cham, 157–171. https://doi.org/10.1007/978-3-030-70604-3_7

[35] Zhaoyang Du, Celimuge Wu, Tsutomu Yoshinaga, Kok-Lim Alvin Yau, Yusheng Ji, and Jie Li. 2020. Federated Learning for Vehicular Internet of Things: Recent Advances and Open Issues. *IEEE Open Journal of the Computer Society* 1 (2020), 45–61. https://doi.org/10.1109/OJCS.2020.2992630

[36] Cynthia Dwork. 2007. An Ad Omnia Approach to Defining and Achieving Private Data Analysis. In *Proceedings of the 1st ACM SIGKDD International Conference on Privacy, Security, and Trust in KDD* (San Jose, CA, USA) *(PinKDD'07)*. Springer-Verlag, Berlin, Heidelberg, 1–13.

[37] Cynthia Dwork. 2008. Differential Privacy: A Survey of Results. In *Proceedings of the 5th International Conference on Theory and Applications of Models of Computation* (Xi'an, China) *(TAMC'08)*. Springer-Verlag, Berlin, Heidelberg, 1–19.

[38] Cynthia Dwork. 2011. A Firm Foundation for Private Data Analysis. *Commun. ACM* 54, 1 (1 2011), 86–95. https://doi.org/10.1145/1866739.1866758

[39] Sannara EK, Francois PORTET, Philippe LALANDA, and German VEGA. 2021. A Federated Learning Aggregation Algorithm for Pervasive Computing: Evaluation and Comparison. In *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 1–10. https://doi.org/10.1109/percom50583.2021.9439129

[40] Morgan Ekmefjord, Addi Ait-Mlouk, Sadi Alawadi, Mattias Åkesson, Desislava Stoyanova, Ola Spjuth, Salman Toor, and Andreas Hellander. 2021. Scalable federated machine learning with FEDn. *arXiv preprint arXiv:2103.00148* (2021), 1–14.

[41] Zakaria Abou El Houda, Diala Nabousli, and Georges Kaddoum. 2022. Cost-efficient Federated Reinforcement Learning- Based Network Routing for Wireless Networks. In *2022 IEEE Future Networks World Forum (FNWF)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 243–248. https://doi.org/10.1109/FNWF55208.2022.00050

[42] Ahmed Roushdy Elkordy, Yahya H. Ezzeldin, and Salman Avestimehr. 2022. Federated K-Private Set Intersection. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (Atlanta, GA, USA) *(CIKM '22)*. Association for Computing Machinery, New York, NY, USA, 436–445. https://doi.org/10.1145/3511808.3557321

[43] Ahmed Roushdy Elkordy, Yahya H. Ezzeldin, Shanshan Han, Shantanu Sharma, Chaoyang He, Sharad Mehrotra, and Salman Avestimehr. 2023. Federated Analytics: A survey. (2023), 1–33. https://doi.org/10.48550/ARXIV.2302.01326

[44] David Enthoven and Zaid Al-Ars. 2021. *An Overview of Federated Deep Learning Privacy Attacks and Defensive Strategies.* Springer International Publishing, Cham, 173–196. https://doi.org/10.1007/978-3-030-70604-3_8

[45] Tom Everitt and Marcus Hutter. 2018. *Universal Artificial Intelligence.* Springer International Publishing, Cham, 15–46. https://doi.org/10.1007/978-3-319-64816-3_2

[46] Allen-Jasmin Farcas, Xiaohan Chen, Zhangyang Wang, and Radu Marculescu. 2022. Model elasticity for hardware heterogeneity in federated learning systems. In *Proceedings of the 1st ACM Workshop on Data Privacy and Federated Learning Technologies for Mobile Edge Network* (Sydney, New South Wales, Australia) *(FedEdge '22).* Association for Computing Machinery, New York, NY, USA, 19–24. https://doi.org/10.1145/3556557.3557954

[47] Ali Farooq, Ali Feizollah, and Muhammad Habib ur Rehman. 2021. *Federated Learning Research: Trends and Bibliometric Analysis.* Springer International Publishing, Cham, 1–19. https://doi.org/10.1007/978-3-030-70604-3_1

[48] Ines Feki, Sourour Ammar, Yousri Kessentini, and Khan Muhammad. 2021. Federated learning for COVID-19 screening from Chest X-ray images. *Applied Soft Computing* 106 (2021), 107330. https://doi.org/10.1016/j.asoc.2021.107330

[49] Pietro Ferrara and Fausto Spoto. 2018. Static Analysis for GDPR Compliance. In *Italian Conference on Cybersecurity.* 1–10. https://api.semanticscholar.org/CorpusID:39745243

[50] Patrick Foley, Micah J Sheller, Brandon Edwards, Sarthak Pati, Walter Riviera, Mansi Sharma, Prakash Narayana Moorthy, Shih han Wang, Jason Martin, Parsa Mirhaji, Prashant Shah, and Spyridon Bakas. 2022. OpenFL: the open federated learning library. *Physics in Medicine and Biology* 67, 21 (10 2022), 214001. https://doi.org/10.1088/1361-6560/ac97d9

[51] Python Software Foundation. 2023. pickle — Python object serialization. https://docs.python.org/3/library/pickle.html Accessed March 20, 2023.

[52] Keith Frankish and William M. Ramsey. 2014. *The Cambridge handbook of artificial intelligence.* Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9781139046855

[53] David Froelicher, Juan R. Troncoso-Pastoriza, Jean Louis Raisaro, Michel A. Cuendet, Joao Sa Sousa, Hyunghoon Cho, Bonnie Berger, Jacques Fellay, and Jean-Pierre Hubaux. 2021. Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption. *Nature Communications* 12, 1 (11 10 2021), 5910. https://doi.org/10.1038/s41467-021-25972-y

[54] Mathieu N Galtier and Camille Marini. 2019. Substra: a framework for privacy-preserving, traceable and collaborative Machine Learning. https://doi.org/10.48550/ARXIV.1910.11567

[55] Melike Gecer and Benoit Garbinato. 2024. Federated Learning for Mobility Applications. *ACM Comput. Surv.* 56, 5, Article 133 (1 2024), 28 pages. https://doi.org/10.1145/3637868

[56] Martin Gjoreski, Matias Laporte, and Marc Langheinrich. 2022. Toward privacy-aware federated analytics of cohorts for smart mobility. *Frontiers in Computer Science* 4 (07 2022), 891206. https://doi.org/10.3389/fcomp.2022.891206

[57] WeBank AI Group. 2018. Federated Learning White Paper V1.0.

[58] Christoph Gröger. 2021. There is No AI without Data. *Commun. ACM* 64, 11 (10 2021), 98–108. https://doi.org/10.1145/3448247

[59] Venkat Gudivada, Amy Apon, and Junhua Ding. 2017. Data Quality Considerations for Big Data and Machine Learning: Going Beyond Data Cleaning and Transformations. *International Journal on Advances in Software* 10 (07 2017), 1–20.

[60] Yunzhe Guo, Dan Wang, Arun Vishwanath, Cheng Xu, and Qi Li. 2020. Towards Federated Learning for HVAC Analytics: A Measurement Study. In *Proceedings of the Eleventh ACM International Conference on Future Energy Systems* (Virtual Event, Australia) *(e-Energy '20).* Association for Computing Machinery, New York, NY, USA, 68–73. https://doi.org/10.1145/3396851.3397717

[61] Dave Hamersma, Kay Schreuder, Gijs Geleijnse, Erik Heeg, Matteo Cellamare, Marc Lobbes, Marc Mureau, Linetta Koppert, Helle Skjerven, Jan Nygård, Catharina Groothuis-Oudshoorn, and Sabine Siesling. 2023. Comparing quality of breast cancer care in the Netherlands and Norway by federated propensity score analytics. *Breast cancer research and treatment* 201 (06 2023). https://doi.org/10.1007/s10549-023-06986-0

[62] Chaoyang He, Songze Li, Jinhyun So, Xiao Zeng, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, Xinghua Zhu, Jianzong Wang, Li Shen, Peilin Zhao, Yan Kang, Yang Liu, Ramesh Raskar, Qiang Yang, Murali Annavaram, and Salman Avestimehr. 2020. FedML: A Research Library and Benchmark for Federated Machine Learning. https://doi.org/10.48550/ARXIV.2007.13518

[63] Yifan Hu, Yuhang Zhou, Jun Xiao, and Chao Wu. 2020. GFL: A Decentralized Federated Learning Framework Based On Blockchain. https://doi.org/10.48550/ARXIV.2010.10996

[64] Johannes Jakubik, Michael Vössing, Niklas Kühl, Jannis Walk, and Gerhard Satzger. 2022. Data-centric Artificial Intelligence. https://doi.org/10.48550/ARXIV.2212.11854

[65] Yuang Jiang, Shiqiang Wang, Víctor Valls, Bong Jun Ko, Wei-Han Lee, Kin K. Leung, and Leandros Tassiulas. 2023. Model Pruning Enables Efficient Federated Learning on Edge Devices. *IEEE Transactions on Neural Networks and Learning Systems* 34, 12 (2023), 10374–10386. https://doi.org/10.1109/TNNLS.2022.3166101

[66] Zhida Jiang, Yang Xu, Hongli Xu, Zhiyuan Wang, Chunming Qiao, and Yangming Zhao. 2022. FedMP: Federated Learning through Adaptive Model Pruning in Heterogeneous Edge Computing. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 767–779. https://doi.org/10.1109/ICDE53745.2022.00062

[67] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. 2021. Advances and Open Problems in Federated Learning. 14, 1–2 (6 2021), 1–210. https://doi.org/10.1561/2200000083

[68] Sai Praneeth Karimireddy, Narasimha Raghavan Veeraragavan, Severin Elvatun, and Jan F. Nygård. 2023. Federated Learning Showdown: The Comparative Analysis of Federated Learning Frameworks. In *2023 Eighth International Conference on Fog and Mobile Edge Computing (FMEC)*. 224–231. https://doi.org/10.1109/FMEC59375.2023.10305961

[69] Tatsuki Koga, Kamalika Chaudhuri, and David Page. 2023. Differentially Private Multi-Site Treatment Effect Estimation. arXiv:2310.06237 [cs.LG]

[70] Abdulkadir Korkmaz, Ahmad Alhonainy, and Praveen Rao. 2022. An Evaluation of Federated Learning Techniques for Secure and Privacy-Preserving Machine Learning on Medical Datasets. In *2022 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 1–7. https://doi.org/10.1109/AIPR57179.2022.10092212

[71] Pranvera Kortoçi, Yilei Liang, Pengyuan Zhou, Lik-Hang Lee, Abbas Mehrabi, Pan Hui, Sasu Tarkom, and Jon Crowcroft. 2022. Federated split GANs. In *Proceedings of the 1st ACM Workshop on Data Privacy and Federated Learning Technologies for Mobile Edge Network* (Sydney, New South Wales, Australia) *(FedEdge '22)*. Association for Computing Machinery, New York, NY, USA, 25–30. https://doi.org/10.1145/3556557.3557953

[72] Yogesh Kumar and Ruchi Singla. 2021. *Federated Learning Systems for Healthcare: Perspective and Recent Progress*. Springer International Publishing, Cham, 141–156. https://doi.org/10.1007/978-3-030-70604-3_6

[73] Weimin Lai and Qiao Yan. 2022. Federated Learning for Detecting COVID-19 in Chest CT Images: A Lightweight Federated Learning Approach. In *2022 4th International Conference on Frontiers Technology of Information and Computer (ICFTIC)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 146–149. https://doi.org/10.1109/ICFTIC57696.2022.10075165

[74] Sangyoon Lee and Dae-Hyun Choi. 2022. Federated Reinforcement Learning for Energy Management of Multiple Smart Homes With Distributed Energy Resources. *IEEE Transactions on Industrial Informatics* 18, 1 (2022), 488–497. https://doi.org/10.1109/TII.2020.3035451

[75] Li Li, Yuxi Fan, and Kuo-Yi Lin. 2020. A Survey on federated learning. In *2020 IEEE 16th International Conference on Control and Automation (ICCA)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 791–796. https://doi.org/10.1109/ICCA51439.2020.9264412

[76] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. 2021. A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection. *IEEE Transactions on Knowledge and Data Engineering* (2021), 1–1. https://doi.org/10.1109/tkde.2021.3124599

[77] Qi Li, Jin Ye, Wenzhan Song, and Zion Tse. 2021. Energy Disaggregation with Federated and Transfer Learning. In *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 698–703. https://doi.org/10.1109/WF-IoT51360.2021.9595167

[78] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. FedDANE: A Federated Newton-Type Method. arXiv:2001.01920 [cs.LG]

[79] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated Optimization in Heterogeneous Networks. arXiv:1812.06127 [cs.LG]

[80] Yiran Li, Hongwei Li, Guowen Xu, Tao Xiang, and Rongxing Lu. 2022. Practical Privacy-Preserving Federated Learning in Vehicular Fog Computing. *IEEE Transactions on Vehicular Technology* 71, 5 (2022), 4692–4705. https://doi.org/10.1109/TVT.2022.3150806

[81] Yi Liu, James J. Q. Yu, Jiawen Kang, Dusit Niyato, and Shuyu Zhang. 2020. Privacy-Preserving Traffic Flow Prediction: A Federated Learning Approach. *IEEE Internet of Things Journal* 7, 8 (2020), 7751–7763. https://doi.org/10.1109/JIOT.

2020.2991401

[82] Sin Kit Lo, Qinghua Lu, Chen Wang, Hye-Young Paik, and Liming Zhu. 2021. A Systematic Literature Review on Federated Machine Learning: From a Software Engineering Perspective. *ACM Comput. Surv.* 54, 5, Article 95 (5 2021), 39 pages. https://doi.org/10.1145/3450288

[83] Heiko Ludwig, Nathalie Baracaldo, Gegi Thomas, Yi Zhou, Ali Anwar, Shashank Rajamoni, Yuya Ong, Jayaram Radhakrishnan, Ashish Verma, Mathieu Sinn, Mark Purcell, Ambrish Rawat, Tran Minh, Naoise Holohan, Supriyo Chakraborty, Shalisha Whitherspoon, Dean Steuer, Laura Wynter, Hifaz Hassan, Sean Laguna, Mikhail Yurochkin, Mayank Agarwal, Ebube Chuba, and Annie Abay. 2020. IBM Federated Learning: an Enterprise Framework White Paper V0.1. arXiv:2007.10987 [cs.LG]

[84] Amirhossein Malekijoo, Mohammad Javad Fadaeieslam, Hanieh Malekijou, Morteza Homayounfar, Farshid Alizadeh-Shabdiz, and Reza Rawassizadeh. 2021. FEDZIP: A Compression Framework for Communication-Efficient Federated Learning. arXiv:2102.01593 [cs.LG]

[85] Elizabeth Margolin, Karan Newatia, Tao Luo, Edo Roth, and Andreas Haeberlen. 2023. Arboretum: A Planner for Large-Scale Federated Analytics with Differential Privacy. In *Proceedings of the 29th Symposium on Operating Systems Principles* (<conf-loc>, <city>Koblenz</city>, <country>Germany</country>, </conf-loc>) *(SOSP '23)*. Association for Computing Machinery, New York, NY, USA, 451–465. https://doi.org/10.1145/3600006.3624566

[86] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2016. Communication-Efficient Learning of Deep Networks from Decentralized Data. (2016), 1–11. https://doi.org/10.48550/ARXIV.1602.05629

[87] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2023. Communication-Efficient Learning of Deep Networks from Decentralized Data. arXiv:1602.05629 [cs.LG]

[88] Xiaopeng Mo and Jie Xu. 2021. Energy-Efficient Federated Edge Learning with Joint Communication and Computation Design. *Journal of Communications and Information Networks* 6, 2 (2021), 110–124. https://doi.org/10.23919/JCIN.2021.9475121

[89] Erum Mushtaq, Yavuz Faruk Bakman, Jie Ding, and Salman Avestimehr. 2023. Federated Alternate Training (Fat): Leveraging Unannotated Data Silos in Federated Segmentation for Medical Imaging. In *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 1–5. https://doi.org/10.1109/ISBI53787.2023.10230533

[90] Farid M. Naini, Jayakrishnan Unnikrishnan, Patrick Thiran, and Martin Vetterli. 2016. Where You Are Is Who You Are: User Identification by Matching Statistics. *IEEE Transactions on Information Forensics and Security* 11, 2 (2016), 358–372. https://doi.org/10.1109/TIFS.2015.2498131

[91] Arvind Narayanan and Vitaly Shmatikov. 2006. How To Break Anonymity of the Netflix Prize Dataset. https://doi.org/10.48550/ARXIV.CS/0610105

[92] Takayuki Nishio and Ryo Yonetani. 2019. Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 1–7. https://doi.org/10.1109/ICC.2019.8761315

[93] State of California Department of Justice. 2018. California Consumer Privacy Act of 2018 [1798.100 - 1798.199.100].

[94] Zirou Pan, Huan Geng, Linna Wei, and Wei Zhao. 2022. Adaptive Client Model Update with Reinforcement Learning in Synchronous Federated Learning. In *2022 32nd International Telecommunication Networks and Applications Conference (ITNAC)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 1–3. https://doi.org/10.1109/ITNAC55475.2022.9998360

[95] Shashi Raj Pandey, Minh N. H. Nguyen, Tri Nguyen Dang, Nguyen H. Tran, Kyi Thar, Zhu Han, and Choong Seon Hong. 2022. Edge-Assisted Democratized Learning Toward Federated Analytics. *IEEE Internet of Things Journal* 9, 1 (2022), 572–588. https://doi.org/10.1109/JIOT.2021.3085429

[96] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. 2018. Scalable Private Learning with PATE. arXiv:1802.08908 [stat.ML]

[97] Kilian Pfeiffer, Martin Rapp, Ramin Khalili, and Jörg Henkel. 2023. Federated Learning for Computationally Constrained Heterogeneous Devices: A Survey. *ACM Comput. Surv.* 55, 14s, Article 334 (7 2023), 27 pages. https://doi.org/10.1145/3596907

[98] Aman Priyanshu, Rakshit Naidu, Fatemehsadat Mireshghallah, and Mohammad Malekzadeh. 2021. Efficient Hyperparameter Optimization for Differentially Private Deep Learning. arXiv:2108.03888 [cs.LG]

[99] Cheng Qiu. 2023. A Network Traffic Classification Method Based on Federated Learning and Extreme Learning Machine. In *2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 284–289. https://doi.org/10.1109/ICCECT57938.2023.10140851

[100] Xinchi Qiu, Titouan Parcollet, Daniel J. Beutel, Taner Topal, Akhil Mathur, and Nicholas D. Lane. 2021. Can Federated Learning Save The Planet? arXiv:2010.06537 [cs.LG]

[101] Suraj Rajendran, Jihad S. Obeid, Hamidullah Binol, Ralph D‘Agostino, Kristie Foley, Wei Zhang, Philip Austin, Joey Brakefield, Metin N. Gurcan, and Umit Topaloglu. 2021. Cloud-Based Federated Learning Implementation Across Medical Centers. *JCO Clinical Cancer Informatics* 5 (2021), 1–11. https://doi.org/10.1200/CCI.20.00060 arXiv:https://doi.org/10.1200/CCI.20.00060 PMID: 33411624.

[102] G. Pradeep Reddy and Y. V. Pavan Kumar. 2023. A Beginner’s Guide to Federated Learning. In *2023 Intelligent Methods, Systems, and Applications (IMSA)*. 557–562. https://doi.org/10.1109/IMSA58542.2023.10217383

[103] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. 2020. FedPAQ: A Communication-Efficient Federated Learning Method with Periodic Averaging and Quantization. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 108)*, Silvia Chiappa and Roberto Calandra (Eds.). PMLR, 2021–2031. https://proceedings.mlr.press/v108/reisizadeh20a.html

[104] Fabio Ricciato, F Ricciato, A Bujnowska, A Wirthmann, M Hahn, and E Barredo-Capelot. 2020. A reflection on privacy and data confidentiality in Official Statistics. (02 2020), 1–8.

[105] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletarì, Holger R. Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N. Galtier, Bennett A. Landman, Klaus Maier-Hein, Sébastien Ourselin, Micah Sheller, Ronald M. Summers, Andrew Trask, Daguang Xu, Maximilian Baust, and M. Jorge Cardoso. 2020. The future of digital health with federated learning. *npj Digital Medicine* 3, 1 (14 9 2020), 119. https://doi.org/10.1038/s41746-020-00323-1

[106] Luc Rocher, Julien M. Hendrickx, and Yves-Alexandrericciato2020privacy de Montjoye. 2019. Estimating the success of re-identifications in incomplete datasets using generative models. *Nature Communications* (2019), 1–9. https://doi.org/10.1038/s41467-019-10933-3

[107] Arkadiusz Rudzki, Zuzsanna Paciorkiewicz, and Tomasz Augustyniak. 2021. An open-source federated learning framework.
https://fedbiomed.gitlabpages.inria.fr/.

[108] Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, and Jonathan Passerat-Palmbach. 2018. A generic framework for privacy preserving deep learning. arXiv:1811.04017 [cs.LG]

[109] Minseok Ryu, Youngdae Kim, Kibaek Kim, and Ravi K. Madduri. 2022. APPFL: Open-Source Software Framework for Privacy-Preserving Federated Learning. arXiv:2202.03672 [cs.LG]

[110] Yuris Mulya Saputra, Dinh Thai Hoang, Diep N. Nguyen, Eryk Dutkiewicz, Markus Dominik Mueck, and Srikathyayani Srikanteswara. 2019. Energy Demand Prediction with Federated Learning for Electric Vehicle Networks. In *2019 IEEE Global Communications Conference (GLOBECOM)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 1–6. https://doi.org/10.1109/GLOBECOM38437.2019.9013587

[111] Karthik V. Sarma, Stephanie A. Harmon, Thomas Sanford, Holger R. Roth, Ziyue Xu, Jesse Tetreault, Daguang Xu, Mona G. Flores, Alex G. Raman, Rushikesh Kulkarni, Bradford J. Wood, Peter L. Choyke, Alan Priester, Leonard S. Marks, Steven S Raman, Dieter R. Enzmann, Baris I Turkbey, W. Speier, and Corey W. Arnold. 2021. Federated learning improves site performance in multicenter deep learning without data sharing. *Journal of the American Medical Informatics Association : JAMIA* 28 (2021), 1259 – 1264. https://api.semanticscholar.org/CorpusID:231803993

[112] René Schwermer, Jonas Buchberger, Ruben Mayer, and Hans-Arno Jacobsen. 2022. Federated office plug-load identification for building management systems. In *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems* (Virtual Event) *(e-Energy ’22)*. Association for Computing Machinery, New York, NY, USA, 114–126. https://doi.org/10.1145/3538637.3538845

[113] René Schwermer, Jonas Buchberger, Ruben Mayer, and Hans-Arno Jacobsen. 2022. Federated Office Plug-Load Identification for Building Management Systems. In *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems* (Virtual Event) *(e-Energy ’22)*. Association for Computing Machinery, New York, NY, USA, 114–126. https://doi.org/10.1145/3538637.3538845

[114] René Schwermer, Ruben Mayer, and Hans-Arno Jacobsen. 2023. Energy vs Privacy: Estimating the Ecological Impact of Federated Learning. In *Proceedings of the 14th ACM International Conference on Future Energy Systems* (Orlando, FL, USA) *(e-Energy ’23)*. Association for Computing Machinery, New York, NY, USA, 347–352. https://doi.org/10.1145/3575813.3597344

[115] Amit P. Sheth and James A. Larson. 1990. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Comput. Surv.* 22, 3 (9 1990), 183–236. https://doi.org/10.1145/96602.96604

[116] Siping Shi, Chuang Hu, Dan Wang, Yifei Zhu, and Zhu Han. 2022. Federated Anomaly Analytics for Local Model Poisoning Attack. *IEEE Journal on Selected Areas in Communications* 40, 2 (2022), 596–610. https://doi.org/10.1109/JSAC.2021.3118347

[117] Nir Shlezinger, Mingzhe Chen, Yonina C. Eldar, H. Vincent Poor, and Shuguang Cui. 2020. Federated Learning with Quantization Constraints. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Institute of Electrical and Electronics Engineers (IEEE), Barcelona, Spain, 8851–8855. https://doi.org/10.1109/ICASSP40776.2020.9054168

[118] Nir Shlezinger, Mingzhe Chen, Yonina C. Eldar, H. Vincent Poor, and Shuguang Cui. 2021. UVeQFed: Universal Vector Quantization for Federated Learning. *IEEE Transactions on Signal Processing* 69 (2021), 500–514. https://doi.org/10.1109/TSP.2020.3046971

[119] James Short, Ken Miyachi, Christian D. Toouli, and Steve Todd. 2022. A field test of a federated learning/federated analytic blockchain network implementation in an HPC environment. In *Frontiers in Blockchain*, Vol. 5. Frontiers in Blockchain, 1–9. https://doi.org/10.3389/fbloc.2022.893747

[120] Personal Data Protection Commission Singapore. 2014. Personal Data Protection Act.

[121] Djura Smits, Bart Beusekom, Frank Martin, Lourens Veen, Gijs Geleijnse, and Arturo Moncada-Torres. 2022. An Improved Infrastructure for Privacy-Preserving Analysis of Patient Data. *Studies in health technology and informatics* 295 (6 2022), 144–147. https://doi.org/10.3233/SHTI220682

[122] Johannes Stutz. 2021. Limitations of Information Flows. https://blog.openmined.org/limitations-of-information-flows/ Accessed March 16, 2023.

[123] Afaf Taik and Soumaya Cherkaoui. 2020. Electrical Load Forecasting Using Edge Computing and Federated Learning. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 1–6. https://doi.org/10.1109/ICC40277.2020.9148937

[124] Hanlin Tang, Xiangru Lian, Chen Yu, Tong Zhang, and Ji Liu. 2020. DoubleSqueeze: Parallel Stochastic Gradient Descent with Double-Pass Error-Compensated Compression. arXiv:1905.05957 [cs.DC]

[125] Federated Learning Team. 2021. Federated Learning for Healthcare Using NVIDIA Clara White Paper.

[126] Jesse Tetreault, Rahul Choudhury, Brad Genereaux, Kristopher Kersten, and Jiahui Guan. 2020. Scalable and Modular AI Deployment Powered by NVIDIA Clara Deploy White Paper.

[127] Chandra Thapa, M. A. P. Chamikara, and Seyit A. Camtepe. 2021. *Advancements of Federated Learning Towards Privacy Preservation: From Federated Learning to Split Learning*. Springer International Publishing, Cham, 79–109. https://doi.org/10.1007/978-3-030-70604-3_4

[128] Adam Thor Thorgeirsson, Stefan Scheubner, Sebastian Fünfgeld, and Frank Gauterin. 2021. Probabilistic Prediction of Energy Demand and Driving Range for Electric Vehicles With Federated Learning. *IEEE Open Journal of Vehicular Technology* 2 (2021), 151–161. https://doi.org/10.1109/OJVT.2021.3065529

[129] László Toka, Márk Konrad, István Pelle, Balázs Sonkoly, Marcell Szabó, Bhavishya Sharma, Shashwat Kumar, Madhuri Annavazzala, Sree Teja Deekshitula, and A. Antony Franklin. 2023. 5G on the Roads: Latency-Optimized Federated Analytics in the Vehicular Edge. *IEEE Access* 11 (2023), 81737–81752. https://doi.org/10.1109/ACCESS.2023.3301330

[130] Andrew Trask, Emma Bluemke, Ben Garfinkel, Claudia Ghezzou Cuervas-Mons, and Allan Dafoe. 2020. Beyond Privacy Trade-offs with Structured Transparency. https://doi.org/10.48550/ARXIV.2012.08347

[131] Ye Lin Tun, Kyi Thar, Chu Myaet Thwal, and Choong Seon Hong. 2021. Federated Learning based Energy Demand Prediction with Clustered Aggregation. In *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 164–167. https://doi.org/10.1109/BigComp51126.2021.00039

[132] European Union. 2016. 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).

[133] Joel Walmsley. 2012. *Classical Cognitive Science and Good Old Fashioned AI*. Palgrave Macmillan UK, London, 30–64. https://doi.org/10.1057/9781137283429_3

[134] Dan Wang, Siping Shi, Yifei Zhu, and Zhu Han. 2022. Federated Analytics: Opportunities and Challenges. *IEEE Network* 36, 1 (2022), 151–158. https://doi.org/10.1109/MNET.101.2100328

[135] Haijin Wang, Caomingzhe Si, and Junhua Zhao. 2021. Fed-NILM: A Federated Learning-based Non-Intrusive Load Monitoring Method for Privacy-Protection. *ArXiv* abs/2105.11085 (2021), 51–60. https://api.semanticscholar.org/CorpusID:235166219

[136] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. 2020. Federated Learning with Matched Averaging. arXiv:2002.06440 [cs.LG]

[137] Sihua Wang, Mingzhe Chen, Walid Saad, and Changchuan Yin. 2020. Federated Learning for Energy-Efficient Task Computing in Wireless Networks. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 1–6. https://doi.org/10.1109/ICC40277.2020.9148625

[138] Ting Wang and Ling Liu. 2011. Output privacy in data mining. *ACM Trans. Database Syst.* 36, 1, Article 1 (3 2011), 34 pages. https://doi.org/10.1145/1929934.1929935

[139] Zibo Wang, Yifei Zhu, Dan Wang, and Zhu Han. 2022. FedFPM: A Unified Federated Analytics Framework for Collaborative Frequent Pattern Mining. In *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 61–70. https://doi.org/10.1109/INFOCOM48880.2022.9796719

[140] Zibo Wang, Yifei Zhu, Dan Wang, and Zhu Han. 2023. Federated Analytics Informed Distributed Industrial IoT Learning With Non-IID Data. *IEEE Transactions on Network Science and Engineering* 10, 5 (2023), 2924–2939. https://doi.org/10.1109/TNSE.2022.3187992

[141] Zibo Wang, Yifei Zhu, Dan Wang, and Zhu Han. 2023. Secure Trajectory Publication in Untrusted Environments: A Federated Analytics Approach. *IEEE Transactions on Mobile Computing* 22, 11 (2023), 6742–6754. https://doi.org/10.1109/TMC.2022.3198550

[142] Kang Wei, Jun Li, Chuan Ma, Ming Ding, and H. Vincent Poor. 2021. *Differentially Private Federated Learning: Algorithm, Analysis and Optimization.* Springer International Publishing, Cham, 51–78. https://doi.org/10.1007/978-3-030-70604-3_3

[143] Qi Xia, Winson Ye, Zeyi Tao, Jindi Wu, and Qun Li. 2021. A survey of federated learning for edge computing: Research problems and solutions. *High-Confidence Computing* 1, 1 (2021), 100008. https://doi.org/10.1016/j.hcc.2021.100008

[144] Yuexiang Xie, Zhen Wang, Dawei Gao, Daoyuan Chen, Liuyi Yao, Weirui Kuang, Yaliang Li, Bolin Ding, and Jingren Zhou. 2022. FederatedScope: A Flexible Federated Learning Platform for Heterogeneity. arXiv:2204.05011 [cs.LG]

[145] Jie Xu, Benjamin S. Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. 2020. *Federated Learning for Healthcare Informatics.* Springer. https://doi.org/10.1007/s41666-020-00082-4

[146] Jie Xu and Heqiang Wang. 2021. Client Selection and Bandwidth Allocation in Wireless Federated Learning Networks: A Long-Term Perspective. *IEEE Transactions on Wireless Communications* 20, 2 (2021), 1188–1200. https://doi.org/10.1109/TWC.2020.3031503

[147] Dong Yang, Ziyue Xu, Wenqi Li, Andriy Myronenko, Holger R. Roth, Stephanie Harmon, Sheng Xu, Baris Turkbey, Evrim Turkbey, Xiaosong Wang, Wentao Zhu, Gianpaolo Carrafiello, Francesca Patella, Maurizio Cariati, Hirofumi Obinata, Hitoshi Mori, Kaku Tamura, Peng An, Bradford J. Wood, and Daguang Xu. 2021. Federated semi-supervised learning for COVID region segmentation in chest CT using multi-national data from China, Italy, Japan. *Medical Image Analysis* 70 (2021), 101992. https://doi.org/10.1016/j.media.2021.101992

[148] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol.* 10, 2, Article 12 (1 2019), 19 pages. https://doi.org/10.1145/3298981

[149] Wenqi Yang, Yang Zhang, Wei Yang Bryan Lim, Zehui Xiong, Yutao Jiao, and Jiangming Jin. 2020. Privacy is not Free: Energy-Aware Federated Learning for Mobile and Edge Intelligence. In *2020 International Conference on Wireless Communications and Signal Processing (WCSP).* Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 233–238. https://doi.org/10.1109/WCSP49889.2020.9299703

[150] Dongdong Ye, Rong Yu, Miao Pan, and Zhu Han. 2020. Federated Learning in Vehicular Edge Computing: A Selective Model Aggregation Approach. *IEEE Access* 8 (2020), 23920–23935. https://doi.org/10.1109/ACCESS.2020.2968399

[151] Xuefei Yin, Yanming Zhu, and Jiankun Hu. 2021. A Comprehensive Survey of Privacy-Preserving Federated Learning: A Taxonomy, Review, and Future Directions. *ACM Comput. Surv.* 54, 6, Article 131 (7 2021), 36 pages. https://doi.org/10.1145/3460427

[152] Sixing Yu, Phuong Nguyen, Ali Anwar, and Ali Jannesari. 2023. Heterogeneous Federated Learning using Dynamic Model Pruning and Adaptive Gradient. arXiv:2106.06921 [cs.LG]

[153] Hui Zang and Jean Bolot. 2011. Anonymization of Location Data Does Not Work: A Large-Scale Measurement Study. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking* (Las Vegas, Nevada, USA) *(MobiCom '11).* Association for Computing Machinery, New York, NY, USA, 145–156. https://doi.org/10.1145/2030613.2030630

[154] Qunsong Zeng, Yuqing Du, Kaibin Huang, and Kin K. Leung. 2020. Energy-Efficient Radio Resource Allocation for Federated Edge Learning. In *2020 IEEE International Conference on Communications Workshops (ICC Workshops).* Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 1–6. https://doi.org/10.1109/ICCWorkshops49005.2020.9145118

[155] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. 2021. A survey on federated learning. *Knowledge-Based Systems* 216 (2021), 106775. https://doi.org/10.1016/j.knosys.2021.106775

[156] Li Zhang, Junji Qiu, Shangguang Wang, and Mengwei Xu. 2022. Device-centric Federated Analytics At Ease. arXiv:2206.11491 [cs.DC]

[157] Yu Zhang, Guoming Tang, Qianyi Huang, Yi Wang, Kui Wu, Keping Yu, and Xun Shao. 2023. FedNILM: Applying Federated Learning to NILM Applications at the Edge. *IEEE Transactions on Green Communications and Networking* 7, 2 (2023), 857–868. https://doi.org/10.1109/TGCN.2022.3167392

[158] Bowen Zhao, Xiaoguo Li, Ximeng Liu, Qingqi Pei, Yingjiu Li, and Robert H. Deng. 2023. CrowdFA: A Privacy-Preserving Mobile Crowdsensing Paradigm via Federated Analytics. *IEEE Transactions on Information Forensics and Security* 18 (2023), 5416–5430. https://doi.org/10.1109/TIFS.2023.3308714

[159] Wanru Zhao, Xinchi Qiu, Javier Fernandez-Marques, Pedro P. B. de Gusmão, and Nicholas D. Lane. 2022. Protea: client profiling within federated systems using flower. In *Proceedings of the 1st ACM Workshop on Data Privacy and Federated Learning Technologies for Mobile Edge Network* (Sydney, New South Wales, Australia) *(FedEdge '22).* Association for

Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3556557.3557950

[160] Jiehan Zhou, Shouhua Zhang, Qinghua Lu, Wenbin Dai, Min Chen, Xin Liu, Susanna Pirttikangas, Yang Shi, Weishan Zhang, and Enrique Herrera-Viedma. 2021. A Survey on Federated Learning and its Applications for Accelerating Industrial Internet of Things. arXiv:2104.10501 [cs.DC]

[161] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. 2021. Federated Learning on Non-IID Data: A Survey. arXiv:2106.06843 [cs.LG]

[162] Xinghua Zhu, Jianzong Wang, Zhenhou Hong, Tian Xia, and Jing Xiao. 2019. Federated Learning of Unsegmented Chinese Text Recognition Model. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 1341–1345. https://doi.org/10.1109/ICTAI.2019.00186

[163] Derun Zou, Xusheng Liu, Lintan Sun, Jianhui Duan, Ruichen Li, Yeting Xu, Wenzhong Li, and Sanglu Lu. 2022. FedMC: Federated Reinforcement Learning on the Edge with Meta-Critic Networks. In *2022 IEEE International Performance, Computing, and Communications Conference (IPCCC)*. Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 344–351. https://doi.org/10.1109/IPCCC55026.2022.9894336

[164] Huan Zou, Yuchao Zhang, Xirong Que, Yilei Liang, and Jon Crowcroft. 2022. Efficient federated learning under non-IID conditions with attackers. In *Proceedings of the 1st ACM Workshop on Data Privacy and Federated Learning Technologies for Mobile Edge Network* (Sydney, New South Wales, Australia) *(FedEdge '22)*. Association for Computing Machinery, New York, NY, USA, 13–18. https://doi.org/10.1145/3556557.3557951