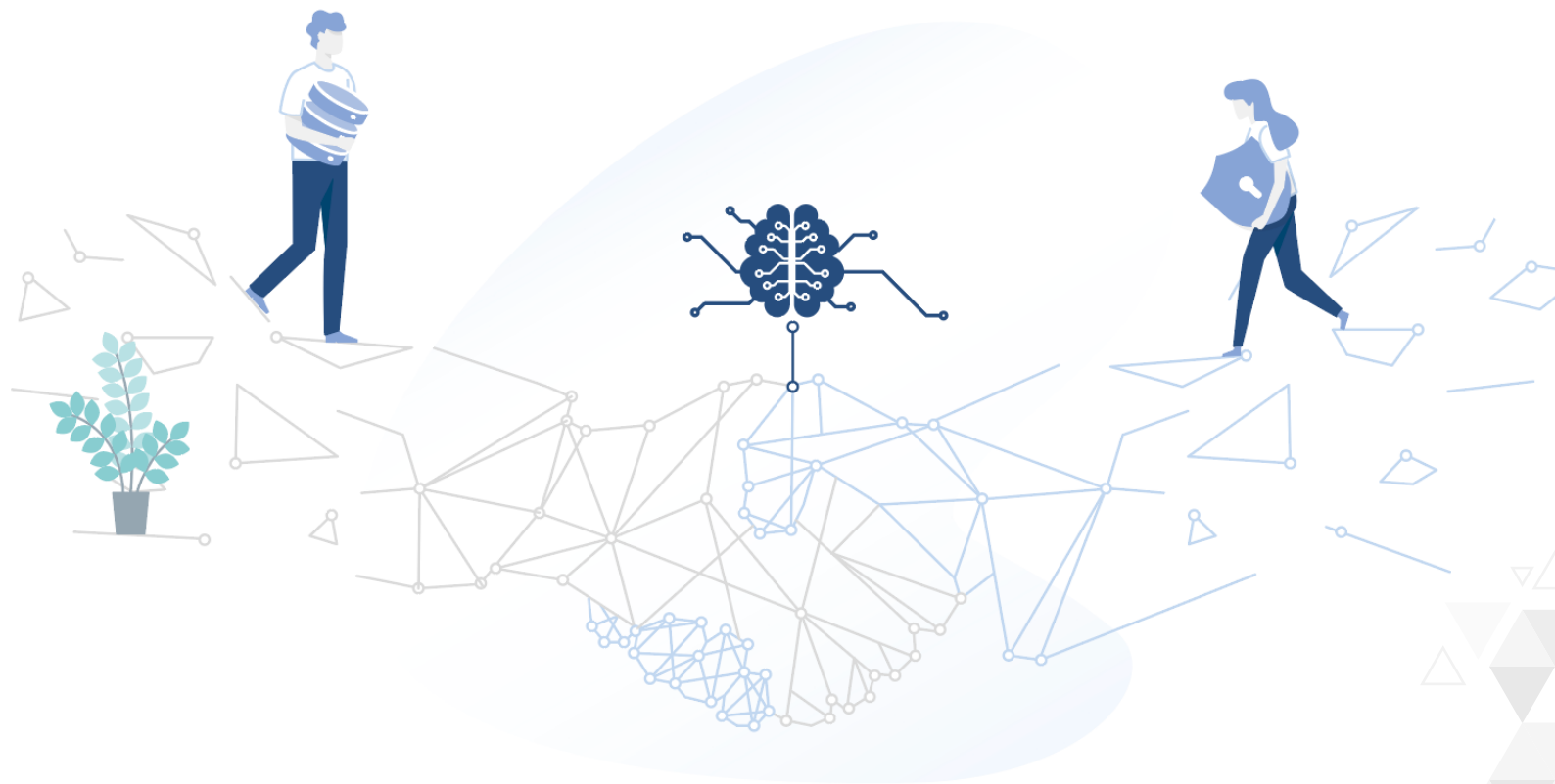


THE EXPANDABLE AI NETWORK

Whitepaper

GDPR-Compliance & Data Privacy Layer
For Machine Learning Applications

Lea Dänschel
Michael Huth
Leif-Nissen Lundbæk



Content

2	1. INTRODUCTION
4	2. BACKGROUND
	2.1 Centralized, Distributed, and Decentralized Machine Learning
	2.2 Adoption Hurdles for Machine Learning
	2.3 Data Privacy and Anonymization
7	3. FEDERATED MACHINE LEARNING
	3.1 A Gentle Introduction: Federated Averaging
	3.2 Types of Learning
	3.3 Aggregation Mechanisms
	3.4 Security
	3.5 Data Privacy: Technical Perspective
	3.6 Experimental Results: Benefits of FedML
16	4. OUR FEDML PLATFORM
17	5. COMPLIANCE
	5.1 Data anonymization and GDPR
	5.2 Principle of Least Privilege: Reducing the “Means Used”
	5.3 Influencing Future AI Regulation
21	6. OUR FOCUS ON GROUPS AND USE CASES
	6.1 Benefits We Offer
	6.2 Values We Promote
23	7. SAMPLE USE CASES
	7.1 ANDY: an AI app by XAIN
	7.2 AI-based Software-as-a-Service
	7.3 Anomaly Detection in IoT and Machine Data
	7.4 Federated Autonomous Driving
	7.5 Mobile Applications
26	8. CONCLUSIONS
27	References



1. Introduction

Artificial Intelligence (AI) refers to the ability of machines to display intelligent behavior. Since the 1950s, Artificial Intelligence is also a major research area across several disciplines, including Computer Science, Mathematics, and Statistics.

In the past 60 years, Artificial Intelligence has seen several “waves of optimism”, interleaved with periods of disillusionments – the so called “AI Winters”. In recent years, another wave of optimism began and is driven in no small part by the availability of cheaper and more powerful hardware and the open access of research findings, for example through open-source software packages for free experimentation and off-the-shelf hardware designed for solving AI problems.

These successes, particularly in the area of Deep Learning, have been widely publicized – for example, the Alpha Go AI that beat a world-class Go player. These advances have also moved AI up the strategic agenda of both politicians and decision makers in industry. AI is seen as a key technology in making territories and companies alike competitive in the future digital marketplaces.

Machine Learning It turns out that almost all of these recent advances stem from an area of AI called machine learning. This is a field that applies methods from statistics to solve two related problems:

- ▶ **Model Training:** training data is prepared and subjected to an algorithm that learns a mathematical model. The model represents patterns identified in the training data.
- ▶ **Model Inference:** the learned model is subjected to new data input in order to make inferences based on such learned patterns for decision support.

In mathematical terms, machine learning wants to learn an approximation of a given probability distribution and the training data is assumed to have been sampled according to that distribution. Inference then uses the learned, approximate distribution.

For example, a Deep Learning algorithm may be used on medical images to learn a model that can then predict from 3-D brain scans whether a patient has a particular brain disease. It is important to keep in mind that such predictions are merely statistical assertions, and not hard and qualitative facts. This means that machine learning needs to support decision making in a manner that is human-centric and meets expectations of ethics and legal requirements. And this is especially true when decisions are being made in an automated fashion, for example in an autonomously driving car.

FROM RESEARCH TO IMPACT

Machine learning has great potential to make decision-support processes smarter, cheaper, more automated, and self-improving. This makes the adoption of machine learning an important topic in the strategic planning of government agencies, industrial companies, and third-sector organizations.

It is also recognized that widespread commercial adoption of machine learning will require approaches that offer sufficient guarantees about the security and privacy of the process of learning models, especially when companies wish to learn collaboratively from datasets that they locally maintain and control. For example, a recent report [Res19] asked German enterprises to give their biggest reason for not considering the cloud for machine learning applications. Over a third of these companies cited data-protection issues as their biggest concern. In that same study, 20 percent of surveyed CEOs and Directors stated reservations about deploying AI solutions for fear that they could cause compliance and data-protection issues.

Federated Machine Learning has been proposed as a viable solution for this. In simple terms, this approach allows parties to learn an updated model on their local datasets, communicate such an updated model to a third party who then aggregates these locally updated models into a global model. That global model is then the basis for subsequent local learning. In particular, all data stays

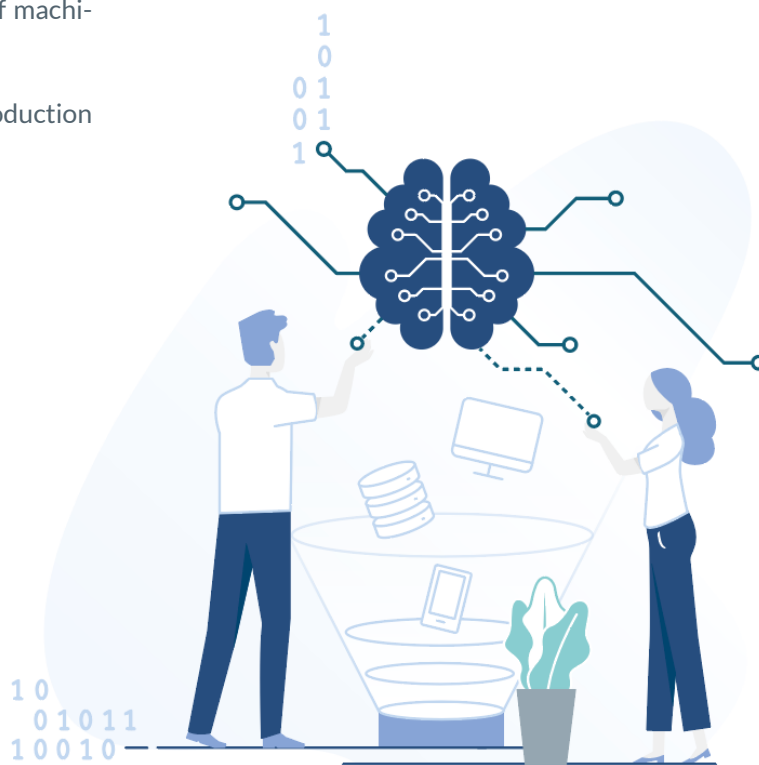
on-premise, only updated models get communicated in the network, and there is no need to perform costly data anonymization on local datasets. Therefore, this approach can accommodate data protection directives such as GDPR with greater ease, scalability, and cost effectiveness.

In that context, we work at building the eXpandable AI Network (XAIN) whose express intent is to achieve the following goals:

- G1** To develop a framework for Federated Machine Learning that is customizable to the particular needs of use cases.
- G2** To design this framework so that its security and privacy guarantees enable many important use cases that standard machine learning could not address in a compliant manner.
- G3** To enhance this framework with techniques from automated machine learning to lower the adoption hurdles of machine learning in companies.
- G4** To deploy this framework in production as an Infrastructure as a Service.

OUTLINE OF PAPER

In Section 2, we review salient machine-learning architectures, identify adoption hurdles for machine learning, and assess the feasibility of data anonymization techniques in practice. Federation Machine Learning is introduced and discussed in Section 3. The technical vision of our FedML platform is the subject of Section 4. A discussion and assessment of legal issues around the compliance of Federated Machine Learning use cases is the subject of Section 5. A review of users and customers that the XAIN Platform targets is provided in Section 6 and pertinent use cases are studied in Section 7. We conclude in Section 8.



2. Background

2.1 CENTRALIZED, DISTRIBUTED, AND DECENTRALIZED MACHINE LEARNING

Machine learning, as explained above, is executing an algorithm over a set of training data on a dedicated computing machine. We may refer to this as *centralized* machine learning. For medium to large scale machine learning problems, it is pretty common to apply *distributed* machine learning instead.

A popular approach for that is to partition the set of training data \mathcal{D} into different buckets \mathcal{D}_i , each hosted on a respective machine M_i , to run the same learning algorithm on each machine and bucket, and to then have a means of combining the learned local models into a global one. For that, let us assume that there is no adversary that may manipulate local data, local learning or message content and delivery. Then the same model can be learned as if centralized machine learning were to be used. Therefore, such distribution does not “break things”, but why would we choose it?

The advantage of distributed machine learning is that it can complete a large learning task at greater speed and scale by distributing the learning task to multiple machines. In order to accomplish this, trade offs between speed gains and model accuracy have to be considered (see for example [GZW17]). Moreover, the tuning of hyperparameters such as the epoch size [GZW17] may be required, communication protocols may need to be optimized to reduce communication complexity and payload (see for example [?]), and the machine-learning algorithms should be robust against communication failure and downtime of machines (see for example [YTR+19]).

Distributed machine learning is also useful when the training set \mathcal{D} is already physically split up, for example when each \mathcal{D}_i represents data from an autonomous vehicle i across an entire fleet of vehicles. In such use cases, economic and other considerations may prevent a combination of all such data into a sole dataset on which centralized machine learning could then be performed.

The combination of results from local learning described above is often done by a third party called a “Parameter Server”. Alternatively, the combination of local models may use communication protocols that do not rely on a third party and we then speak of *decentralized* machine learning as a special form of distributed machine learning [yA18].

The area of Distributed Systems has a lot to offer in answering open research questions in distributed machine learning. Let us illustrate this with two such questions: To what degree can model combination happen asynchronously in order to improve parallel processing, without compromising the mathematical behavior of the combined learning? How can we make mechanisms for model combination resilient to adversarial manipulation? Answering such questions needs to draw from non-trivial insights of Distributed Systems and we will revisit the latter question further below.

2.2 ADOPTION HURDLES FOR MACHINE LEARNING

Although machine learning and AI are high on the strategic agenda, there are significant hurdles faced in the adoption of machine-learning technology in enterprises, and similarly in noncommercial institutions. Let us list and then discuss important hurdles for such adoption:

- AH1** Machine learning technology is the output of research and its transfer to commercial usage is non-trivial and requires high levels of expertise [ea18] [Res19].
- AH2** Many companies (for example SMEs) or use cases (for example smart-phone usage data) deal with “small data” on which machine learning may be less or not at all effective [ea18] [Res19].
- AH3** The ideal training data of a company often resides in *data silos* separated across physical boundaries (for example different IT systems) or political/legal boundaries (for example across

departments or subsidiaries) which creates obstacles to combining such data [oCCA19].

AH4 Successful training of a model involves a complex workflow, including the pre-processing of training data and tuning of so called “hyperparameters”, for example the topological structure of a Deep Learning network. These are typically manual and high-skill tasks [oCCA19].

Hurdle AH1 is a reflection of the fact that Machine Learning is research-driven. There is a need for better knowledge and technology transfer from research-based machine learning tools to their usability and integration in commercial decision-support workflows.

The hurdle AH2 suggests that there are incentives in designing and using *collaborative* forms of machine learning. Such collaboration may take on many forms but will certainly have to accommodate constraints that stem from hurdle AH3. In fact, Google developed *Federated Machine Learning* as a form of collaborative, distributed machine learning for predicting which word a user wants to type on her keyboard as a function of already typed in keys [MMRyA16].

Regulatory constraints around privacy prevented the combination of all user keyboard data into a global set of training data. Rather, the training was done on local training data for each user/device and only model information was then combined. We will return to this approach in Section 3.

One reaction to the problem of data silos stated in hurdle AH3 is to work on resolving the political, legal or technical issues that are at the heart of this. But we believe that such solutions are unlikely to come forward. Obtaining such solutions is either too disruptive and costly, may take too much time to complete or may not be consistent with existing laws and regulations as in the Google use case just described.

Therefore, we identify the development and deployment of technology that mitigates against the factors of hurdle AH3 as a pivotal contribution to the wider and successful adoption of Machine Learning in practice. Moreover, such technology will positively impact the issues in hurdle AH2 as such mitigating effects will also apply to datasets of smaller size.

Hurdle AH4 also is quite significant. The workflows of machine learning are fairly complex, and small mistakes at steps of such workflows can then corrupt the entire learning process and make the learned models unusable. These issues are amplified by a substantial shortage in available AI engineers/developers and AI knowledge in the global workforce, which severely limits the successful pursuit of AI projects in companies.

Therefore, any tools and technologies that can realize the execution of machine-learning workflows more easily, without compromising the quality of the obtained outputs, will have strategic importance. We argue that *Automated Machine Learning* (AutoML) has great potential to that end.

AutoML and ENAS

The workflow of machine learning consists of many tasks, and it is not realistic to replace all of them with automated tools. Fortunately, many of these tasks are similar to those found in software engineering projects (for example the entire phase of use case conception and formulation). Therefore, companies may have in-house expertise for completing such tasks – which also do not seem automatable with current AI techniques.

In contrast, we think that the tuning of hyperparameters is a very promising target for AutoML in order to make it easier for companies to use and integrate machine learning into their systems. In machine learning, a parameter is a variable (for example the mean or the variance of a probability distribution) whose value will be learned through training. A *hyperparameter*, is a parameter whose value is set to a constant before training takes place. This

is typically done manually, often in an ad-hoc manner and based on considerable experience.

Replacing this manual process of hyperparameter selection with an automated one, for example one that uses machine learning to learn values for hyperparameters, is therefore compelling.¹ This is particularly the case in Deep Learning, where determining an optimal network topology is hard to do even for experts. Effective Neural Architecture Search (ENAS) is one such approach that shows great promise. Therefore, we want to develop solutions that are supported by AutoML tools, in particular ENAS, in order to help with the adoption of AI in commercial settings.

2.3 DATA PRIVACY AND ANONYMIZATION

Let us discuss one possible solution to the problems around hurdle AH3. Suppose that we have two datasets \mathcal{D}_1 and \mathcal{D}_2 , each collected by a different entity, and each containing personally identifiable information (PII) collected with the informed consent of those persons for a specific usage. That usage consent for \mathcal{D}_1 , say, is unlikely to include merging that dataset with \mathcal{D}_2 ; for example dataset \mathcal{D}_2 may not have existed at the time when users gave consent for usage of \mathcal{D}_1 . So we assume that we are not permitted to combine \mathcal{D}_1 and \mathcal{D}_2 .

One way around this issue is to anonymize each dataset to obtain anonymized versions \mathcal{D}_1^a and \mathcal{D}_2^a , respectively. We use the term “data anonymization” to stand for the process of cleaning datasets so that their cleaned versions can no longer create any links to any PII. The interpretation of such *linkability* is guided by regulations and their legal interpretation, for example for the EU GDPR and the legal interpretation thereof.

Data anonymization does not refer to other anonymization techniques such as those provided by cryptography and discussed further below. We refer to [2914] for a detailed assessment of standard anonymization techniques from a legal perspective.

Once anonymized versions of datasets have been produced, the combined dataset $\mathcal{D}_1^a \cup \mathcal{D}_2^a$ can then be subjected to machine learning algorithms while still being compliant with privacy regulation such as GDPR. However, there are several issues with this approach:

- ▶ data anonymization is a labor intense, manual tasks and so is costly and non-scalable,
- ▶ data anonymization may still be subject to re-identification attacks, particularly when an attacker has meta-data available,
- ▶ data anonymization needs to pass legal scrutiny which sets a high standard for making links to PII irrevocable,
- ▶ data anonymization may require different tools depending on the type of data, for example when dealing with micro-data as opposed to larger datasets.

Given this, we believe that such data anonymization techniques, while having their place in the toolbox of machine learning, are not an effective means for helping with the widespread adoption of machine learning in commercial decision support. Instead, we will have to seek alternative solutions and we will advocate Federated Machine Learning as such a solution next.

3. Federated Machine Learning

Federated Machine Learning (FedML) is a form of collaborative and distributed machine learning. FedML seems imminently suitable in overcoming regulatory constraints around the use of machine learning. FedML was proposed for learning from datasets generated by and stored on small devices – keyboards [MMRyA16], where different data sets will not show identical patterns and may vary considerably in size. At XAIN, we want to pioneer the transfer of research in Federated Machine Learning into industrial practice.

Scope of FedML at XAIN

Specifically, we stress that we mean to develop FedML as a software framework and a technology infrastructure whose use cases are not confined to machine learning on small devices and datasets – an association many make with FedML due to its initial keyboard use case. Datasets hosted on enterprise servers are firmly in scope for us.

A class of use cases that is of particular interest to us is that of a company C that is data controller for a range of data sets \mathcal{D}_i for a set of its clients i in I . Here, regulations prevent company C from computing the union of all data sets \mathcal{D}_i in order to perform machine learning on that union. But company C meet demands of data privacy when it performs FedML on these datasets in order to compute a global model. We will discuss the legal aspects of FedML technology in more depth in Section 5.

We conceive FedML as a software *framework* since there is no sole FedML algorithm that is best for all use cases. Rather, requirements will allow us to group use cases into different types, and where such a type informs the configuration of a generic FedML algorithm into a concrete instance. Over time, we plan to build up an infrastructure through which we can host our FedML services for our clients.

3.1 A GENTLE INTRODUCTION: FEDERATED AVERAGING

We will now present the original FedML algorithm informally but also at a more technical level. This minimal level of technical exposure is needed so that we can have a subsequent di-

scussion of security and privacy issues around the use of a FedML framework. The *FederatedAveraging* algorithm of [MMR⁺17] is depicted in Figure 1.

The intuition of the algorithm is that clients will learn local models based on their local datasets and that a parameter server will aggregate these local models into a global one that will then be used again for local learning. Let us describe this in more detail. At each round, the Parameter Server selects a set of clients of fixed size, and asks each selected client k to update their local model based on the current global model and the local dataset \mathcal{D}_k of that client, and to send this updated model to the Parameter Server. Once the Parameter Server has received all these updated models, the weighted average of all of these is computed as the updated global model which is then sent to all selected clients of the next round.

It is not important to understand the pseudo-code in function *ClientUpdate* for the update of local models, except for noting that all parties apply the same learning algorithm (here a Gradient Descent method) and on the same model structure (abstracted here into a vector of size d).

We note that the selection of clients in *FederatedAveraging* is done *randomly*, and so this is *Stochastic Gradient Descent*. Other selection strategies are possible. The weights are determined by the relative size of a client's dataset, but this is similarly only one possible choice. Further, we note that FedML is fairly agnostic in the choice of learning algorithm used by clients, i.e. how function *ClientUpdate* is being implemented, as long as the resulting instance of FedML offers good performance or theoretical convergence guarantees.

These degrees of freedom in choice illustrate that FedML can be seen as a generic algorithm in which mechanisms such as client selection, weight determination, aggregation function (here a weighted sum), and locally used learning algorithm are additional parameters whose instantiations result in a concrete instance of FedML. We thus think of FedML

```

// executed by a central, trusted "Parameter Server"
initialize  $w_0$ ;
for (round  $t = 1, 2, \dots$ ) {
     $S_t = \text{random set of } \lceil C \cdot K \rceil \text{ clients};$ 
    for (each client  $k \in S_t$  in parallel) {
         $w_{t+1}^k = \text{ClientUpdate}(k, w_t);$ 
    }
     $w_{t+1} = \sum_{k=1}^K \frac{n_k}{n} w_{t+1}^k;$ 
}

// executed by each client locally
ClientUpdate( $k, w$ ) {
     $\mathcal{B} = \text{partition of } \mathcal{D}_k \text{ into batches of size } B;$ 
    for ( $i = 1, \dots, E$ ) { // local training epochs
        for ( $b \in \mathcal{B}$ ) {
             $w = w - \eta \cdot \nabla l(w; b);$  // local SGD steps
        }
    }
    return  $w$ ;
}

```

Figure 1: Pseudo-code for algorithm FederatedAveraging of [MMR+17]: C is the percentage of clients that randomly participate in model updates in each round, $\{1, \dots, K\}$ is the set of all clients, w_t is the global model learned after round t , whereas w_t^k is the local model learned after round t (both w_t and w_t^k are d -dimensional real-valued vectors), variable n_k denotes the size of local data set \mathcal{D}_k where $n = \sum_{k=1}^K n_k$, parameter η is the learning rate, and $l(\cdot, \cdot)$ is the loss function used for local learning.

as a software framework. In particular, we may therefore optimize such choices for particular use cases and their requirements. And such optimizations may be achieved through standard tools, including those of machine learning itself.

Another aspect to consider is that FedML is also agnostic to how clients integrate globally learned models into their own decision-support systems. For example, they may set their own criteria for when the global model would be the source of truth for their own decision support and thus replace a local model. This flexibility is nice but it has a flip side: other clients and the Parameter Server do not have any guarantees that the updated model w_{t+1} that client k returns to the Parameter Server has really been computed by the intended implementation of function *ClientUpdate*. We will revisit this issue below when discussing the security of FedML.

3.2 TYPES OF LEARNING

FedML can support different types of learning in order to meet the varying needs of use cases. In [YLCT19], three types were identified and their explanation benefits from defining more structure of local datasets \mathcal{D}_k . In [YLCT19], \mathcal{D}_k is conceptualized as a $|\mathcal{I}_k| \times (|\mathcal{X}_k| + |\mathcal{Y}_k|)$ sized matrix whose rows are indexed by a set of sample identities \mathcal{I}_k , and whose columns are indexed by a set of features \mathcal{X}_k and by a set of labels \mathcal{Y}_k . We note that there may be no labels (when \mathcal{Y}_k is empty) and that we abuse notation and sometimes use the term “feature” to refer to an element of either \mathcal{X}_k or \mathcal{Y}_k .

For effective federated learning, better results may be obtained if semantic links across datasets are recognized and incorporated into the machine learning. These link patterns are grouped into three types, the first one being:

- ▶ **Horizontally FedML:** All datasets have the same or almost the same features and labels. And datasets have identities that do not overlap significantly: more formally, for all $k \neq k'$:

- ▶ there are many features that are in both sets \mathcal{X}_k and $\mathcal{X}_{k'}$,
- ▶ there are many labels that are in both sets \mathcal{Y}_k and $\mathcal{Y}_{k'}$,
- ▶ there are no or very few labels that are in both sets \mathcal{I}_k and $\mathcal{I}_{k'}$.

As an example of this semantic link type, consider the datasets of different local branches of the Department of Motor Vehicles. They are likely to use the same features and labels in their datasets, for example by relying on the AI provider for all branches in the same state. But there will be few users that have registered vehicles in different local branches of the same state. Another type of semantic link structure is dual to the one above:

- ▶ **Vertically FedML:** All datasets have the same or almost the same identities. Datasets have feature and label sets that do not overlap significantly: more formally, for all $k \neq k'$:

- ▶ there are many identities that are in both sets \mathcal{I}_k and $\mathcal{I}_{k'}$,
- ▶ there are no or very few features that are in both sets \mathcal{X}_k and $\mathcal{X}_{k'}$,
- ▶ there are no or very few labels that are in both sets \mathcal{Y}_k and $\mathcal{Y}_{k'}$.

For example, a local branch of the Department of Motor Vehicles will have a dataset with a set of users (the identities) that overlaps significantly with the set of users from a dataset of consumer sales data at a local supermarket. And the features and labels will not overlap significantly, given the different domains.

We stress that expressions such as “many” and “no or very few” are not meant to be for-

mal mathematical definitions. Rather, they are guidelines for classifying FedML use cases and aid in understanding what is required for implementing different classes of use cases.²

We also consider a type for which there is no significant semantic link across data sets:

- ▶ **Federated Transfer Learning (“Transfer FedML”):** All datasets have no significant overlap between their identity sets, feature sets, and label sets: more formally, for all $k \neq k'$:

- ▶ there are no or very few identities that are in both sets \mathcal{I}_k and $\mathcal{I}_{k'}$,
- ▶ there are no or very few features that are in both sets \mathcal{X}_k and $\mathcal{X}_{k'}$,
- ▶ there are no or very few labels that are in both sets \mathcal{Y}_k and $\mathcal{Y}_{k'}$.

As an example, consider a maintenance company that services rail transportation infrastructure (tracks, signal points, and so forth) and has maintenance data where identities refer to staff who do inspections and service. A rail company has a dataset in which identities are train numbers such as “RE 2” and where features contain information such as which locomotive ran this service most often, and where labels include information such as meeting a punctuality metric. It is clear that there will be very little overlap between identities, features, and labels across these datasets. But some overlap, for example pertaining to problems recorded on specific segments of rail tracks, would improve the quality of learning.

In general, there may be valuable statistical relationships between datasets that Transfer FedML could enable in order to get better decision support.

Discussion

Let us discuss these three different types. Horizontally FedML is a natural starting point for technical development and commercial use cases for the following reasons:

- ▶ Matching links between features and labels does not seem to raise any privacy or other regulatory concerns, so this is easy to operationalize.
- ▶ With no links between identities, we do not have to worry about how to implement such links without compromising regulatory demands such as those pertaining to privacy.
- ▶ Many companies hold structurally similar or identical datasets for different clients, where the semantic link structure is reflected very well by Horizontally FedML.

It seems that Transfer FedML has similar benefits in that it does not appear to require the establishment of semantic links between identities, and so this seems to not raise any regulatory issues – where such issues may or may not be addressable with technological means.

In contrast, as stated in [YLCT19], the establishment of semantic links across identities of different datasets for Vertically FedML requires a different architecture that relies on cryptographic protocols. This is so since two Clients should only learn which identities their datasets have in common, not which identities the other dataset has that their own does not have. However, these protocols allow at least one client to learn which of the identities of her own dataset also occur as identities in datasets of another client. Moreover, this client will also learn which identities that occur in her own dataset do not occur in the other data sets – as the protocol computes the intersection these two sets of identities.

This is very problematic from the perspective of data protection, notably GDPR, and will therefore not be a feasible approach in most practical use cases that fit the semantic link structure of Vertically FedML. Therefore, *our FedML platform development will focus on Horizontally FedML first and consider Transfer FedML use cases next.*

3.3 AGGREGATION MECHANISMS

The aggregation in the FederatedAveraging

algorithm in Figure 1 uses weights n_k/n that are the quotient of the size of local datasets with the sum of sizes of all datasets. It is an implementation detail that n is not the sum of all sizes of datasets of selected clients, presumably since the initial use case for this involved unbalanced data [MMRyA16]. But this illustrates that the definition of the aggregation operator, here the definition of weights, may want to reflect by requirements of the use case, here presumably the fact that data unbalanced across datasets.

Therefore, we think of our FedML software as a framework that can easily be instantiated to best meet the needs of use cases, and where aggregation operators are part of such instantiations. To illustrate this, here are a few sensible choices of aggregation operator:

- ▶ **Byzantine Gradient Descent [CSX17]:** the aim is here to offer robustness of the aggregation operator in the presence of up to q Byzantine clients in each round, where q is less than³ $K/2$. Byzantine clients can behave arbitrarily, for example they may not send any model updates or flawed ones to the Parameter Server, i.e. such nodes are malicious. Different rounds may have different Byzantine nodes. This is thus a strong security model. In [CSX17], it is proved that a variant of a FedML algorithm is secure under this security model when local gradients are combined in batches using a geometric mean. The geometric mean of a set of points in a d -dimensional, real vector space can be seen as the vector that has the minimal Euclidean distance to all points in this set.⁴
- ▶ **Secure Aggregation:** a practical means of securely aggregating weighted local models is presented in [BIK⁺17]. This uses a secure multi-party computation protocol that is bespoke for the computation of a sum of values and that is resilient against the dropping out of a number of clients, both under the honest-but-curious (HBC) and the active attack (ACT) models defined on page 13 below.

³Formally, when $2 \cdot (1+\epsilon) \cdot q \leq K$ for some fixed, positive ϵ .

⁴In [CSX17], the algorithm lets the Parameter Server compute gradient descent steps whereas for the algorithm in Figure 1 this is done by the clients, and in several steps over local batches, before communicating an updated local model; it would be very interesting to transfer the techniques of Byzantine Gradient Descent to this setting to get strong security guarantees while still having good convergence properties.

- ▶ **Automated Update of Weights:** online optimization algorithms could be designed that dynamically adjust the weights for aggregation in order to reflect observed performance and other metrics of interest. For example, a client who repeatedly submits updated models that seem to have poor quality would see her weight being decreased accordingly. The use of reinforcement learning may lead to effective such algorithms that also offer sufficient degrees of resiliency against adversarial manipulation.
- ▶ **Bespoke Operators:** Some use cases may require bespoke aggregation operators. For example, XGBoosting combines techniques from stochastic gradient boosting [Fri02] with ensemble methods on decision tree models to compute fast and accurate models. These staged combinations of models may use their own operators such as majority votes, and are consistent with the notion of aggregation operator for FedML.

3.4 SECURITY

Cybersecurity of software platforms is no doubt important and our platform development will follow Security By Design principles (see for example [Smi12]) and seek corresponding quality certifications. But here we want to focus on security aspects that are specific to the realization of federated learning.

We already saw how requirements of a use case or regulatory demands can drive the need for such additional security. For sake of illustration, consider a use case in which several car manufacturers want to mutually benefit from data for driving behavior across brands. These companies may then not trust each other and so the aggregation operator needs to be resilient to adversarial manipulation, for example by using some of the techniques outlined in Section 3.3.

Similarly, consider a use case with small local datasets and where the Parameter Server is not a data controller of datasets. Then, additional cryptographic or information-theoretic

means may be required for the aggregation operator so that the Parameter Server or other clients cannot possibly make any inferences about privacy-sensitive data in local datasets. Such means may include secure multi-party computation protocols, homomorphic encryption or the addition of noise to signals for obfuscation. Such choices may also decentralize aggregation and so eliminate the need for a Parameter Server or may offer the ability to challenge the Parameter Server.

In contrast, some use cases will not require such additional machinery and so can rely on simpler and more efficient platform instances. For example, consider a sole company as data controller of many different datasets and where the company, as Parameter Server, is capable of reading such local data anyway. In such use cases, the question is rather whether the company as data controller is using the local datasets in the FedML instance in a manner that is consistent with the contractual framework of the use case and existing regulation such as GDPR. We revisit this issue in Section 5.

The `FederatedAveraging` algorithm in Figure 1, works well when all clients and the Parameter Server are honest-but-curious: all parties play by the rules of the protocol but may make offline inferences from information they learn. In particular, no data poisoning attacks or other active attacks will take place. However, curious clients or the Parameter Server may, offline, attempt to recover private data from information that they are legitimately learning during execution of this algorithm. Therefore, even under this honest-but-curious security model, hardening mechanisms for model aggregation may be required for both security and regulatory compliance.

When one or more clients behave maliciously, the `FederatedAveraging` algorithm will no longer function correctly. But we already mentioned hardening mechanisms such as Byzantine Gradient Descent and Secure Aggregation that can protect against this.

3.5 DATA PRIVACY: TECHNICAL PERSPECTIVE

Data privacy is a central aspect in the design, implementation, and deployment of our FedML software framework and its infrastructure. A technologist may want to do this by deploying state of the art techniques for privacy preservation such as homomorphic encryption. A lawyer would then review the implemented system to judge whether it meets legal and regulatory demands, based on well understood legal tests such those for the EU GDPR and “best practices”.

At XAIN, we firmly believe that legal reviews and technology decisions should go hand in hand when designing a FedML platform, and this is indeed the approach we will take – by collaborating with leading technology lawyers in this space. This is also in line with the principles behind GDPR, where Privacy by Default should be an integral part of a system’s overall design. We will discuss the relevant legal issues for this in more depth in Section 5.

In order to prepare for this discussion, let us now conceptualize the essence of FedML mechanisms at a level of abstraction that accommodates different FedML instances of our software framework but ignores details that won’t have legal relevance for data protection. This conceptualization is meant to be accessible to lawyers, technologists, and business people who own use cases:

Participating Parties:

There are two types of participants:

- ▶ A Parameter Server that aggregates locally updated models to generate an updated global model, and sends the latter to all clients
- ▶ Clients that use the global model and their local datasets to generate an updated local model that they then send to the parameter server.

What concrete tasks the Parameter Server performs will depend on the instance of the FedML framework. In the algorithm of Figure

1, it selects clients and waits to receive their locally updated models from which it computes and communicates a new global model. In the Secure Aggregation approach of [BIK⁺17], the Parameter Server executes a quite different, 4-round protocol in which it never learns any values of locally updated models.

Staged Analysis of Data Protection

The above approach suggests that we stage our analysis of data protection. We want to show that nobody can learn personally identifiable information (PII) in this FedML process, or that if they can do that then this is legitimate given their role – for example as data processor or data controllers. The stages are:

- S1 Understand which party gets to learn the values of which of the locally updated models. We assume that all parties get to learn the value of the global model.
- S2 Given the understanding in S1, we want to understand the capabilities of parties to then learn anything about the PII residing within local datasets.

The advantage of such staging is that we can concentrate on the analysis of the FedML algorithms. If we can show that they offer strong protection against learning values of locally updated models, and that parties will not be able to extract PII from their knowledge of locally or globally updated models, then we do have a strong and compelling case that FedML offers data protection at a level that meets the demands of regulation such as GDPR.

This is so since parties then will not gain any information from the FedML algorithm that they could use to launch such attacks on data protection – or to launch them more effectively than in the case when FedML would not be running at all. For data protection, we here assume that an attacker aims to recover parts of a local dataset. We do not assume that specific parts are targeted, and we recognize that such recovery could be used to re-identify data subjects. For example, an attacker may com-

promise PII by getting to know some values of that matrix – as the values of some features and labels may suffice to make inferences about PII.

Security Models

Let us recall here two standard security models, since we want to understand how they can inform a legal opinion about FedML:

HBC *Honest-But-Curious Attackers*: this assumes that all participating parties behave according to the rules set out in the algorithms and protocols that they use.

ACT *Malicious, Active Attackers*: this assumes that parties may have an arbitrary behavior, including the breaking of rules or that external parties may impersonate participants.

To illustrate, in the first security model HBC, a selected client will not refuse to send his locally updated model and will indeed send that model without modifying its value. And the parameter server will dutifully perform the specified aggregation and send back the computed value without any manipulation. But clients and the parameter server may use the information they learn from the FedML computation (including the global updated models and, in the case of clients, their own locally updated models) in order to make inferences about data values in D_k for clients k .

In the second security model ACT, a selected client may refuse to send a model to the parameter server, may send a manipulated such model, may try to impersonate another client, and so forth. And the parameter server may refuse its service, manipulate the client selection mechanism, manipulate the aggregation process, and so forth.

Both security models may be relevant for our FedML platform, depending on the use case. For example, a data controller who controls datasets of all clients on his own premise and performs the “local” learning may be a good fit for the honest-but-curious security model

as any malicious attacks would fall under the scope of cyberattacks launched by external forces on company systems, where existing mechanisms such as security operations centers, cyberinsurance, and so forth would deal with this. And the active attacker security model is appropriate when there is not sufficient trust between clients and between clients and the Parameter Server.

Assumptions on Aggregation Operator of our FedML Platform

As already stated, our FedML platform will have to accommodate a number of aggregation operators, not just the one shown in Figure 1 for FederatedAveraging. This is because not all use cases will have the same (not necessarily legal) requirements pertaining to security or privacy. We therefore state some assumptions that should hold for a wide range of aggregation operators, including those that our platform will implement:

- A1** At least two clients submit their updated local models w_{t+1}^k to the parameter server at iteration $t+1$.
- A2** The parameter server will compute the new updated global model w_{t+1} as a function of the received updated local models w_{t+1}^k and sends w_{t+1} to all clients.
- A3** So this aggregation operator is a function that takes locally updated models as input and outputs an updated global model. The specification of this function is public knowledge.⁵

Apart from this, an aggregation operator may use additional privacy-preserving techniques that may help with meeting legal requirements for data protection, for example:

- ▶ A publicly verifiable random function that select the set of clients S_t such that the parameter server will not know who these clients are, nor will clients in S_t know which other clients are in S_t . Such unpredictability can strengthen the security of aggregation protocols. One could also imagine the use of zero-knowledge proofs so that a client

can prove to the Parameter Server that she is entitled to submit a locally updated model without revealing who she really is. These techniques can make it much harder to attribute locally updated models that were input to the updated global model to their actual clients.

- ▶ Use of homomorphic encryption so that the parameter server will perform the aggregation on ciphertext of locally updated models and return the resulting ciphertext that clients can then decrypt. This can help, since the parameter server would then not learn the values of locally updated models. But it would still know the value of the global updated model.
- ▶ Alternatively, secure aggregation protocols such as the one in [BIK⁺17] can be used so that the Parameter Server will not learn the values of any locally updated models while still computing and communicating the value of the updated global model.
- ▶ Secure multi-party computation – of which the one in [BIK⁺17] is an instance: since the aggregation function is assumed to be public, the updated global model as its output can be computed from the locally updated models of selected clients as private inputs. Clients would then only learn the global updated model and the “acceptable leakage”, which is what can be inferred by a client from his own private input, the knowledge of the public function (aggregator), and the knowledge of its public output (the global updated model).
- ▶ Differential privacy techniques [DR14] for the creation of noise may help here as well. This approach offers protection against the ability to distinguish between two databases that differ in only one entry. We think that it needs to be further evaluated how suitable this protection model is for the legal needs of data protection for FedML.

We stress that “privacy protection” here refers to protecting the values of locally updated models, not PII. Also, it is important to realize

that legal data protection demands may be met even when curious or malicious parties manage to learn values of locally updated models as long as one can demonstrate that such knowledge does not lead to the compromise of PII. We will discuss this further in the next section.

3.6 EXPERIMENTAL RESULTS: BENEFITS OF FEDML

Let us now demonstrate the benefits of FedML when compared to the unitary setting in which machine learning takes place on each dataset separately, without any sharing of locally updated models or computation of a global model. For sake of illustration, we report experimental results obtained by using a standard machine-learning benchmark and depict the results in Figure 2.

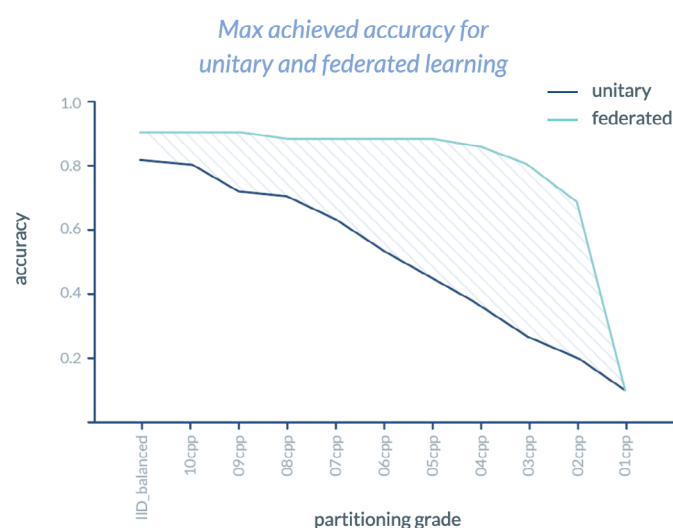


Figure 2: Benefits of FedML over the standard “unitary” approach of machine learning: model accuracy (y-axis) plotted over the grade of a partitioning scheme (x-axis). FedML makes much more accurate predictions when data is evenly or very unevenly balanced across partitions. Both approaches are ineffective in the pathological case, where each partition contains only one class used for classification of inputs. “Partition” here refers to a local dataset for FedML.

Experimental Setup

Our results in Figure 2 are based on the previously described partitioning schemes applied to the benchmark Fashion-MNIST. We train a convolutional neural network (CNN) consisting of

- ▶ a convolutional layer with 64 filters and kernel size 2,
- ▶ a second convolutional layer with 32 filters and kernel size 2,
- ▶ each followed by ReLU activations and max pooling,
- ▶ then a fully-connected layer with 256 units and ReLU activation, and finally
- ▶ the output layer which uses a soft max activation.

This model has a total number of 412,778 trainable parameters (more than four-hundred thousand). The three hidden layers use dropout probabilities of 0.3, 0.3, and 0.5 respectively. All layers use Glorot uniform initialization. In this experiment, the machine learning problem is the classification of inputs. Intuitively, a model captures a set of classes and assigns each input to one of these classes. The experiment takes an entire dataset and then applies different partitioning schemes to that dataset, where each such scheme gives us a set of partitions. To relate this to the FedML setting, each partition would then amount to a local dataset in the FedML terminology.

Discussion of Results

As we can see from the horizontal axis of the graph in Figure 2, the schemes we used range from those that have all classes present in each partition (left, called `IID_balanced`) to those that have only one class present in each partition (right, called `01ccp`). The vertical axis shows the accuracy of the models computed by unitary machine learning (blue line) and FedML (orange line), respectively. The accuracy is a statistical measure that computes the proportion of true predictions – both true positives and true negatives – from all the inputs that have been classified by the model.

We can see that the benefits of FedML are already visible when the data is independent and identically distributed across partitions (`IID_balanced`). FedML can do better here since it simply has access to more data, whe-

re this access is indirect through the repeated aggregation of locally updated models.

At the other extreme of data partitioning, we see that FedML and the unitary approach fare equally poorly when the partitions are such that they cannot really distinguish between classes of input. Such artificial partitions are a pathological case for any attempt of learning classifications.

Crucially, we can see that FedML clearly outperforms the unitary approach when the data is distributed in a very uneven fashion across partitions: for FedML, the accuracy is consistently higher and deteriorates much more gracefully when subjected to more and more pathological partitioning schemes.

Let us stress that FedML use cases typically have little or no control over the partitioning scheme, as local datasets cannot be merged or repartitioned with other datasets prior to learning. We varied these schemes here merely for experimental insights. In other words, a FedML use case will have a given partitioning whose statistical nature will very likely be that data are neither pathologically distributed nor indential and independently distributed. Therefore, practical use cases are better represented by the partition grades in Figure 2 for which FedML offers huge benefits.

4. Our FedML Platform

In this section, we want to outline the technical vision of our federated machine learning platform and the key mechanisms and principles that underpin and will realize this vision.

Open Source

One such principle is that we want to offer the core of the technology stack for our FedML platform as an open-source project. We plan to open source our FedML project under Apache License version 2.0. We do this partly because we firmly believe that no company should face technology lock-in or any barriers posed by proprietary software.⁶

Independent self-hosting

As our technology is provided as open-source, you have the option of downloading and integrating it into your own AI projects and infrastructure. For this, we will offer an SDK whose specifications we also provide. However, using our FedML technology in this manner will require considerable in-house expertise for such integration. And you then will have to consider whether this way of using XAIN's FedML technology is more beneficial to you than relying on the XAIN full-service platform, which we outline next.

Full-service platform

XAIN's full-service platform will be built on the open-source software that we develop and share, but it will also contain proprietary parts that are specific to the execution of our platform, notably monitoring, advanced aggregation mechanisms for increased privacy and learning accuracy, and user interfaces. We offer this full service so that you and your company can onboard to FedML use cases with ease, get a robust, reliable and scalable FedML execution, and have quality monitoring and seamless user experiences, and all of this at known and budgetable costs. This full service will therefore allow you to focus your energy on building and running your AI application.

Additional benefits

If you want to host our full-service solution on premise or in some particular compute environment, we can provide specific instances

of our platform to accommodate these needs. This is particularly useful when you want to do federated learning across companies within your enterprise group or if you have datasets from different clients all stored within your company premises. Another service we may offer in the medium to long term is support for automated machine learning, such as Effective Neural Architecture Search.

To summarize, we have designed our managed FedML service solution in a manner that makes the use of Federated Machine Learning as easy and seamless as possible. That way, you can focus on what matters most to you, namely building meaningful and effective AI use cases. By using our full-service platform, you will be sure that your participation in federated learning will be GDPR compliant. In particular, you will not have to rely on implementing costly and complex data anonymization techniques nor have to create and manage data lakes.



5. Compliance

As already stated, there is evidence that many decision makers in enterprises are concerned about compliance and data-protection issues when it comes to considering the use of AI applications or using cloud infrastructure for their deployment [Res19]. In that context, we already highlighted the enormous potential that FedML has in enabling machine learning use cases whose local datasets cannot be combined due to compliance reasons. In the last section, we also discussed data-protection issues for FedML from a more technical perspective.

Now, we discuss what technical or other means may be required to make FedML usage itself compliant. While there are many, in part sector-specific, forms of compliance, our FedML technology will consider data privacy (pertaining to natural persons) and data protection (including the protection of sensitive information that is not about a natural person). For sake of exposition, we limit our discussion here to GDPR, as the central EU Directive for this, which regulates data privacy for natural persons.

We fully recognize that GDPR does not consider which actors may have and use means for compromising data privacy and data protection. Nonetheless, we will do a case analysis over different actors in FedML as these will have different states of knowledge and so different means to be used. The completeness of such a case analysis will then ensure that our legal opinion does not depend on which actor means to compromise sensitive data in local datasets.

5.1 DATA ANONYMIZATION AND GDPR

We recall our argument that data anonymization techniques are too costly and not scalable as an alternative to FedML for enabling machine learning use cases. From a legal perspective, however, there is a helpful analogue between data-anonymization techniques and technical realizations of FedML. Let us quote Recital 26 of the GDPR to that end, shown in Table 1. Recital 26 sets out that data protection is applicable to information pertaining to an identified or identifiable natural person.

It then points out that pseudonymization of such information will still render information on an identifiable natural person, and so this will transform personal data into personal data.

Crucially, Recital 26 implies that a data controller needs to consider whether taken data protection measures such as anonymization techniques are sufficient so that one cannot identify a natural person through “*all the means reasonably likely to be used*”. The latter is essentially a risk assessment, informed by the current state of the art of such means, their cost and availability, and the likely capabilities of anyone who attempts such identification. Of concern are the singling out of an individual, the linking of different information about the same individual, and the ability to infer other information about that individual.

The above risk assessment may also have to reflect how the state of the art may evolve, for example for guaranteeing that an anonymization technique irrevocably removes links to PII. Anonymized data that meets such, very high, demands is therefore expressly not under the scope of GDPR and so cannot break any compliance with GDPR.

We now argue that the legal considerations around the protection of information about a natural person and data anonymization can be fruitfully transferred and expanded to the setting of FedML:

- L1 Local and global models learned in a FedML instance are data. We therefore mean to evaluate whether such data generated by instances of FedML meet the legal tests associated with anonymization techniques.
- L2 This approach to a legal interpretation of FedML also lends itself to broadening the scope of data protection to not only information pertaining to a natural person but also to other sensitive information contained in local datasets.

“(26) The principles of data protection should apply to any information concerning an identified or identifiable natural person. Personal data which have undergone pseudonymisation, which could be attributed to a natural person by the use of additional information should be considered to be information on an identifiable natural person.

To determine whether a natural person is identifiable, account should be taken of all the means reasonably likely to be used, such as singling out, either by the controller or by another person to identify the natural person directly or indirectly.

To ascertain whether means are reasonably likely to be used to identify the natural person, account should be taken of all objective factors, such as the costs of and the amount of time required for identification, taking into consideration the available technology at the time of the processing and technological developments.

The principles of data protection should therefore not apply to anonymous information, namely information which does not relate to an identified or identifiable natural person or to personal data rendered anonymous in such a manner that the data subject is not or no longer identifiable.

This Regulation does not therefore concern the processing of such anonymous information, including for statistical or research purposes.”

Table 1: Quote of Recital 26 of the EU GDPR

The latter point made in L2 is important. For example, an enterprise is a legal entity and its sensitive information, say, whether it plans to construct a new plant in a certain location, should not be inferrable from global models or local FedML models of other enterprises. Therefore, the inability of singling out, creating links, and making inferences about such entities – mentioned above – are equally relevant when protecting information that is not about a natural person.

Item L1 says that we want to analyze whether, and under which circumstances, the models generated by FedML constitute anonymized data. Through such analyses, we and others are able to shape and evaluate legal interpretations of FedML instances in regard to data privacy and the protection of sensitive data.

We will now reconnect with the discussion of Section 3.5 in order to assess how the use of certain techniques can ensure that the knowledge of global or local models does not help with the identification of personal or other sensitive data within local datasets over which FedML operates.

5.2 PRINCIPLE OF LEAST PRIVILEGE: REDUCING THE “MEANS USED”

The Principle of Least Privilege is a central principle in computer security, by which access rights for users are restricted to the bare minimum needed for them to perform their work. Applying this principle to FedML has the benefit that this reduces the means that actors may be able to use in order to gain access to sensitive or private information.

For FedML, the users are the participants of the FedML instance and the access rights are embodied in the communication protocols that specify which participants receive what messages; the latter include local or global models. We state some desired requirements for minimizing such access rights for FedML:

- R1** All participants get to know the value of the global model w_t for all rounds t .
- R2** Client k knows her local model w_t^k for all rounds t , but we require that client k has no actionable information about local models $w_t^{k'}$ of other clients k' .

R3 We recommend that the Parameter Server gains no information about the local models w_i^k for any round or client.

Item R1 captures the reasonable assumption that all legitimate participants get to know the value of the global model. In a conservative legal analysis, we may therefore assume that this knowledge is part of any means used to learn sensitive or private information about local datasets.

As for item R2, clients will of course know their own local models as these are trained clientside. But it stands to reason that clients should not learn the values of locally learned models of other clients. Otherwise, clients would obtain an additional means, whether effective or not, they could use for learning sensitive information about datasets from other clients.

FedML with exactly two clients ($K=2$)

This class of use cases requires a separate analysis. Using the FederatedAveraging algorithm in Figure 1, one can easily see that client 1 can compute the value of the local model of client 2 from the value of the global model and her own local model, and vice versa. This is so since averaging defines linear equations where the client only needs to solve for a single variable since K equals 2. Therefore, this special class of use cases seems to require additional technical means, such as the addition of noise, to ensure that clients will not get to know actionable information about the local model of the other client. “Actionable” here means that such information, together with any other means used, could reveal sensitive information about local datasets.

FedML with more than two clients ($K>2$)

In use cases in which K is greater than 2, we can argue that the linear nature of Federated Averaging does not seem to reveal actionable information about local models of other clients. Let us convey this through a simple analogy. Suppose you obtain the average height of three or more people and you know that this average contains your own height, 175cm. If the average is 180cm, then you only know that at least one of these people must be taller

than you. But there is little else that you can infer from this. For example, you may be the shortest person from that sample space or the second tallest one. It therefore stands to reason that such information is of a statistical nature and that this cannot be used to make informative inferences about local models, let alone about data stemming from the corresponding local dataset.

Parameter Server:

Trusted or Untrusted Party

Whether or not to follow the recommendation in item R3 will be markedly influenced by whether or not the Parameter Server is a party trusted by all clients. If so, it may be acceptable to let the Parameter Server get to know all local models – as is the case in the algorithm shown in Figure 1; such trust may also involve contracts drawn up between the entity that provides the Parameter Service and all Clients. However, the Principle of Least Privilege suggests to add techniques such as Secure Aggregation so that the Parameter Server is only a Coordinator and does not learn the values of local models.

For R3, a balance needs to be struck between potentially conflicting aspects. For example, a protocol in which the Parameter Server does not get to know values of local models ensures that the Parameter Server, as an attacker, will only know the global model and some history about communication patterns (say, which clients participated in which rounds). But this will not be sufficient to retrieve sensitive information from local datasets. Moreover, this approach may be preferred by the clients who are reluctant to share locally learned models for a variety of reasons. However, not learning the values of local models will make it harder to monitor the behavior of clients in terms of the quality of local models they submit. For example, we may want a client to be banned from participating in the next few rounds because she submitted low-quality models in two consecutive rounds, seeing this as a sign of incompetence or malice.

These considerations further strengthen our argument for building a FedML framework that

allows for such flexibility in order to accommodate the needs of use cases and the expectations of clients.

5.3 INFLUENCING FUTURE AI REGULATION

The EU is keen to provide an environment in which AI Innovation within Europe can thrive, foster economic growth, and be consistent with European values such as basic rights of citizens and ethical principles. The EU also recognizes that regulation about AI has to be formulated in a manner that does not stifle innovation and that can cope with the rapid pace and scope of AI development.

For example, we hopefully demonstrated above that approaches and legal tests around GDPR are transferable to FedML. In that context, it should be noted that the computation of local and global models is a form of data processing as well. So the contractual frameworks that govern the relationships between data controllers and clients would need to reflect this.

Another relevant legal aspect is that of Intellectual Property Rights (IPR), specifically pertaining to the learned models. One might argue that the IPR status of locally and globally learned models should be the same, since their computation is inextricably linked. This computation does not seem to offer, however, a degree of creativity that one would expect when wanting to copyright the expression of works. It also remains to be seen whether a FedML model could be patented, for example as a specific process for supporting a decision. Since all clients need to share the same model structure, the ability to patent a model will also depend on who created that initial structure (for example, a particular deep learning topology) and whether its creation was non-obvious.

Given that, we think that the ownership and usage rights of models learned through FedML should best be governed by contractual frameworks. That way, there will be legal clarity as to who owns these models and who has what usage rights over them.

At XAIN, together with a globally leading law firm, we therefore plan to craft legal contract templates that are specific to the use of FedML so that clients, data controllers, and subsequent users of learned models have legal certainty as to the rights and responsibilities about the FedML learning process and the use of FedML models in decision support. Together with said law firm, we furthermore want to help with shaping the legal opinions about FedML and its classes of use cases. We hope that, in doing so, we will positively influence future EU guidelines and directives for the use of innovative AI.

To summarize, at XAIN, we want to develop FedML technology that

- ▶ meets ethical guidelines,
- ▶ is consistent with an EU social contract on responsible innovation,
- ▶ has a clearly defined legal status,
- ▶ is fully compliant with regulations retaining to data privacy, and
- ▶ satisfies the needs of a wide range of industrial use cases.

6. Our Focus on Groups and Use Cases

As discussed, Google introduced the FedML technology as a means to apply machine learning for the auto completion of search queries entered on hand-held devices. This is perhaps why FedML is often thought to be about data originating from, or being stored at, small devices [ACG⁺16]. However, the FedML technology we are building at XAIN expressly includes use cases that involve large data sets hosted on enterprise servers. We believe that our approach provides a solution to a significant technology need: AI data protection, where compliance with regulation and the protection of sensitive data create such needs.

You can benefit from applying our FedML technology if you have the following needs:

1. You have identified a specific business need or problem that can be addressed by the use of machine learning.
2. The machine learning you consider for this requires you to process, for training purposes, datasets that contain personal or otherwise sensitive data.
3. These datasets are stored in different buckets or databases and cannot be moved or combined, given the regulatory constraints or the sensitive nature of data.
4. Anonymization techniques can merge data from buckets for centralized learning but you either lack resources or capabilities for this, or you are averse to the risks of their use.

To address these needs, XAIN is targeting, among others, the following user groups: Enterprises and startups, but also academia, and freelance developers or volunteers.

▶ Enterprises

XAIN is explicitly designed to enable AI applications by addressing GDPR compliance and sensitive data issues. This makes enterprise customers our first user group. Our initial use cases, highlighted in Section 7, target established enterprises that have AI endeavours for internal or external use cases.

▶ Startups

Our second major user group is comprised of AI-related startups. These have either limited AI capabilities in-house, too few data for training purposes or want to ensure that data-privacy regulations will be upheld throughout a continuous training process.

▶ Volunteers & Freelancers

Private individuals and developers may want to contribute to XAIN's technology either for fun or for profit. At a mature stage of our platform, certification programs might be established, offering extra benefits to this user group in particular.

▶ Academia

Our XAIN platform will not be over-engineered. But we value engagement with academia in research projects. This will assure our readiness for particular use cases (say, in the pharmaceutical industry) and advancements in technology that influence best practices in data privacy; it also fosters the strong research DNA of our company.

While our initial focus will predominantly be on the first two user groups for business development, we do welcome other types of users to join our network. We think that the advantages of our FedML technology are quite universal, and that all five user groups will have benefits from joining our effort and will make valuable contributions to XAIN.

6.1 BENEFITS WE OFFER

We believe that most companies will have and use AI-based software and that they will see the value in extending the scope of their machine-learning algorithms to larger data sets in order to get better insights for decision support. For example, a company may use our technology on datasets hosted at its subsidiaries, where these datasets cannot be aggregated for reasons of compliance or IT complexity. In addition, companies will have an incentive to join the platform as they will mutually benefit from insights driven by all datasets of participating third parties.

► Enablement

The infrastructure will enable entities who currently have low AI capabilities to use and exploit AI applications that they could not operate otherwise.

► Costs

AI training and development costs decrease due to network effects of our platform. Financial entry barriers for new enterprise clients will also be lowered.

► Privacy

Stored data will neither be copied nor moved, it stays entirely in the individual corporate environments, on-premise or in their cloud environments. Given correct operation, GDPR compliance holds.

► Scalability

Federated Learning creates network effects in AI training. The more participants are involved, the better the model accuracy will be.

► Security

The technology will support mechanisms that offer high data security and mitigate or prevent malicious practices and non-compliant behavior.

Our platform will be able to reflect the different privacy & security needs of use cases by using appropriate FedML instances.

We promote the democratization of AI, as each platform participant will profit from the learnings gained by others on their local datasets.

We will develop our software under an open-source license model that encourages commercial use and reuse.

We will offer software production and use-case support that meet the highest quality standards, for example through certifications from relevant agencies.

6.2 VALUES WE PROMOTE

We think that our technology captures core values that we, at XAIN, mean to promote and live by. We are driving by the principles that our AI work should be human-centric, deliver the highest quality with precision, and be completely trust-worthy and transparent.

We place great importance on a seamless and intuitive user experience and optimal value creation through participation in our AI-enabling platform.

We will build and maintain the platform with a high degree of cybersecurity and accountability in order to make it resilient against potential attacks such as data poisoning.



1
1 0
0 1
0
1

7. Sample Use Cases

It may be conceived wisdom that centralized or distributed machine learning will be the method of choice for deploying AI solutions in enterprises. However, it turns out that those approaches can only be adopted in relatively few use cases in practice. This stems from the fact that data is most often residing in data silos where either sensitivity issues or regulatory constraints prevent the creation of data bridges across silos or enterprises do not have the technical means of migrating and merging such data for subsequent machine learning. The latter is a particular issue with legacy systems.

Therefore, we firmly believe that FedML is a strategically important AI technology as it enables many more use cases for AI in practice. Table 7 depicts a classification of use cases into those that are confined to internally controlled data and those that also involve externally controlled data. We make this differentiation as the sources of data and their storage spaces have impact on the relevance and interpretation of data-privacy regulations. Internal use cases typically mean that the company is the controller of all data. AI is then typically used to improve in-house processes and decision making.

External use cases mean that there is more than one Controller or different datasets, which is more complex from a regulatory perspective and also may need trust management mechanisms as these Controllers do not necessarily trust each other. For external use cases, AI is most often used to offer benefits to customers, such as improved services. This classification is not meant to be strict but serves as a guide in configuring our FedML technology to a particular use case.

7.1 ANDY: AN AI APP BY XAIN

The first use case we want to outline is our own production-ready application ANDY. “ANDY” is an acronym for Anomaly Detection and the application supports accounting workflows around invoice processing in enterprises. We will use ANDY as a first AI app based on our FedML technology. ANDY provides an AI-backed solution to the accounting industry that could reap tremendous benefits from AI-backed automation and, at the same time, holds some of the most sensitive data firms own, namely supplier invoices. Backed with our technology, companies who will use ANDY will profit from pre-trained, yet privacy-preserving AI models.

The benefits of ANDY’s solution are ideally seen in its deployment within an enterprise group. When ANDY is applied to an entire enterprise holding or group, including all subsidiaries and affiliated companies, each of these entities has different databases that derive in part from legacy ERP landscapes. If one were to apply ANDY not only to one enterprise, but to the entire group, then all databases would first have to be aggregated if centralized learning were to be applied. Creating such a centralized data warehouse is a tremendous undertaking, which might render the application of an AI-backed solution unprofitable (the business case). And this is not even considering any legal or regulatory issues associated with such data aggregation.

Our FedML technology offers a viable alternative that avoids these issues, yet lets all members of the enterprise group benefit from a more accurate model learned with FedML.

	Internal Use Cases	External Use Cases
Data source	Your internally created company data is the only data source used for the AI application	You wish to use data sources from different customers for your AI training
Infrastructure	Training data is stored in several silos all under your control	Customer data is stored in separate data buckets, not all of which you control
Examples	Internal data analysis, Anomaly detection, Predictive maintenance, Transactions data	Customer services and Intelligent assistants, Customer analysis, Anomaly detection, Mobility sharing services

Table 2: Matrix of Use Cases

7.2 AI-BASED SOFTWARE-AS-A-SERVICE

Companies that want to build and use AI applications often have a lack of own data or face a poor quality of their own data. Making such applications production-ready therefore is costly and time consuming.

Fortunately, this does not have to be a reality for AI app builders, companies, and startups. Through our FedML technology, it suffices that we onboard only a small number of customers for an AI app/use case to get a first global model. With this in place, other customers can easily onboard and draw the benefits of learning with a model that already captures insights from larger datasets. At the same time, the existing customers benefit from the insights inherent in the datasets of new customers. All of this is happening without jeopardizing the data privacy and by protecting sensitive information for all participating customers.

The use case of ANDY, which we offer as a Software as a Service (SaaS), is thus a classic example of how SaaS business solutions can profit from FedML. First, the anti-centralization approach of FedML makes SaaS business models substantially more attractive to even the most conservative enterprise customers who need high assurance for information security and compliance.

Second, the results of an AI application running with FedML are likely to outperform considerably any competing solutions, which employ a combination of centralized training and data anonymization techniques. We already discussed how FedML can cut down or eliminate the costs for data anonymization and yield more accurate models, as it can tap into a rich source of datasets for machine learning. FedML also works well when datasets have different statistical distributions, for example when datasets capture behaviors of cars and their users.

7.3 ANOMALY DETECTION IN IOT AND MACHINE DATA

In the process of machine engineering, engineers rely on the monitoring of data to assess the health status of machinery and to manage

maintenance activities. Data are also an important input for providers of health-monitoring and predictive-maintenance software. The application of AI is currently booming in this area. We have held a substantial amount of discussions with industry representatives who have requested solutions for incidence-prediction models. The hypothesis is here that an AI can apply an anomaly-detection based model to real-time data, for example to test data or production-monitoring data, to predict the occurrence of incidents that are far from the normal behavior of engines or machinery. Engineers hope to use such AI applications to gain insights for detecting, mitigating, and even preventing such incidents – including those that impact safety.

Despite the obvious advantages of applying AI in this area, machine data is usually quite sensitive or contains personal data. For example, consider a machine that produces data based on its handling by a user. A motor engine of a car, for example, would provide continual information about the driving maneuvers of its users. Hence, such use cases also fall under the scope of privacy regulation even though their machine-centric nature may suggest otherwise.

7.4 FEDERATED AUTONOMOUS DRIVING

Autonomous driving is one of the most sought-after use cases of artificial intelligence, with huge financial investments in startups and established large enterprises. AI in autonomous driving also raises highly critical data-privacy issues. The data that are used for training algorithms that are at the heart of autonomous driving is usually of a highly sensitive and personal nature, as driving behaviour and history includes a substantial amount of personal information. This requires car manufacturing companies to store driving data in separate buckets per vehicle.

One reaction to this may be to anonymize all those data so that its combination can no longer link to specific vehicles or persons that use such vehicles. But we already discussed the problems associated with this approach in Section 2.3 in detail. Another alternative is to

synthesize “artificial” data that therefore has no link to private or sensitive data. But this is problematic for such a use case, as the generation of useful synthetic data is a very complex process that requires a lot of fine-tuning, and there is no guarantee that models trained on synthetic data will be accurate enough to perform well in use cases such as autonomous driving, where safety is one of the critical aspects.

In contrast, our FedML technology can be used here without the need to anonymize data or to generate synthetic data. As an external use case, this would allow different car manufacturers to learn together without having to worry about the compromise of sensitive data or about violating data privacy regulation. This could be a game changer for the use of AI in autonomous driving.

7.5 MOBILE APPLICATIONS

Generally speaking, AI applications that train on mobile devices have several issues whose resolution requires alternatives to the centralized approach of machine learning:

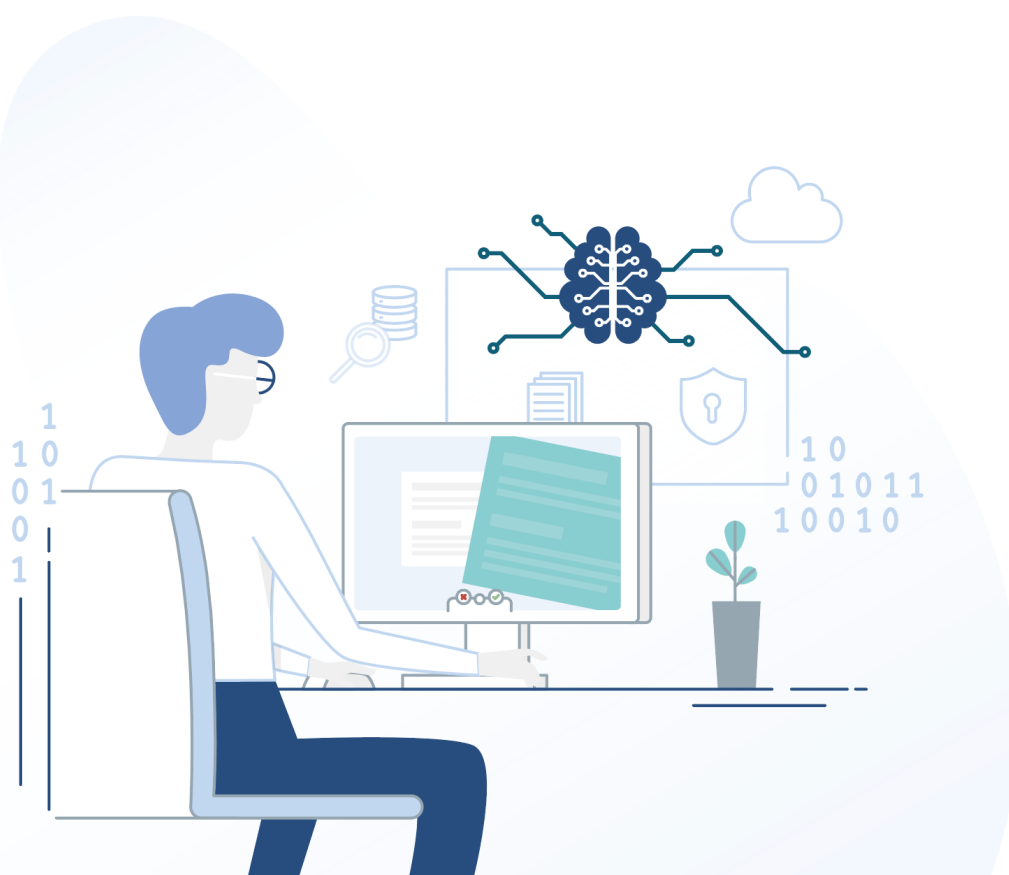
1. Most data produced on mobile phones originates from a user’s interactions with that device and is hence highly personal information. Examples could include data produced from

mobile-banking applications and second-layer authentication based on user behaviour.

2. Training is constrained by limitations of the device’s capabilities, including CPU performance, battery power, and communication latency. Mobile devices may be subject to connectivity issues or may not render enough computing power or storage for complex machine-learning applications. At the same time, the user experience of the device should not be impaired by any AI applications running on that device.

3. Datasets on a device are rather small. When training is confined to such local data only, training will produce models whose accuracy is likely to be too low, questioning the usability of such models.

Through FedML, providers of AI apps can effectively circumvent these problems. While personal data will influence the training of local models, which are then deemed to be personal data as well, models get aggregated into an updated global model in a manner that makes the latter non personal data. Therefore, one solves the issue of data privacy while also getting models that are accurate enough to effectively support decision making.



8. Conclusions

In this white paper, we began with a review of machine learning and its predominant approaches, centralized machine learning and its optimization of distributed machine learning. We then discussed why these approaches are problematic for the effective adoption of AI in practical use cases across commercial sectors. This led us to the identification of adoption barriers that our solution is expressly conceived to overcome.

Next, we reviewed data anonymization techniques, which are considered as the primary means of addressing data-privacy issues in the use of machine learning. We saw that their application is problematic as the regulatory burden on them to achieve data-privacy compliance is very high and since these techniques are not fully automatable and so are costly and do not scale.

We then gave a gentle introduction to Federated Machine Learning, as an approach that can overcome these issues while still being able to compute accurate models for decision support. For this to work, we identified that the technical realization of FedML requires both the formation of an authoritative legal opinion about FedML and its compliance, as well as an alignment of the design of FedML technology with such opinions.

We noted our engagement with a leading law firm so that we can offer thought leadership in this co-evolution of technology and its legal interpretation. This detailed discussion of compliance was followed by a description of the types of users of our FedML technology, including where our initial focus for engagement, product development and deployment will be.

Finally, we provided descriptions of pertinent use cases for our FedML technology. For this, we suggested that it is useful to classify such use cases into internal ones (where all data-buckets are under control of the company or its subsidiaries) and external ones (where data buckets from other entities, including potential competitors may be used).

We firmly believe that we made a compelling proposition for an AI technology, FedML, that has a clear and convincing potential of enabling the wider adoption of AI in commercial and industrial sectors to the benefit of all. Our belief comes, in no small part, from the fact that our proposition deals with two crucial pain points faced in that regard: data-privacy regulations and the protection of the confidentiality of sensitive data.

In the near future, we plan to publish details of the system design for our FedML technology, including reports on its performance against established benchmarks from machine learning.

We also plan to publish outcomes of the legal opinion on FedML technology and how this opinion will shape our technology road map. We are particularly excited about the advancements of this FedML technology when paired with such legal studies. Our technology should bring substantial benefits to all sectors that consider the use of AI. The EU has always put great importance on data privacy as a value. Now even the US and Chinese agencies are becoming more and more conscious of data-privacy issues and are currently working on new regulation directives.

But any regulation needs to also consider how it impacts future innovation and the social and economic well being of the people and institutions it regulates. The EU is not alone in seeking such a balance between the protection of social values such as privacy and allowing for innovation in AI to thrive. Japan's Text and Data Mining (TDM) copyright exception is another example of pursuing such a balance.

At XAIN, we want to develop FedML technology that respects social values such as privacy and mutually beneficial collaboration. And we are eager to build up a portfolio of FedML use cases that are deployed in production so that our FedML technology will get network effects and become The eXpandable AI Network (XAIN), the vision that led to the creation of this company.

References

- [2914]
Article 29. Opinion 05/2014 on Anonymisation Techniques. Legal Opinion by the Article 29 Data Protection Working Party, April 2014. 0829/14/EN WP216.
- [ACG⁺16]
Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep Learning with Differential Privacy. In ACM Conference on Computer and Communications Security, pages 308–318. ACM, 2016.
- [BIK⁺17]
Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017, pages 1175–1191, 2017.
- [CSX17]
Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *PO-MACS*, 1(2):44:1–44:25, 2017.
- [DR14]
Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.
- [ea18]
Philip Gerbert et al. The big leap towards ai at scale. Online, 2018.
- [Fri02]
Jerome H. Friedman. Stochastic gradient boosting. *Comput. Stat. Data Anal.*, 38(4):367–378, February 2002.
- [GZW17]
Suyog Gupta, Wei Zhang, and Fei Wang. Model accuracy and runtime tradeoff in distributed deep learning: A systematic study. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017, pages 4854–4858, 2017.
- [MMR⁺17]
Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA, pages 1273–1282, 2017.
- [MMRyA16]
H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *CoRR*, abs/1602.05629, 2016.
- [oCCA19]
The Association of Chartered Certified Accountants. Machine learning - more science than fiction. Online, 2019.
- [Res19]
IDG Research. Studie machine learning / deep learning 2019. Downloadable Report, 2019.
- [Smi12]
Richard E. Smith. A contemporary look at saltzer and schroeder’s 1975 design principles. *IEEE Security & Privacy*, 10(6):20–25, 2012.
- [yA18]
Blaise Agüera y Arcas. Decentralized machine learning. In IEEE International Conference on Big Data, Big Data 2018, Seattle, WA, USA, December 10-13, 2018, page 1, 2018.
- [YLCT19]
Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2):12:1–12:19, January 2019.
- [YTR⁺19]
Chen Yu, Hanlin Tang, Cédric Renggli, Simon Kassing, Ankit Singla, Dan Alistarh, Ce Zhang, and Ji Liu. Distributed learning over unreliable networks. In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, pages 7202–7212, 2019.